Training Nondeficient Variants of IBM-3 and IBM-4 for Word Alignment

Thomas Schoenemann
Heinrich-Heine-Universität Düsseldorf, Germany
Universitätsstr. 1
40225 Düsseldorf, Germany

Abstract

We derive variants of the fertility based models IBM-3 and IBM-4 that, while maintaining their zero and first order parameters, are nondeficient. Subsequently, we proceed to derive a method to compute a likely alignment and its neighbors as well as give a solution of EM training. The arising M-step energies are non-trivial and handled via projected gradient ascent.

Our evaluation on gold alignments shows substantial improvements (in weighted F-measure) for the IBM-3. For the IBM-4 there are no consistent improvements. Training the nondeficient IBM-5 in the regular way gives surprisingly good results.

Using the resulting alignments for phrase-based translation systems offers no clear insights w.r.t. BLEU scores.

1 Introduction

While most people think of the translation and word alignment models IBM-3 and IBM-4 as inherently deficient models (i.e. models that assign non-zero probability mass to impossible events), in this paper we derive nondeficient variants maintaining their zero order (IBM-3) and first order (IBM-4) parameters. This is possible as IBM-3 and IBM-4 are very special cases of general log-linear models: they are properly derived by the chain rule of probabilities. Deficiency is only introduced by ignoring a part of the history to be conditioned on in the individual factors of the chain rule factorization. While at first glance this seems necessary to obtain zero and first order dependencies, we show that with proper renormalization all factors can be made nondeficient.

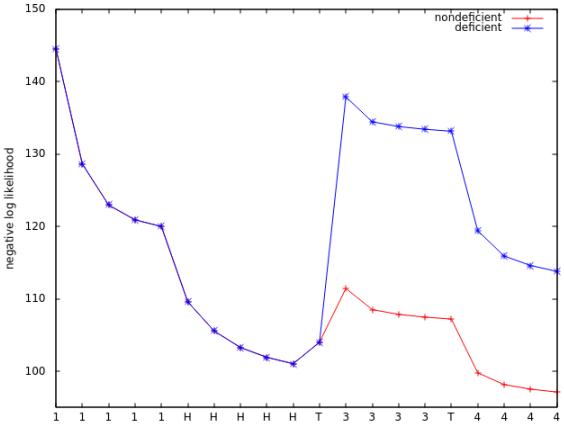

Figure 1: Plot of the negative log. likelihoods (the quantity to be minimized) arising in training deficient and nondeficient models (for Europarl German | English, training scheme $1^5\mathrm{H}^5 3^5 4^5$). 1/3/4=IBM-1/3/4, H=HMM, T=Transfer iteration. The curves are identical up to iteration 11.

Iteration 11 shows that merely 5.14% of the (HMM) probability mass are covered by the Viterbi alignment and its neighbors. With deficient models (and deficient empty words) the final negative log likelihood is higher than the initial HMM one, with nondeficient models it is lower than for the HMM, as it should be for a better model.

Having introduced the model variants, we proceed to derive a hillclimbing method to compute a likely alignment (ideally the Viterbi alignment) and its neighbors. As for the deficient models, this plays an important role in the E-step of the subsequently derived expectation maximization (EM) training scheme. As usual, expectations in EM are approximated, but we now also get non-trivial M-step energies. We deal with these via projected gradient ascent.

The downside of our method is its resource consumption, but still we present results on corpora with 100.000 sentence pairs. The source code of this project is available in our word alignment software RegAligner[1], version 1.2 and later.

Figure 1 gives a first demonstration of how much the proposed variants differ from the standard models by visualizing the resulting negative log likelihoods[2], the quantity to be minimized in EM-training. The nondeficient IBM-4 derives a lower negative log likelihood than the HMM, the regular deficient variant only a lower one than the IBM-1. As an aside, the transfer iteration from HMM to IBM3 (iteration 11) reveals that only 5.14% of the probability mass[3] are preserved when using the Viterbi alignment and its neighbors instead of all alignments.

Indeed, it is widely recognized that – with proper initialization – fertility based models outperform sequence based ones. In particular, sequence based models can simply ignore a part of the sentence to be conditioned on, while fertility based models explicitly factor in a probability of words in this sentence to have no aligned words (or any other number of aligned words, called the *fertility*). Hence, it is encouraging to see that the nondeficient IBM-4 indeed derives a higher likelihood than the sequence based HMM.

Related Work Today's most widely used models for word alignment are still the models IBM 1-5 of Brown et al. (1993) and the HMM of Vogel et al. (1996), thoroughly evaluated in (Och and Ney, 2003). While it is known that fertility-based models outperform sequence-based ones, the large bulk of word alignment literature following these publications has mostly ignored fertility-based models. This is different in the present paper which deals exclusively with such models.

One reason for the lack of interest is surely that computing expectations and Viterbi alignments for these models is a hard problem (Udupa and Maji, 2006). Nevertheless, computing Viterbi alignments for the IBM-3 has been shown to often be practicable (Ravi and Knight, 2010; Schoenemann, 2010).

