Regularizing Mono- and Bi-Word Models for Word Alignment

Thomas Schoenemann
Lund University, Sweden

Abstract

Conditional probabilistic models for word alignment are popular due to the elegant way of handling them in the training stage. However, they have weaknesses such as garbage collection and scale poorly beyond single word based models (DeNero et al., 2006): not all parameters should actually be used.

To alleviate the problem, in this paper we explore regularity terms that penalize the used parameters. They share the advantages of the standard training in that iterative schemes decompose over the sentence pairs. We explore the models IBM-1 and HMM, then generalize to models we term *Bi-word* models, where each target word can be aligned to up to *two* source words.

We give two optimization strategies for the arising tasks, using EM and projected gradient descent. While both are well-known, to our knowledge they have never been compared experimentally for the task of word alignment. As a side-effect, we show that, against common belief, for parametric HMMs the M-step is *not* solved by renormalizing expectations.

We demonstrate that the regularity terms improve on the f-measures of the standard HMMs and that they improve translation quality.

1 Introduction

State-of-the art approaches for word alignment are based on probabilistic models. They can be split into joint models (Melamed, 2000; Marcu and Wong, 2002) and conditional models (Brown et al., 1993; Vogel et al., 1996; Wang and Waibel, 1998; Toutanova et al., 2002; Sumita et al., 2004; Deng and Byrne, 2005; Fraser and Marcu, 2007a). While in early works the underlying basic entity was a single word, today's advanced approaches build on sequences of words, called *phrases*.

For joint models the advanced models are standalone approaches (Marcu and Wong, 2002). However, these models are computationally hard to handle, which frequently results in maximum approximations being made. This is different for the conditional models, which are easier to handle but where most approaches are based on initializing from single-word based models (Brown et al., 1993; Vogel et al., 1996; Al-Onaizan et al., 1999). However, the recent work of Mauser et al. (2009) deals with pairs of source words and is trained without considering single word based models.

In this paper we much generalize on this work, considering a class of models we term *Bi-word models*. We consider a variant of (Mauser et al., 2009) which we call Bi-1, then proceed to derive a Bi-HMM. Our main focus is however on regularizing such models. We first address known conditional models called *single-word based* models, focusing on a weakness known as *garbage collection*. We show that this weakness can be alleviated by adding an entry to every dictionary distribution as well as adding a regularity term (a weighted L_1 norm). Afterwards we generalize this idea to Bi-word models. The regularity term will now become crucial since the garbage problem is known to worsen for conditional models that generalize single-word based ones (DeNero et al., 2006).

We cast all this as compact objective functions subject to simplex constraints, and show two ways to optimize these: via EM and via projected gradient descent (Bertsekas, 1999, chap. 2.1). Since each iteration decomposes over the sentence pairs, the approach is efficient and scalable. In contrast to our recent work (Schoenemann, 2011) (where we used an L_0-norm) we do not use the maximum approximation and also address Bi-word models.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1313–1322,
Chiang Mai, Thailand, November 8 – 13, 2011.

Related Work on Word Alignment For a systematic comparison of the most commonly used models see (Och and Ney, 2003). Apart from the classical approaches, a few other lines of work have been pursued. Indeed, for single word based models regularity terms have been considered before, in particular in our recent work on the L_0-norm (Schoenemann, 2011). Otherwise most of the work has focused on combining asymmetric conditional approaches. Zens et al. (2004) intertwine the training of both directions by exchanging information in-between the iterations. Liang et al. (2006) propose to include the products of the conditional marginals for each training direction into the objective function. Graça et al. (2010) postulate that the posterior marginals for both directions be equal. They also propose an asymmetric variant that favors 1-to-1 alignments. The idea of posterior regularization has further been pursued in the machine learning community (Mann and McCallum, 2007).

We further note the approaches (Matusov et al., 2004; Taskar et al., 2005; Lacoste-Julien et al., 2006) that focus on the computation of alignments given symmetrized cost. Some of them also include novel ways to train the models.

Finally, our EM-scheme bears resemblance to the works (Berg-Kirkpatrick et al., 2010; Ganchev et al., 2010), but we address substantially different models.

2 Mono-Word Models

In this section we review the employed single word based models. We call them *Mono-word models* as we find the term more handy, in particular when it comes to distinguishing them from the pair-based models in the next section.

All discussed models formalize the (conditional) probability that a given English sentence $\mathbf{e} = e_1^I$, consisting of I words, produces a foreign sentence $\mathbf{f} = f_1^J$ with J words. This probability is denoted $p(\mathbf{f}|\mathbf{e})$. We will refer to $\mathbf{e}$ as the source sentence and to $\mathbf{f}$ as the target sentence. The considered models are all based on hidden variables called *alignments*. For Mono-word models the assumption is that each target word is aligned to at most one source position. The aligned position of target word j is denoted $a_j \in \{0, \ldots, I\}$, where 0 indicates unaligned words. The alignment of the entire sentence pair is denoted $\mathbf{a} = a_1^J$ and the probability is modeled as

$$p(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} p(\mathbf{f}, \mathbf{a}|\mathbf{e}) \ .$$

The models differ in how this new joint probability is modeled, but they all factor it as

$$p(\mathbf{f}, \mathbf{a} \,|\, \mathbf{e}) = \prod_j p(f_j \,|\, e_{a_j}) \cdot p(a_j \,|\, a_{j-1}, j, I) \ .$$

For the first term (dictionary probability) all models use the same non-parametric representation. For the second term (alignment probability) they differ. The IBM-1 simply sets $p(a_j|a_{j-1}, j, I) = 1/(I+1)$, resulting in a convex model. We also consider the non-convex HMM, which models $p(a_j|a_{j-1}, I)$, getting rid of the dependence on j. To avoid overfitting a parametric model is used, based on considering the difference $a_j - a_{j-1}$. Details are given in the next section.

3 Bi-Word Models

In this paper we consider a more general class of conditional models, which we call Bi-word models. Here we are much generalizing on the work of (Mauser et al., 2009).

Now each target word is allowed to align to up to *two* source words. The alignment of target word j is expressed as the tuple $(a_{j,1}, a_{j,2})$, where the allowed set of values is a subset of $\{0, \ldots, I\} \times \{0, \ldots, I\}$. The value $(0, 0)$ will denote unaligned words. In any other case we require that $a_{j,2} > a_{j,1}$. If $a_{j,1}$ is 0 the word is aligned only once. If $a_{j,1} > 0$ it is aligned twice. We further forbid the case where $a_{j,1} > 0$ and $a_{j,2} = I$ since at the sentence end the considered data usually contain a punctuation mark which aligns only once. Note that otherwise there are no restrictions, in particular we do not require that the two aligned words are at consecutive positions (although such knowledge could be enforced in our framework).

In the generative story of the models we first take a decision of whether the alignment of position j is a double alignment or not. We denote this by a binary variable $b_j \in \{0, 1\}$, where a value of 1 denotes a Bi-alignment. Obviously $b_j = 0$ implies $a_{j,1} = 0$. Afterwards we decide on the aligned positions and the identity of the target word:

$$\begin{aligned} p(\mathbf{f}, \mathbf{a}|\mathbf{e}) \ = \ \prod_j & p(f_j \,|\, e_{a_{j,1}}, e_{a_{j,2}}) \cdot p(b_j) \\ & \cdot p(a_{j,1}, a_{j,2} \,|\, b_j, a_{j-1,1}, a_{j-1,2}, I) \ . \end{aligned} \tag{1}$$

Note that compared to the Mono-word models we have now many more dictionary parameters, as well as much more probability mass to spread. Moreover, $e_{a_{j,1}}$ and $e_{a_{j,2}}$ can be the empty word $NULL$.

In this work we consider two models that generalize the IBM-1 and the HMM to the new set of alignments. We call them Bi-1 (a variant of Mauser et al.'s model) and Bi-HMM, and again they differ only in the alignment probabilities. The values $p(b_j=0)$ and $p(b_j=1)$ are chosen independently of j and fixed to 0.1 and 0.9 in this work.

The **Bi-1** is a convex model and treats non-Bi-alignments as $p(0, a_{j,2} \,|\, b_j = 0, I) = 1/(I+1)$, just like the IBM-1. For Bi-alignments it sets $p(a_{j,1}, a_{j,2} \,|\, b_j = 1, I) = 1/K$, where K is the number of possible Bi-alignments (and where $a_{j,1}, a_{j,2}$ is an allowed constellation). Note the (subtle) difference to the work of Mauser et al.: this work did not consider the variables b_j, so for long sentences the pairwise alignments become dominant. Further, for our models the word order matters, i.e. generally $p(f|e_1, e_2) \neq p(f|e_2, e_1)$.

The proposed **Bi-HMM** factors the alignment probability in a manner similar to the Mono-HMM. First of all, for a given alignment we introduce the notion of the *head* of target position j, denoted h_j. In case the position was aligned at least once, we define h_j as the smallest target position aligned to j, i.e. $h_j = a_{j,1}$ if $a_{j,1} \neq 0$ and $h_j = a_{j,2}$ else. In case of unaligned positions h_j is set to the head of the largest *aligned* previous target position. Hence we use a full first-order dependence, which in practice requires doubling the state space - see (Vogel et al., 1996). The alignment probabilities are now

$$p(0, a_{j,2} \,|\, h_{j-1}, b{=}0, I) \;=\; p_{inter}(a_{j,2} \,|\, h_{j-1}, I)$$

and for $a_{j,1} > 0$

$$p(a_{j,1}, a_{j,2} | h_{j-1}, b{=}1, I) = p_{inter}(a_{j,1} \,|\, h_{j-1}, I) \cdot p_{intra}(a_{j,2} \,|\, a_{j,1})\,.$$

Note that both cases rely on the same probability model $p_{inter}(\cdot|\cdot)$. The second case has an additional distribution $p_{intra}(\cdot|\cdot)$. Both are modeled separately using a parametric distribution described below. Note that $p_{intra}(i|i') = 0$ if $i \leq i'$.