Much work has been spent on HMM-based formulations, focusing on the computationally tractable side (Toutanova et al., 2002; Sumita et al., 2004; Deng and Byrne, 2005). In addition, some rather complex models have been proposed that usually aim to replace the fertility based models (Wang and Waibel, 1998; Fraser and Marcu, 2007a).

Another line of models (Melamed, 2000; Marcu and Wong, 2002; Cromières and Kurohashi, 2009) focuses on joint probabilities to get around the garbage collection effect (i.e. that for conditional models, rare words in the given language align to too many words in the predicted language). The downside is that these models are computationally harder to handle.

A more recent line of work introduces various forms of regularity terms, often in the form of symmetrization (Liang et al., 2006; Graça et al., 2010; Bansal et al., 2011) and recently by using L_0 norms (Vaswani et al., 2012).

2 The models IBM-3, IBM-4 and IBM-5

We begin with a short review of fertility-based models in general and IBM-3, IBM-4 and IBM-5 specifically. All are due to (Brown et al., 1993) who proposed to use the deficient models IBM-3 and IBM-4 to initialize the nondeficient IBM-5.

For a foreign sentence $\mathbf{f} = f_1^J = (f_1, \ldots, f_J)$ with J words and an English one $\mathbf{e} = e_1^I = (e_1, \ldots, e_I)$ with I words, the (conditional) probability $p(f_1^J|e_1^I)$ of getting the foreign sentence as a translation of the English one is modeled by introducing the word alignment $\mathbf{a}$ as a hidden variable:

$$p(f_1^J|e_1^I) = \sum_{\mathbf{a}} p(f_1^J, \mathbf{a}|e_1^I)$$

All IBM models restrict the space of alignments to those where a foreign word can align to at most one target word. The resulting alignment is then written as a vector a_1^J, where each a_j takes integral values between 0 and I, with 0 indicating that f_j has no English correspondence.

The fertility-based models IBM-3, IBM-4 and IBM-5 factor the (conditional) probability $p(f_1^J, a_1^J|e_1^I)$ of obtaining an alignment and a translation given an English sentence according to the following generative story:

[1] `https://github.com/Thomas1205/RegAligner`, for the reported results we used a slightly earlier version.

[2] Note that the figure slightly favors IBM-1 and HMM as for them the length J of the foreign sequence is assumed to be known whereas IBM-3 and IBM-4 explicitly predict it.

[3] This number regards the corpus probability as in (9) to the power of $1/S$, i.e. the objective function in maximum likelihood training. The number is not entirely fair as alignments where more than half the words align to the empty word are assigned a probability of 0. Still, this is an issue only for short sentences.

1. For $i = 1, 2, \ldots, I$, decide on the number Φ_i of foreign words aligned to e_i. This number is called the *fertility* of e_i. Choose with probability $p(\Phi_i|e_1^I, \Phi_1^{i-1}) = p(\Phi_i|e_i)$.

2. Choose the number Φ_0 of unaligned words in the (still unknown) foreign sequence. Choose with probability $p(\Phi_0|e_1^I, \Phi_1^I) = p(\Phi_0|\sum_{i=1}^I \Phi_i)$. Since each foreign word belongs to exactly one English position (including 0), the foreign sequence is now known to be of length $J = \sum_{i=0}^I \Phi_i$.

3. For each $i = 1, 2, \ldots, I$, and $k = 1, \ldots, \Phi_i$ decide on
   (a) the identity $f_{i,k}$ of the next foreign word aligned to e_i. Choose with probability $p(f_{i,k}|e_1^I, \Phi_0^I, \mathbf{d}_1^{i-1}, d_{i,1}, \ldots, d_{i,k-1}, \mathbf{f}_{i,k}) = p(f_{i,k}|e_i)$, where $\mathbf{d}_i$ comprises all $d_{i,k}$ for word i (see point b) below) and $\mathbf{f}_{i,k}$ comprises all foreign words known at that point.
   (b) the position $d_{i,k}$ of the just generated foreign word $f_{i,k}$, with probability $p(d_{i,k}|e_1^I, \Phi_0^I, \mathbf{d}_1^{i-1}, d_{i,1}, \ldots, d_{i,k-1}, \mathbf{f}_{i,k}, f_{i,k}) = p(d_{i,k}|e_i, \mathbf{d}_1^{i-1}, d_{i,1}, \ldots, d_{i,k-1}, f_{i,k}, J)$.

4. The remaining Φ_0 open positions in the foreign sequence align to position 0. Decide on the corresponding foreign words with $p(f_{d_{0,k}}|e_0)$, where e_0 is an artificial "empty word".

To model the probability for the number of unaligned words in step 2, each of the $\sum_{i=1}^I \Phi_i$ properly aligned foreign words generates an unaligned foreign word with probability p_0, resulting in

$$p\Big(\Phi_0 \Big| \sum_{i=1}^I \Phi_i\Big) = \binom{\sum_{i=1}^I \Phi_i}{\Phi_0} p_0^{\Phi_i} (1-p_0)^{(\sum_i \Phi_i)-\Phi_0},$$

with a base probability p_0 and the combinatorial coefficients $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, where $n! = \prod_{k=1}^n k$ denotes the factorial of n. The main difference between IBM-3, IBM-4 and IBM-5 is the choice of probability model in step 3 b), called a *distortion model*. The choices are now detailed.

2.1 IBM-3

The IBM-3 implements a zero order distortion model, resulting in

$$p(d_{i,k}|i, J) \quad .$$

Since most of the context to be conditioned on is ignored, this allows invalid configurations to occur with non-zero probability: some foreign positions can be chosen several times, while others remain empty. One says that the model is *deficient*. On the other hand, the model for $p(\Phi_0|\sum_{i=1}^I \Phi_i)$ is nondeficient, and in training this often results in very high probabilities p_0. To prevent this it is common to make this model deficient as well (Och and Ney, 2003), which improves performance immensely and gives much better results than simply fixing p_0 in the original model.

As for each i the $d_{i,k}$ can appear in any order (i.e. need not be in ascending order), there are $\prod_{i=1}^I \Phi_i!$ ways to generate the same alignment a_1^J (where the Φ_i are the fertilities induced by a_1^J). In total, the IBM-3 has the following probability model:

$$p(f_1^J, a_1^J|e_1^I) = \prod_{j=1}^J \Big[p(f_j|e_{a_j}) \cdot p(j|a_j, J)\Big] \quad (1)$$
$$\cdot\, p\Big(\Phi_0|\sum_{i=1}^I \Phi_i\Big) \cdot \prod_{i=1}^I \Phi_i!\, p(\Phi_i|e_i)\ .$$

Reducing the Number of Parameters While using non-parametric models $p(j|i, J)$ is convenient for closed-form M-steps in EM training, these parameters are not very intuitive. Instead, in this paper we use the *parametric* model

$$p(j|i, J) = \frac{p(j|i)}{\sum_{j=1}^J p(j|i)} \quad (2)$$

with the more intuitive parameters $p(j|i)$. The arising M-step energy is addressed by projected gradient ascent (see below).

These parameters are also used for the nondeficient variants. Using the original non-parametric ones can be handled in a very similar manner to the methods set forth below.

2.2 IBM-4

The distortion model of the IBM-4 is a first order one that generates the $d_{i,k}$ of each English position i in ascending order (i.e. for $1 < k \leq \Phi_i$ we have $d_{i,k} > d_{i,k-1}$). There is then a one-to-one correspondence between alignments a_1^J and (valid) distortion parameters $(d_{i,k})_{i=1,\ldots,I,\, k=1,\ldots,\Phi_i}$ and therefore no longer a factor of $\prod_{i=1}^I \Phi_i!$.

The IBM-4 has two sub-distortion models, one for the first aligned word ($k = 1$) of an English position and one for all following words ($k > 1$, only

if $\Phi_i > 1$). For position i, let $[i] = \arg\max\{i' | 1 \leq i' < i, \Phi_{i'} > 0\}$ denote[4] the closest preceding English word that has aligned foreign words. The aligned foreign positions of $[i]$ are combined into a *center position* $\odot_{[i]}$, the rounded average of the positions. Now, the distortion probability for the first word ($k = 1$) is

$$p_{=1}(d_{i,1} | \odot_{[i]}, \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]}), J) \ ,$$

where $\mathcal{A}$ gives the word class of a foreign word and $\mathcal{B}$ the word class of an English word (there are typically 50 classes per language, derived by machine learning techniques). The probability is further reduced to a dependency on the difference of the positions, i.e. $p_{=1}(d_{i,1} - \odot_{[i]} \,|\, \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]}))$. For $k > 1$ the model is

$$p_{>1}(d_{i,k} | d_{i,k-1}, \mathcal{A}(f_{i,k}), J) \ ,$$

which is likewise reduced to $p_{>1}(d_{i,k} - d_{i,k-1} \,|\, \mathcal{A}(f_{i,k}))$. Note that in both difference-based formulations the dependence on J has to be dropped to get closed-form solutions of the M-step in EM training, and Brown et al. note themselves that the IBM-4 can place words before the start and after the end of the sentence.

Reducing Deficiency In this paper, we also investigate the effect of reducing the amount of wasted probability mass by enforcing the dependence on J by proper renormalization, i.e. using

$$p_{=1}(j | j', \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]}), J) = \frac{p_{=1}(j - j' | \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]}))}{\sum_{j''=1}^{J} p_{=1}(j'' - j' | \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]}))} \ , \quad (3)$$

for the first aligned word and

$$p_{>1}(j | j', \mathcal{A}(f_{i,k}), J) = \frac{p_{>1}(j - j' \,|\, \mathcal{A}(f_{i,k}))}{\sum_{j''=1}^{J} p_{>1}(j'' - j' \,|\, \mathcal{A}(f_{i,k}))} \quad (4)$$

for all following words, again handling the M-step in EM training via projected gradient ascent. With this strategy words can no longer be placed outside the sentence, but a lot of probability mass is still wasted on configurations where at least one foreign (or predicted) position j aligns to two or more positions i, i' in the English (or given) language (and consequently there are more unaligned source words than the generated Φ_0). Therefore, here, too, the probability for Φ_0 has to be made deficient to get good performance.