Superficially the Bi-HMM looks similar to (Deng and Byrne, 2005). However, this latter is actually a Mono-word model.

Parametric HMMs. It is well-known that HMMs for word alignment perform best using parametric alignment probabilities. For both the Mono-HMM and the Bi-HMM, we follow Vogel et al. (1996) and consider only the difference $i - i'$ to model $p_{inter}(i|i', I)$. Here, only differences between -5 and 5 are modeled by separate parameters $r_{-5}, \ldots, r_5$, all larger differences are captured by a single parameter r_L. To make this a probability distribution, the latter parameter is spread uniformly over all possible differences (with absolute larger than 5) in the respective context. Lastly, we introduce parameters p_0 and p_1 ($p_0{+}p_1{=}1$), where p_0 denotes the probability for unaligned words. The alignment probability is now

$$p_{inter}(i \,|\, i', I) = \begin{cases} p_0 & \text{if } i = 0 \\ p_1 \, \frac{r_{i-i'}}{\tau_{i',I}} & \text{if } i > 0,\ |i - i'| \leq 5 \\ \dfrac{p_1 \cdot r_L}{\tau_{i',I} \, |\{i'' : |i'' - i'| > 5\}|} & \text{else,} \end{cases}$$

with[1]

$$\tau_{i',I} = \sum_{1 \leq i \leq I : |i-i'| \leq 5} r_{i-i'} + r_L \quad . \qquad (2)$$

A special case arises for the initial alignment probabilities $p(h_1{=}i|I)$. Rather than fixing them to $1/(2I)$ (including empty words), as is common, we model these parametrically (with renormalization, but without grouping).

In case of the Bi-HMM, there is further the probability $p_{intra}(\cdot|\cdot)$, which we also parameterize based on positive distances, grouping those larger than 5. In principle, each of the three arising distributions has its own parameter set. However, the initial probability and the inter-alignment model share the parameters p_0 and p_1.

4 Objective Functions

In word alignment one is given a large set of sentence pairs, not a single pair. We denote the sth pair by $\mathbf{f}^s, \mathbf{e}^s$. The standard approach to word alignment is maximum likelihood, i.e. minimizing

$$-\sum_s \log(p(\mathbf{f}^s \,|\, \mathbf{e}^s))$$

over the parameters of the model. Here, we are considering a conditional model, which can be any of the above mentioned.

[1] If differences of more than 5 are impossible, the term r_L is dropped from the equation.

Such models are known to have weaknesses called *garbage collection*. This refers to the phenomenon that rarely occurring source words tend to align to a significant portion of the target words in the respective sentences, since the probability mass of the frequent words is better used to explain the sentences without rare words. The effect is known to worsen when one moves beyond single word based models (DeNero et al., 2006).

It is known that joint models suffer less from this deficiency when dealing with the same set of possible alignments. However, joint models are usually hard to handle computationally, whereas the mentioned conditional models behave quite nicely. Hence, we use conditional models, but propose to alter the training criterion. We add a regularity term that penalizes the used probability mass in a (non-negative) weighted L_1 manner. We state this for Bi-word models, but note that the Mono-word models are included by fixing $e_1 = NULL$:

$$-\sum_s \log(p(\mathbf{f}^s \mid \mathbf{e}^s)) + \sum_{\mathbf{e_1},\mathbf{e_2},f} w^f_{e_1,e_2}\, p(f|e_1,e_2) \quad (3)$$

Here $w_{e_1,e_2} \geq 0$ are known weights (see below). For the new objective to make sense, we need to augment the parameter space: for every constellation e_1, e_2, we add a probability $p(NULL|e_1,e_2)$. In the standard ML-criterion this entry will always be set to 0. Not so with our new criterion: since we set the respective weighting factor $w^{NULL}_{e_1,e_2}$ to 0 it may be cheaper not to use the entire mass to explain the corpus.

Choice of Weights. When dealing with Mono-word models we only penalize rare words since they cause the garbage collection phenomenon. Let $N(e)$ be the number of times the source word e occurs in the corpus. If $N(e) \geq 6$ we set $w^*_{0,e}$ to 0, otherwise it is set to $\lambda[6 - N(e)]$, where λ is some weight. We found $\lambda = 2.5$ to work well.

For Bi-word models we presently set all Mono-word weights $w^*_{0,*}$ to 0. The Bi-word penalties are based on a value of $\lambda = 0.5$, but rare source word pairs pay a larger penalty (The equation is $\lambda \cdot \max\{1, 5 - N(e_1,e_2)\}$, where $N(e_1,e_2)$ is the number of times the pair e_1, e_2 occurred).