In summary, the base model for the IBM-4 is:

$$\begin{aligned} p(f_1^J, a_1^J | e_1^I) \ = \ & p\Big(\Phi_0 | \sum_{i=1}^{I} \Phi_i\Big) \quad (5) \\ & \cdot \prod_{j=1}^{J} p(f_j | e_{a_j}) \cdot \prod_{i=1}^{I} p(\Phi_i | e_i) \\ & \cdot \prod_{i:\Phi_i > 0} \Big[p_{=1}(d_{i,1} - \odot_{[i]} | \mathcal{A}(f_{i,1}), \mathcal{B}(e_{[i]})) \\ & \quad \cdot \prod_{k=2}^{\Phi_i} p_{>1}(d_{i,k} - d_{i,k-1} | \mathcal{A}(f_{i,k})) \Big] \ , \end{aligned}$$

where empty products are understood to be 1.

2.3 IBM-5

We note in passing that the distortion model of the IBM-5 is nondeficient and has parameters for filling the nth open gap in the foreign sequence given that there are N positions to choose from – see the next section for exactly what positions one can choose from. There is also a dependence on word classes for the foreign language.

This is neither a zero order nor a first order dependence, and in (Och and Ney, 2003) the first order model of the IBM-4, though deficient, outperformed the IBM-5. The IBM-5 is therefore rarely used in practice. This motivated us to instead reformulate IBM-3 and IBM-4 as nondeficient models. In our results, however, the IBM-5 gave surprisingly good results and was often superior to all variants of the IBM-4.

3 Nondeficient Variants of IBM-3 and IBM-4

From now on we always enforce that for each position i the indices $d_{i,k}$ are generated in ascending order ($d_{i,k} > d_{i,k-1}$ for $k > 1$). A central concept for the generation of $d_{i,k}$ in step 3(b) is the set of positions in the foreign sequence that are still without alignment. We denote the set of these positions by

$$\begin{aligned} \mathcal{J}_{i,k,J} \ = \ & \{1, \ldots, J\} \ - \ \{d_{i,k'} \,|\, 1 \leq k' < k\} \\ & - \{d_{i',k'} \,|\, 1 \leq i' < i, 1 \leq k' \leq \Phi_{i'}\} \end{aligned}$$

where the dependence on the various $d_{i',k'}$ is not made explicit in the following.

It is tempting to think that in a nondeficient model all members of $\mathcal{J}_{i,k,J}$ can be chosen for

[4] If the set is empty, instead a sentence start probability is used. Note that we differ slightly in notation compared to (Brown et al., 1993).

$d_{i,k}$, but this holds only $\Phi_i = 1$. Otherwise, the requirement of generating the $d_{i,k}$ in ascending order prevents us from choosing the $(\Phi_i - k)$ *largest* entries in $\mathcal{J}_{i,k,J}$. For $k > 1$ we also have to remove all positions smaller than $d_{i,k-1}$.

Let $\mathcal{J}_{i,k,J}^{\Phi_i}$ denote the set where these positions have been removed. With that, we can state the nondeficient variants of IBM-3 and IBM-4.

3.1 Nondeficient IBM-3

For the IBM-3, we define the auxiliary quantity

$$q(d_{i,k} = j \mid i, \mathcal{J}_{i,k,J}^{\Phi_i}) = \begin{cases} p(j|i) & \text{if } j \in \mathcal{J}_{i,k,J}^{\Phi_i} \\ 0 & \text{else} , \end{cases}$$

where we use the zero order parameters $p(j|i)$ we also use for the standard (deficient) IBM-3, compare (2). To get a nondeficient variant, it remains to renormalize, resulting in

$$p(d_{i,k} = j|i, \mathcal{J}_{i,k,J}^{\Phi_i}) = \frac{q(j|i, \mathcal{J}_{i,k,J}^{\Phi_i})}{\sum_{j=1}^{J} q(j|i, \mathcal{J}_{i,k,J}^{\Phi_i})} . \quad (6)$$

Further, note that the factors $\Phi_i!$ now have to be removed from (1) as the $d_{i,k}$ are generated in ascending order. Lastly, here we use the original nondeficient empty word model $p(\Phi_0 | \sum_{i=1}^{I} \Phi_i)$, resulting in a totally nondeficient model.

3.2 Nondeficient IBM-4

With the notation set up, it is rather straightforward to derive a nondeficient variant of the IBM-4. Here, there are the two cases $k = 1$ and $k > 1$. We begin with the case $k = 1$. Abbreviating $\alpha = \mathcal{A}(f_{i,1})$ and $\beta = \mathcal{B}(e_{[i]})$, we define the auxiliary quantity

$$q_{=1}(d_{i,1} = j|\odot_{[i]}, \alpha, \beta, \mathcal{J}_{i,k,J}^{\Phi_i}) = \quad (7)$$
$$\begin{cases} p_{=1}(j - \odot_{[i]}|\alpha, \beta) & \text{if } j \in \mathcal{J}_{i,k,J}^{\Phi_i} \\ 0 & \text{else} , \end{cases}$$

again using the - now first order - parameters of the base model. The nondeficient distribution $p_{=1}(d_{i,1} = j|\odot_{[i]}, \alpha, \beta, \mathcal{J}_{i,k,J}^{\Phi_i})$ is again obtained by renormalization.