5 Optimization Strategies

We present two optimization schemes to handle the arising minimization problems: one is based on Expectation Maximization (EM), the other on projected gradient descent (PGD). To make the paper self-contained, we include a sketch of the relevant equations, noting that they are probably known in other contexts. We detail the scheme on the Bi-word models, the Mono-word models can be handled analogously.

Constraints First of all we note that we are dealing with a *constrained* optimization problem, since the objective (3) is minimized over the parameters of probability distributions. For the dictionary parameters we have positivity constraints and normalization constraints:

$$p(f|e_1,e_2) \geq 0 \quad \forall f, e_1, e_2 \ ,$$

$$\sum_f p(f|e_1,e_2) = 1 \quad \forall e_1, e_2 \ .$$

This is known as a *product of simplices*, a relatively easy constraint system. For the Bi-1 there are no more parameters to optimize.

For the Bi-HMM (and also the Mono-HMM) there are the parameters $r_{-5}, \ldots r_5$ and r_L of the inter-alignment model. Each one comes with a positivity constraint. Moreover, these parameters are determined only up to scale, so we introduce the simplex constraint that they sum to 1: $\sum_{k=-5}^{5} r_k + r_L = 1$. The same principle applies to the parameters of the initial probability and of $p_{intra}(\cdot|\cdot)$.

5.1 Projected Gradient Descent

We first present a solution based on *projected gradient descent* (PGD) (Bertsekas, 1999, chap. 2), which is applicable since our constraint set is convex. Even though EM is usually the better suited method, we recommend reading this section as some auxiliary problems of EM are optimized by a very similar method.

PGD is similar to unconstrained gradient descent: one iteratively computes the gradient of the objective and takes a step in this direction. In general one will leave the feasible region, so one takes the closest feasible point instead. This operation is called *projection*. In our case we use the method of (Michelot, 1986). Finally, this point can have a higher energy than the previous, but the direction between the two points is a *descent direction*. We do a backtracking line-search to find a step in this direction that gives a sufficient decrease in the objective value. For the convex Bi-1 model this will eventually reach the global optimum, for the Bi-HMM a local optimum (as is standard for HMMs).

Obviously, the gradient of the regularity term (w.r.t. the dictionary parameters) is the weight vector with entries $w^f_{e_1,e_2}$. Further, the gradient of the standard maximum likelihood term is additive over the sentences. Hence, in the following we only state the gradients of a single sentence pair, i.e. $\frac{\partial}{\partial\theta} - \log(p(\mathbf{f}^s|\mathbf{e}^s))$, where θ is either a dictionary or an alignment parameter.

All considered models are so-called *multinomial* distributions. As shown in the appendix, for such distributions the gradient w.r.t. the dictionary parameters is given by

$$\frac{\partial}{\partial p(f|e_1,e_2)} - \log(p(\mathbf{f}^s|\mathbf{e}^s)) = -\frac{\sum_{\mathbf{a}} k_{\mathbf{a}}(f,e_1,e_2)\, p(\mathbf{a}|\mathbf{f}^s,\mathbf{e}^s)}{p(f|e_1,e_2)} \tag{4}$$

where $k_{\mathbf{a}}(f,e_1,e_2)$ is the number of times f aligns to both e_1 and e_2 in the alignment $\mathbf{a}$ and for the considered sentence pair. Note that the numerator of the ratio is the *expectation* of f aligning to e_1 and e_2 in the given sentence pair. This expression is also a fundamental building block of standard EM. For the Mono-1 and Bi-1 this is simply a sum over the source positions j. For the Mono- and Bi-HMM it can be calculated by the forward-backward algorithm (Baum et al., 1970).

With a similar argument one can derive the partial derivatives of the alignment parameters. We exemplarily detail this for $p_{\text{inter}\cdot}$. Let θ denote any of the parameters $p_0, p_1, r_{-5}, \ldots, r_5$ and r_L. Then one can show that

$$\frac{\partial}{\partial\theta} - \log(p(\mathbf{f}^s|\mathbf{e}^s)) = -\sum_{i,i'} \frac{\sum_{\mathbf{a}} k_{\mathbf{a}}(i|i',I_s)\, p(\mathbf{a}|\mathbf{f}^s,\mathbf{e}^s)}{p(i|i',I_s)} \cdot \frac{\partial p(i|i',I_s)}{\partial\theta}, \tag{5}$$

where $k_{\mathbf{a}}(i|i')$ denotes the number of times a source word is aligned to position i when the head of the previous source word was i'.