For the case $k > 1$, we abbreviate $\alpha = \mathcal{A}(f_{i,k})$ and introduce the auxiliary quantity

$$q_{>1}(d_{i,k} = j|d_{i,k-1}, \alpha, \mathcal{J}_{i,k,J}^{\Phi_i}) = \quad (8)$$
$$\begin{cases} p_{>1}(j - d_{i,k-1}|\alpha) & \text{if } j \in \mathcal{J}_{i,k,J}^{\Phi_i} \\ 0 & \text{else} , \end{cases}$$

from which the nondeficient distribution $p_{>1}(d_{i,k}{=}j|d_{i,k-1}, \alpha, \mathcal{J}_{i,k,J}^{\Phi_i})$ is again obtained by renormalization.

4 Training the New Variants

For the task of word alignment, we infer the parameters of the models using the maximum likelihood criterion

$$\max_{\theta} \prod_{s=1}^{S} p_\theta(\mathbf{f}_s|\mathbf{e}_s) \quad (9)$$

on a set of training data (i.e. sentence pairs $s = 1, \ldots, S$). Here, θ comprises all base parameters of the respective model (e.g. for the IBM-3 all $p(f|e)$, all $p(\Phi, e)$ and all $p(j|i)$) and p_θ signifies the dependence of the model on the parameters. Note that (9) is truly a *constrained* optimization problem as the parameters θ have to satisfy a number of probability normalization constraints.

When $p_\theta(\cdot)$ denotes a fertility based model the resulting problem is a non-concave maximization problem with many local minima and no (known) closed-form solutions. Hence, it is handled by computational methods, which typically apply the logarithm to the above function.

Our method of choice to attack the maximum likelihood problem is expectation maximization (EM), the standard in the field, which we explain below. Due to non-concaveness the starting point for EM is of extreme importance. As is common, we first train an IBM-1 and then an HMM before proceeding to the IBM-3 and finally the IBM-4.

As in the training of the deficient IBM-3 and IBM-4 models, we approximate the expectations in the E-step by a set of likely alignments, ideally centered around the Viterbi alignment, but already for the regular deficient variants computing it is NP-hard (Udupa and Maji, 2006). A first task is therefore to compute such a set. This task is also needed for the actual task of word alignment (annotating a given sentence pair with an alignment).

4.1 Alignment Computation

For computing alignments, we use the common procedure of hillclimbing where we start with an alignment, then iteratively compute the probabilities of all alignments differing by a move or a swap (Brown et al., 1993) and move to the best of these if it beats the current alignment.

Since we cannot ignore parts of the history and still get a nondeficient model, computing the probabilities of the neighbors cannot be handled incrementally (or rather only partially, for the dictionary and fertility models). While this does increase running times, in practice the M-steps take longer than the E-steps.

For self-containment, we recall here that for an alignment a_1^J applying the **move** $a_1^J[j \to i]$ results in the alignment $\hat{a}_1^J$ defined by $\hat{a}_j = i$ and $\hat{a}_{j'} = a_{j'}$ for $j' \neq j$. Applying the **swap** $a_1^J[j_1 \leftrightarrow j_2]$ results in the alignment $\hat{a}_1^J$ defined by $\hat{a}_{j_1} = a_{j_2}$, $\hat{a}_{j_2} = a_{j_1}$ and $\hat{a}_{j'} = a_{j'}$ elsewhere. If a_1^J is the alignment produced by hillclimbing, the **move matrix** $m \in \mathbb{R}^{J \times I+1}$ is defined by $m_{j,i}$ being the probability of $a_1^J[j \to i]$ as long as $a_j \neq i$, otherwise 0. Likewise the **swap matrix** $s \in \mathbb{R}^{J \times J}$ is defined as s_{j_1,j_2} being the probability of $a_1^J[j_1 \leftrightarrow j_2]$ for $a_{j_1} \neq a_{j_2}$, 0 otherwise. The move and swap matrices are used to approximate expectations in EM training (see below).

4.2 Parameter Update

Naive Scheme It is tempting to account for the changes in the model in hillclimbing, but to otherwise use the regular M-step procedures (closed form solution when not conditioning on J for the IBM-4 and for the non-parametric IBM-3, otherwise projected gradient ascent) for the deficient models. However, we verified that this is not a good idea: not only can the likelihood go down in the process (even if we could compute expectations exactly), but these schemes also heavily increase p_0 in each iteration, i.e. the same problem Och and Ney (2003) found for the deficient models. There is therefore the need to execute the M-step properly, and when done the problem is indeed resolved.

Proper EM The expectation maximization (EM) framework (Dempster et al., 1977; Neal and Hinton, 1998) is a class of template procedures (rather than a proper algorithm) that iteratively requires solving the task

$$\max_{\theta_k} \sum_{s=1}^{S} \sum_{\mathbf{a}_s} p_{\theta_{k-1}}(\mathbf{a}_s|\mathbf{f}_s, \mathbf{e}_s) \log\left(p_{\theta_k}(\mathbf{f}_s, \mathbf{a}_s|\mathbf{e}_s)\right) \tag{10}$$

by appropriate means. Here, θ_{k-1} are the parameters from the previous iteration, while θ_k are those derived in the current iteration. Of course, here and in the following the normalization constraints on θ apply, as already in (9). On explicit request of a reviewer we give a detailed account for our setting here. Readers not interested in the details can safely move on to the next section.