It remains to derive the partial derivatives of $p(i|i',I_s)$ w.r.t. the alignment parameters. For p_0 and p_1 this is straightforward. For a regular count r_k with $|k| \leq 5$ we have

$$\frac{\partial}{\partial r_k} p(i|i',I) = \begin{cases} p_1 \frac{\tau_{i',I} - r_k}{\tau^2_{i',I}} & \text{if } k = i - i' \\ p_1 \frac{-r_{i-i'}}{\tau^2_{i',I}} & \text{if } |i-i'| \leq 5 \\ p_1 \frac{-r_L}{\tau^2_{i',I} \cdot n^L_{i',I}} & \text{else}, \end{cases}$$

where $\tau_{i',I}$ is as in (2). The derivative w.r.t. r_L is

$$\frac{\partial}{\partial r_L} p(i|i',I) = \begin{cases} p_1 \frac{\tau_{i',I} - r_L}{\tau^2_{i',I} \cdot n^L_{i',I}} & \text{if } |i-i'| > 5 \\ p_1 \frac{-r_{i-i'}}{\tau^2_{i',I} \cdot n^L_{i',I}} & \text{else}, \end{cases}$$

with $n^L_{i',I} = |\{i'' : 1 \leq i'' \leq I, |i''-i'| > 5\}|$.

5.2 Expectation Maximization

A very commonly used method for word alignment is *expectation maximization* (Neal and Hinton, 1998). We give a modified version that handles our new objective function. Note that modifications of EM have been derived before, e.g. (Ganchev et al., 2010).

Traditionally, EM is used for standard maximum likelihood optimization. Denoting the parameters of the model as $\boldsymbol{\theta}$, the respective minimization problem would be

$$\min_{\boldsymbol{\theta}} \sum_{s=1}^{S} -\log(p(\mathbf{f}_s|\mathbf{e}_s,\boldsymbol{\theta})) \ .$$

The function to be minimized is called *negative log-likelihood*. It follows from (Neal and Hinton, 1998) that the function

$$F(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}) = \sum_{s=1}^{S} \sum_{\mathbf{a_s}} -p(\mathbf{a_s}|\mathbf{f}_s,\mathbf{e}_s,\tilde{\boldsymbol{\theta}}) \Big[\log(p(\mathbf{f}_s,\mathbf{a}_s|\mathbf{e}_s,\boldsymbol{\theta})) - \log(p(\mathbf{a_s}|\mathbf{f}_s,\mathbf{e}_s,\tilde{\boldsymbol{\theta}})) \Big]$$

is an upper bound on the negative log-likelihood function, independent of the choice of $\tilde{\boldsymbol{\theta}}$. In fact, $F(\boldsymbol{\theta},\boldsymbol{\theta})$ is exactly the negative log-likelihood for $\boldsymbol{\theta}$. As a consequence,

$$F(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}) + \sum_{\mathbf{e_1},\mathbf{e_2}} \sum_{f} w^f_{e_1,e_2}\, p(f\,|e_1,e_2) \tag{6}$$

upper bounds our new objective (3) - note that all $p(f|e_1,e_2)$ are entries in the vector $\boldsymbol{\theta}$. As in standard EM, we now perform coordinate descent on this new function: we iteratively update $\tilde{\boldsymbol{\theta}}$ to the vector that minimizes the objective for fixed $\boldsymbol{\theta}$. The optimal value is given as the expectation of alignments given $\boldsymbol{\theta}$ (Neal and Hinton, 1998), which is why this term is generally called E-step. The respective calculations in our case are exactly as the ones performed in gradient descent.

The second step in each iteration is called M-step and consists of setting $\boldsymbol{\theta}$ to the optimal value

for the given $\tilde{\boldsymbol{\theta}}$ and hence the given coefficients $p(\mathbf{a_s} \mid \mathbf{f}_s, \mathbf{e}_s, \tilde{\boldsymbol{\theta}})$. While for simple models there is often an analytic solution, in our case we are not aware of one for *any* of the parameters (except for special cases, e.g. when all $w^f_{e_1,e_2}$ are 0). Note, that this implies that the popular toolkit GIZA++ is not doing the M-step correctly: when applying the equations derived below, we verified that renormalizing expectations does *not* minimize the M-step energy. Moreover, with the common procedure the total energy usually only decreases in the first few iterations, after that it often increases.

The arising M-step decomposes into several independent optimization problems. In particular, there is a separate problem for each e_1, e_2 to update the respective dictionary distribution. The function to be minimized is

$$\sum_f \left[w^f_{e_1,e_2} p(f|e_1,e_2) - c^f_{e_1,e_2} \log(p(f|e_1,e_2)) \right] \ ,$$

where the weights c_{f,e_1,e_2} are the expectations (under the previous $\boldsymbol{\theta}$) of f aligning to e_1 and e_2. We solve this via gradient descent with the gradient

$$\frac{\partial}{\partial p(f|e_1,e_2)} = \sum_f \left[w^f_{e_1,e_2} - \frac{c^f_{e_1,e_2}}{p(f|e_1,e_2)} \right] \ ,$$

In special cases more efficient schemes are applicable. In particular it is well-known that if $w^f_{e_1,e_2} = 0$ for all f the optimal solution is given by re-normalizing the coefficients $c^f_{e_1,e_2}$. If $w^f_{e_1,e_2}$ is constant for all $f \neq NULL$, then in principle one only has to determine the probability $p(NULL|e_1,e_2)$. The remaining mass can again be spread according to normalized coefficients.

For the alignment parameters, we again only discuss $p_{\text{inter}}(\cdot|\cdot)$, where the auxiliary energy is

$$\sum_I \sum_{i,i'=1}^{I} -c^I_{i,i'} \log(p(i|i',I)) \ ,$$

and the gradient for an alignment parameter θ

$$\frac{\partial}{\partial \theta} = \sum_{i,i'=1}^{I} -\frac{c^I_{i,i'}}{p(i|i',I)} \frac{\partial p(i|i',I)}{\partial \theta} \ .$$

The inner derivatives were given in section 5.1. The parameters p_0 and p_1 are very simple to derive.

6 Experiments

We report results on three different data sets, in both directions each. The first two are Europarl sets (in the original casing), where we consider English-German[2] and English-Spanish[3]. Further, we consider the well-known Canadian Hansards task (French-English, lowercased). In all cases we report weighted f-measures (Fraser and Marcu, 2007b) on the publicly available gold alignments. We use a weighting factor of $\alpha = 0.1$, which performed well in Fraser and Marcu's work.

	EP De-En	**EP** Es-En	**Hs.** Fr-En
#sentences (large task)	500K	500K	1M
#sentences (small task)	15K	15K	25K
sent. length	80	75	40

Table 1: Statistics of the considered tasks. Es = Spanish, De = German, Fr = French, En = English, Hs. = Canadian Hansards, EP = Europarl. "K" denotes a thousand, "M" a million.

For the Mono-word models we consider large scale tasks with at least 500000 sentence pairs. For the Bi-word models the demand on computational resources is much higher, so we use tasks with 15000 to 25000 sentence pairs. We also evaluate the Mono-word models here, showing that the regularity term becomes more important in the case of scarce training data.

The most important statistics of all tasks are listed in Table 1. The methods required no more than 4 GB memory on these tasks. The running times on the large scale tasks sometimes slightly exceeded a day. For the small scale tasks even the Bi-word models need less than 12 hours. Without regularity, EM is clearly faster. But with regularity terms, EM and PGD are roughly equal in speed. In general, PGD finds a slightly higher energy than EM.

6.1 Comparison of Models

In this paper we have introduced new objective functions and argued that they alleviate some of the deficiencies of standard maximum likelihood for conditional models. As a consequence, we are interested in comparing models and objective functions, and not so much in getting the last bit of practical performance (f-measure).

Hence, when comparing[4] to GIZA++ we turn

[2] Gold alignments available at `http://www.maths.lth.se/matematiklth/personal/tosch/download.html`.

[3] Gold alignments from (Lambert et al., 2005).

[4] It is common to run GIZA++ with smoothing and only 5 iterations. Indeed, this improves the f-measures. However,

	EUParl Es-En 500K		EUParl De-En 500K		CHans Fr-En 1M	
	Es|En	**En|Es**	**De|En**	**En|De**	**Fr|En**	**En|Fr**
IBM-1, EM, no reg.	64.0	64.6	68.5	71.5	82.7	83.3
IBM-1, EM, with reg.	64.5	64.9	68.6	72.0	83.1	83.5
IBM-1, PGD, no reg.	63.5	63.6	67.0	71.0	82.6	82.3
IBM-1, PGD, with reg.	63.8	64.1	66.9	71.8	83.3	81.8
HMM, EM (GIZA++)	75.0	74.2	72.5	75.3	91.4	90.8
HMM, EM (our), no reg.	77.4	76.1	73.2	77.8	89.6	90.3
HMM, EM (our), with reg.	77.7	76.3	73.1	78.2	90.3	90.6
HMM, PGD, no reg.	75.3	73.5	70.9	75.3	89.2	88.8
HMM, PGD, with reg.	74.9	73.8	72.2	75.7	89.3	88.7
IBM-4 (GIZA++)	79.6	80.0	76.8	80.5	92.3	93.2

Table 2: F-measures ($\times 100$) for the large-scale tasks.

off smoothing. Also, we run more iterations than usual: for all methods (GIZA++, EM, PGD) we run 30 iterations of IBM-1, followed by 50 for the HMM. Here we use the same regularity terms for both models. For reference, we also evaluate the IBM-4 as implemented in GIZA++ (starting from the 50 HMM iterations, then doing 5 iterations of IBM-3 and 5 iterations of IBM-4).

For the Bi-word models we initialize the non-convex Bi-HMM by running the Bi-1 first (with the same regularity term, if any). The number of iterations is the same as for the respective Mono-word models.

Large Scale Tasks. In Table 2 we show the resulting f-measures on the large scale tasks for all mentioned strategies, including GIZA++'s HMM and IBM-4. Often our HMM outperforms GIZA++ (without smoothing), which may be due to the more precise M-step. Moreover, the regularity terms usually improve the results, where the effect is generally stronger the higher inflected the source language is. Still, the IBM-4 performs best everywhere, so in future work we will transfer our new objective to this model.