Details on EM For the corpora occurring in practice, the function (10) has many more terms than there are atoms in the universe. The trick is that $p_{\theta_k}(\mathbf{f}_s, \mathbf{a}_s|\mathbf{e}_s)$ is a product of factors, where each factor depends on very few components of θ_k only. Taking the logarithm gives a sum of logarithms, and in the end we are left with the problem of computing the weights of each factor, which turn out to be expectations. To apply this to the (deficient) **IBM-3** model with parametric distortion we simplify $p_{\theta_{k-1}}(\mathbf{a}_s|\mathbf{f}_s, \mathbf{e}_s) = p(\mathbf{a}_s)$ and define the counts $n_{f,e}(\mathbf{a}_s) = \sum_{j=1}^{J_s} \delta(f_j^s, f) \cdot \delta(e_{a_j^s}^s, e)$, $n_{\Phi,e}(\mathbf{a}_s) = \sum_{i=1}^{I_s} \delta(e_i^s, e) \cdot \delta(\Phi_i(\mathbf{a}_s), \Phi)$ and $n_{j,i}(\mathbf{a}_s) = \delta(a_j^s, i)$. We also use short hand notations for sets, e.g. $\{p(f|e)\}$ is meant as the set of all translation probabilities induced by the given corpus. With this notation, after reordering the terms problem (10) can be written as

$$\begin{aligned} \max_{\{p(f|e)\},\{p(\Phi|e)\},\{p(j|i)\}} & \sum_{e,f} \Big[\sum_{s=1}^{S} \sum_{\mathbf{a}_s} p(\mathbf{a}_s)\, n_{f,e}(\mathbf{a}_s)\Big] \log\big(p(f|e)\big) \\ & + \sum_{e,\Phi} \Big[\sum_{s=1}^{S} \sum_{\mathbf{a}_s} p(\mathbf{a}_s)\, n_{\Phi,e}(\mathbf{a}_s)\Big] \log\big(p(\Phi,e)\big) \\ & + \sum_{i,j} \Big[\sum_{s=1}^{S} \sum_{\mathbf{a}_s} p(\mathbf{a}_s)\, n_{j,i}(\mathbf{a}_s)\Big] \log\big(p(j|i,J)\big)\,. \end{aligned} \tag{11}$$

Indeed, the weights in each line turn out to be nothing else than expectations of the respective factor under the distribution $p_{\theta_{k-1}}(\mathbf{a}_s|\mathbf{f}_s, \mathbf{e}_s)$ and will henceforth be written as $w_{f,e}$, $w_{\Phi,e}$ and $w_{j,i,J}$. Therefore, executing an iteration of EM requires first calculating all expectations (E-step) and then solving the maximization problems (M-step). For models such as IBM-1 and HMM the expectations can be calculated efficiently, so the enormous sum of terms in (10) is equivalently written as a manageable one. In this case it can be shown[5] that the new θ_k must have a higher likelihood (9) than θ_{k-1} (unless a stationary point is reached). In fact, any θ that has a higher value in the auxiliary function (11) than θ_{k-1} must also have a higher likelihood. This is an important background for parametric models such as (2) where the M-step cannot be solved exactly.

For IBM-3/4/5 computing exact expectations is intractable (Udupa and Maji, 2006) and approximations have to be used (in fact, even computing the likelihood for a given θ is intractable). We

[5] See e.g. the author's course notes (in German), currently `http://user.phil-fak.uni-duesseldorf.de/~tosch/downloads/statmt/wordalign.pdf`.

use the common procedure based on hillclimbing and the move/swap matrices. The likelihood is not guaranteed to increase but it (or rather its approximation) always did in each of the five run iterations. Nevertheless, the main advantage of EM is preserved: problem (11) decomposes into several smaller problems, one for each probability distribution since the parameters are tied by the normalization constraints. The result is one problem for each e involving all $p(f|e)$, one for each e involving all $p(\Phi|e)$ and one for each i involving all $p(j|i)$.

The problems for the translation probabilities and the fertility probabilities yield the known standard update rules. The most interesting case is the problem for the (parametric) distortion models. In the deficient setting, the problem for each i is

$$\max_{\{p(j|i)\}} \sum_J w_{i,j,J} \log \left(\frac{p(j|i)}{\sum_{j'=1}^{J} p(j'|i)} \right)$$

In the nondeficient setting, we now drop the subscripts i, k, J and the superscript Φ from the sets defined in the previous sections, i.e. we write $\mathcal{J}$ instead of $\mathcal{J}^{\Phi}_{i,k,J}$. The M-step problem is then

$$\max_{\{p(j|i)\}} E_i = \sum_j \sum_{\mathcal{J}: j \in \mathcal{J}} w_{j,i,\mathcal{J}} \log \big(p(j|i,\mathcal{J})\big) \ ,$$

where $w_{j,i,\mathcal{J}}$ (with $j \in \mathcal{J}$) is the expectation for aligning j to i when one can choose among the positions in $\mathcal{J}$, and with $p(j|i,\mathcal{J})$ as in (6). In principle there is an exponential number of expectations $w_{j,i,\mathcal{J}}$. However, since we approximate expectations from the move and swap matrices, and hence by $\mathcal{O}((I+J)\cdot J)$ alignments per sentence pair, in the end we get a polynomial number of terms. Currently we only consider alignments with (approximated) $p_{\theta_{k-1}}(\mathbf{a}_s|\mathbf{f}_s,\mathbf{e}_s) > 10^{-6}$.

Importantly, the fact that we get separate M-step problems for different i allows us to reduce memory consumption by using refined data structures when storing the expectations.

For both the deficient and the nondeficient variants, the M-step problems for the distortion parameters $p(j|i)$ are non-trivial, non-concave and have no (known) closed form solutions. We approach them via the method of projected gradient ascent (PGA), where the gradient for the nondeficient problem is

$$\frac{\partial E_i}{\partial p(j|i)} = \sum_{\mathcal{J}: j \in \mathcal{J}} \left[\frac{w_{j,\mathcal{J}}}{p(j|i)} - \frac{\sum_{j' \in \mathcal{J}} w_{j',\mathcal{J}}}{\sum_{j' \in \mathcal{J}} p(j'|i)} \right] .$$

When running PGA we guarantee that the resulting $\{p(j|i)\}$ has a higher function value E_i than the input ones (unless a stationary point is input). We stop when a cutoff criterion indicates a local maximum or 250 iterations are used up.

Projected Gradient Ascent This method is used in a couple of recent papers, notably (Schoenemann, 2011; Vaswani et al., 2012) and is briefly sketched here for self-containment (see those papers for more details). To solve a maximization problem

$$\max_{p(j|i)\geq 0, \sum_j p(j|i)=1} E_i(\{p(j|i)\})$$

for some (differentiable) function $E_i(\cdot)$, one iteratively starts at the current point $\{p_k(j|i)\}$, computes the gradient $\nabla E_i(\{p_k(j|i)\})$ and goes to the point

$$q(j|i) = p_k(j|i) + \alpha \nabla E_i(p_k(j|i)) \ , \ j = 1, \ldots, J$$

for some step-length α. This point is generally not a probability distribution, so one computes the *nearest* probability distribution

$$\min_{q'(j|i)\geq 0, \sum_j q'(j|i)=1} \sum_{j=1}^{J} \big(q'(j|i) - q(j|i)\big)^2 \ ,$$

a step known as *projection* which we solve with the method of (Michelot, 1986). The new distribution $\{q'(j|i)\}$ is not guaranteed to have a higher $E_i(\cdot)$, but (since the constraint set is a convex one) a suitable interpolation of $\{p_k(j|i)\}$ and $\{q'(j|i)\}$ is guaranteed to have a higher value (unless $\{p_k(j|i)\}$ is a local maximum or minimum of $E_i(\cdot)$). Such a point is computed by backtracking line search and defines the next iterate $\{p_{k+1}(j|i)\}$.

IBM-4 When moving from the IBM-3 to the IBM-4, only the last line in (11) changes. In the end one gets two new kinds of problems, for $p_{=1}(\cdot)$ and $p_{>1}(\cdot)$. For $p_{=1}(\cdot)$ we have one problem for each foreign class α and each English class β, of the form

$$\max_{\{p_{=1}(j|j',\alpha,\beta)\}} \sum_{j,j',J} w_{j,j',J,\alpha,\beta} \log\big(p_{=1}(j|j',\alpha,\beta,J)\big)$$

for reduced deficiency (with $p_{=1}(j|j',\alpha,\beta,J)$ as in (3)) and of the form

$$\max_{\{p_{=1}(j|j',\alpha,\beta)\}} \sum_{j,j',\mathcal{J}} w_{j,j',\mathcal{J},\alpha,\beta} \log\big(p_{=1}(j|j',\alpha,\beta,\mathcal{J})\big)$$

Model	Degree of Deficiency	De|En	En|De	Es|En	En|Es
HMM	nondeficient (our)	73.8	77.6	77.4	76.1
IBM-3	full (GIZA++)	74.2	76.5	74.3	74.5
IBM-3	full (our)	75.6	79.2	75.2	73.7
IBM-3	nondeficient (our)	76.1	79.8	76.8	75.5
IBM-4, 1 x 1 word class	full (GIZA++)	77.9	79.4	78.6	78.4
IBM-4, 1 x 1 word class	full (our)	76.1	81.5	77.8	78.0
IBM-4, 1 x 1 word class	reduced (our)	77.2	80.6	77.9	78.3
IBM-4, 1 x 1 word class	nondeficient (our)	77.6	81.5	80.0	78.4
IBM-4, 50 x 50 word classes	full (GIZA++)	78.6	80.4	79.3	79.3
IBM-4, 50 x 50 word classes	full (our)	78.0	82.4	79.2	79.4
IBM-4, 50 x 50 word classes	reduced (our)	78.5	82.1	79.2	79.0
IBM-4, 50 x 50 word classes	nondeficient (our)	77.9	82.5	79.7	78.2
IBM-5, 50 word classes	nondeficient (GIZA++)	79.4	81.1	80.0	79.5
IBM-5, 50 word classes	nondeficient (our)	79.2	82.7	79.7	79.5

Table 1: **Alignment accuracy** (weighted F-measure times 100, $\alpha = 0.1$) on Europarl with 100.000 sentence pairs. Reduced deficiency means renormalization as in (3) and (4), so that words cannot be placed before or after the sentence. **For the IBM-3**, the nondeficient variant is clearly best. **For the IBM-4** it is better in roughly half the cases, both with and without word classes.

for the nondeficient variant, with $p_{=1}(j|j', \alpha, \beta, \mathcal{J})$ based on (7).