Small Scale Tasks. The results for the small tasks are given in Table 3. Here it can be seen that adding the regularity to the Mono-word models greatly improves on the f-measures of the baseline HMM and sometimes gets close to the IBM-4. For the Bi-word models the regularity terms also help greatly, and in the majority of cases beat the baseline Mono-HMM (without regularity).

Like for the large scale tasks, EM performs better than PGD and the corrected M-steps often beat GIZA++.

with the new objective function we also get better results for less iterations. A systematic comparison of this is left for future work.

Method	BLEU	TER
our HMM, no reg.	27.94	56.98
our HMM, with reg.	28.33	56.20
GIZA++, HMM	28.04	56.83
GIZA++, IBM-4	28.71	56.15

Table 4: Evaluation of the translation quality for the large scale German $\rightarrow$ English task.

Method	BLEU	TER
our HMM, no reg.	21.50	63.44
our HMM, with reg.	21.77	62.97
GIZA++, HMM	21.90	63.34
GIZA++, IBM-4	22.24	62.81
Bi-HMM, no reg.	21.78	63.58
Bi-HMM, with reg.	21.70	63.38

Table 5: Evaluation of the translation quality for the small scale German $\rightarrow$ English task.

6.2 Effect on Phrase-based Translation

We give a first evaluation of the effect of our alignments on phrase-based translation, where we ran MOSES with a 5-gram language model. We randomly picked translation from German to English with 750 unseen development and 3000 unseen test sentences.

As shown in the tables 4 and 5 the regularity terms do improve translation for Mono-word models. The Bi-word models are presently not competetive. Here we are showing the BLEU accuracy measure and the Translation Edit Rate (TER).

7 Conclusion

This paper has introduced the idea of regularizing the mass of the probability parameters that is

	EUParl Es-En 15K		EUParl De-En 15K		CHans Fr-En 25K	
	Es|En	**En|Es**	**De|En**	**En|De**	**Fr|En**	**En|Fr**
IBM-1, EM, no reg.	54.5	56.0	59.0	63.6	77.1	79.2
IBM-1, EM, with reg.	56.0	57.2	60.4	64.5	78.7	79.5
IBM-1, PGD, no reg.	50.6	56.0	59.2	63.1	76.6	79.3
IBM-1, PGD, with reg.	55.8	56.9	60.0	63.5	78.0	79.4
HMM, EM, (GIZA++)	68.0	66.9	65.7	67.9	82.8	84.9
HMM, EM, (our), no reg.	69.3	68.9	65.0	70.2	80.9	86.9
HMM, EM, (our), with reg.	72.2	72.0	68.0	71.8	83.5	87.7
HMM, PGD, no reg.	68.0	68.9	57.9	68.7	79.3	85.6
HMM, PGD, with reg.	68.0	71.0	61.0	69.4	82.1	86.1
IBM-4 (GIZA++)	72.5	72.3	76.8	73.0	86.4	89.0
Bi-1, EM, no reg.	52.0	54.7	57.8	63.8	74.2	77.3
Bi-1, EM, with reg.	54.2	55.8	59.3	64.3	76.8	78.8
Bi-1, PGD, no reg.	52.8	55.2	58.0	64.0	75.0	77.8
Bi-1, PGD, with reg.	53.7	56.0	59.1	63.8	75.7	77.8
Bi-HMM, EM, no reg.	66.2	68.5	64.4	67.2	79.6	82.2
Bi-HMM, EM, with reg.	70.8	71.2	66.0	70.8	80.5	86.8
Bi-HMM, PGD, no reg.	68.4	68.2	71.5	68.0	78.2	84.3
Bi-HMM, PGD, with reg.	69.8	71.2	63.5	68.1	78.4	84.1

Table 3: Resulting F-measures ($\times 100$) for the small scale tasks.

used to explain the data. We have argued that these terms reduce overfitting and demonstrated experimentally that the introduced objectives improve the f-measures of the generated alignments. We often beat the baseline HMM, and transferring our objective to the IBM-4 would probably beat a baseline IBM-4.

Our comparison of projected gradient descent (PGD) and expectation maximization (EM) revealed that EM leads to better alignments, although PGD finds a comparable but slightly higher objective value. We also showed that parametric HMMs induce non-trivial M-steps.

In future work we want to address IBM-3 and IBM-4 and explore the effect on phrase-based translation in greater detail.

To facilitate further research in this area, the source code associated to this work is integrated into a tool called **RegAligner**, publicly available at the author's homepage and at `https://github.com/Thomas1205/RegAligner`.

Acknowledgments The author thanks Ben Taskar and João Graça for helpful discussions, as well as Alexander Engau and UC Denver for helping out with computational resources after the author had left Lund University. This work was in large part funded by the European Research Council (GlobalVision grant no. 209480).

Appendix

We now derive the partial derivative of the negative log-likelihood of a general (multinomial) probability w.r.t. a dictionary parameter $p(f|e_1, e_2)$. This derivation is probably not novel, but included here for completeness. The partial derivative is given by

$$\frac{\partial}{\partial p(f|e_1,e_2)} - \log(p(\mathbf{f}^s|\mathbf{e}^s)) = -\frac{1}{p(\mathbf{f}^s|\mathbf{e}^s)} \cdot \left[\sum_{\mathbf{a}^s} \frac{\partial}{\partial p(f|e_1,e_2)} p(\mathbf{f}^s, \mathbf{a}^s|\mathbf{e}^s) \right] .$$

Now take a fixed $\mathbf{a}$, and denote $k_{\mathbf{a}} \in \mathbb{N}_0$ the number of times the factor $p(f|e_1, e_2)$ is used in its probability, i.e.

$$p(\mathbf{f}^s, \mathbf{a}|\mathbf{e}^s) = c_{\mathbf{a}} \cdot p(f|e_1, e_2)^{k_{\mathbf{a}}} \ ,$$

where $c_{\mathbf{a}}$ is constant w.r.t. $p(f|e_1, e_2)$. Clearly

$$\begin{aligned} \frac{\partial p(\mathbf{f}^s, \mathbf{a}|\mathbf{e}^s)}{\partial p(f|e_1,e_2)} &= c_{\mathbf{a}} \cdot k_{\mathbf{a}} \cdot p(f|e_1, e_2)^{k_{\mathbf{a}}-1} \\ &= k_{\mathbf{a}} \frac{p(\mathbf{f}^s, \mathbf{a}|\mathbf{e}^s)}{p(f|e_1, e_2)} . \end{aligned}$$

This is how the claimed formula arises, i.e. the entire derivative is

$$-\frac{\sum_{\mathbf{a}} k_{\mathbf{a}}\, p(\mathbf{a}|\mathbf{f}^s, \mathbf{e}^s)}{p(f|e_1, e_2)} .$$

References

Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, I. D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky. 1999. Statistical machine translation, Final report, JHU workshop. `http://www.clsp.jhu.edu/ws99/`.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. 1970. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1):164–171.

T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein. 2010. Painless unsupervised learning with features. In *Conference of the North American Chapter of the Association for Computational Linguistisc (NAACL)*, Los Angeles, California, June.

D.P. Bertsekas. 1999. *Nonlinear Programming,* 2*nd edition*. Athena Scientific.

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

J. DeNero, D. Gillick, J. Zhang, and D. Klein. 2006. Why generative phrase models underperform surface heuristics. In *StatMT '06: Proceedings of the Workshop on Statistical Machine Translation*, pages 31–38, Morristown, NJ, USA, June.

Y. Deng and W. Byrne. 2005. HMM word and phrase alignment for statistical machine translation. In *HLT-EMNLP*, Vancouver, Canada, October.

A. Fraser and D. Marcu. 2007a. Getting the structure right for word alignment: LEAF. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, June.

A. Fraser and D. Marcu. 2007b. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303, September.

K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049, July.

J. Graça, K. Ganchev, and B. Taskar. 2010. Learning tractable word alignment models with complex constraints. *Computational Linguistics*, 36, September.

S. Lacoste-Julien, B. Taskar, D. Klein, and M. Jordan. 2006. Word alignment via quadratic assignment. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, New York, New York, June.

P. Lambert, A.D. Gispert, R. Banchs, and J.B. Marino. 2005. Guidelines for word alignment evaluation and manual alignment. *Language Resources and Evaluation*, 39(4):267–285.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, New York, New York, June.

G.S. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via Expectation Regularization. In *International Conference on Machine Learning*, Corvallis, Oregon.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, Pennsylvania, July.

E. Matusov, R. Zens, and H. Ney. 2004. Symmetric word alignments for statistical machine translation. In *International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.

A. Mauser, S. Hasan, and H. Ney. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, August.

D. Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

C. Michelot. 1986. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal on Optimization Theory and Applications*, 50(1), July.

R.M. Neal and G.E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*. MIT press.

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

T. Schoenemann. 2011. Probabilistic word alignment under the l_0-norm. In *Conference on Computational Natural Language Learning (CoNLL)*, Portland, Oregon, June.

E. Sumita, Y. Akiba, T. Doi, A. Finch, K. Imamura, H. Okuma, M. Paul, M. Shimohata, and T. Watanabe. 2004. EBMT, SMT, Hybrid and more: ATR spoken language translation system. In *International Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, September.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vancouver, Canada, October.

K. Toutanova, H.T. Ilhan, and C.D. Manning. 2002. Extensions to HMM-based statistical word alignment models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, Pennsylvania, July.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *International Conference on Computational Linguistics (COLING)*, pages 836–841, Copenhagen, Denmark, August.

Y.-Y. Wang and A. Waibel. 1998. Modeling with structures in statistical machine translation. In *International Conference on Computational Linguistics (COLING)*, Montreal, Canada, August.

R. Zens, E. Matusov, and H. Ney. 2004. Improved word alignment using a symmetric lexicon model. In *International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, August.