For $p_{>1}(\cdot)$ we have one problem per foreign class α, of the form

$$\max_{\{p_{>1}(j|j',\alpha)\}} \sum_{j,j',J} w_{j,j',J,\alpha} \log \left(p_{>1}(j|j', \alpha, J)\right)$$

for reduced deficiency, with $p_{>1}(j|j', \alpha, J)$ based on (4), and for the nondeficient variant it has the form

$$\max_{\{p_{>1}(j|j',\alpha)\}} \sum_{j,j',\mathcal{J}} w_{j,j',\mathcal{J},\alpha} \log \left(p_{>1}(j|j', \alpha, \mathcal{J})\right) ,$$

with $p_{>1}(j|j', \alpha, \mathcal{J})$ based on (8). Calculating the gradients is analogous to the IBM-3.

5 Experiments

We test the proposed methods on subsets of the Europarl corpus for German and English as well as Spanish and English, using lower-cased corpora. We evaluate alignment accuracies on gold alignments[6] in the form of weighted F-measures with $\alpha = 0.1$, which performed well in (Fraser and Marcu, 2007b). In addition we evaluate the effect on phrase-based translation on one of the tasks.

We implement the proposed methods in our own framework RegAligner rather than GIZA++, which is only rudimentally maintained. Therefore, we compare to the deficient models in our own software as well as to those in GIZA++.

We run 5 iterations of IBM-1, followed by 5 iterations of HMM, 5 of IBM-3 and finally 5 of IBM-4. The first iteration of the IBM-3 collects counts from the HMM, and likewise the first iteration of the IBM-4 collects counts from the IBM-3 (in both cases the move and swap matrices are filled with probabilities of the former model, then theses matrices are used as in a regular model iteration). A nondeficient IBM-4 is always initialized by a nondeficient IBM-3. We did not set a fertility limit (except for GIZA++).

Experiments were run on a Core i5 with 2.5 GHz and 8 GB of memory. The latter was the main reason why we did not use still larger corpora[7]. The running times for the entire training were half a day without word classes and a day with word classes. With 50 instead of 250 PGA iterations in all M-steps we get only half these running times, but the resulting F-measures deteriorate, especially for the IBM-4 with classes.

The running times of our implementation of the IBM-5 are much more favorable: the entire training then runs in little more than an hour.

[6] from (Lambert et al., 2005) and from `http://user.phil-fak.uni-duesseldorf.de/~tosch/downloads.html`.

[7] The main memory bottleneck is the IBM-4 (6 GB without classes, 8 GB with). Using refined data structures should reduce this bottleneck.

5.1 Alignment Accuracy

The alignment accuracies – weighted F-measures with $\alpha = 0.1$ – for the tested corpora and model variants are given in Table 1. Clearly, nondeficiency greatly improves the accuracy of the IBM-3, both compared to our deficient implementation and that of GIZA++.

For the IBM-4 we get improvements for the nondeficient variant in roughly half the cases, both with and without word classes. We think this is an issue of local minima, inexactly solved M-steps and sensitiveness to initialization.

Interestingly, also the reduced deficient IBM-4 is not always better than the fully deficient variant. Again, we think this is due to problems with the non-concave nature of the models.

There is also quite some surprise regarding the IBM-5: contrary to the findings of (Och and Ney, 2003) the IBM-5 in GIZA++ performs best in three out of four cases - when competing with both deficient and nondeficient variants of IBM-3 and IBM-4. Our own implementation gives slightly different results (as we do not use smoothing), but it, too, performs very well.

5.2 Effect on Translation Performance

We also check the effect of the various alignments (all produced by RegAligner) on translation performance for phrase-based translation, randomly choosing translation from German to English. We use MOSES with a 5-gram language model (trained on 500.000 sentence pairs) and the standard setup in the MOSES Experiment Management System: training is run in both directions, the alignments are combined using `diag-grow-final-and` (Och and Ney, 2003) and the parameters of MOSES are optimized on 750 development sentences.

The resulting BLEU-scores are shown in Table 2. However, the table shows no clear trends and even the IBM-3 is not clearly inferior to the IBM-4. We think that one would need to handle larger corpora (or run multiple instances of Minimum Error Rate Training with different random seeds) to get more meaningful insights. Hence, at present our paper is primarily of theoretical value.

Model	#Classes	Deficiency	BLEU
HMM	-	nondeficient	29.72
IBM-3	-	deficient	29.63
IBM-3	-	nondeficient	29.73
IBM-4	1 x 1	fully deficient	29.91
IBM-4	1 x 1	reduced deficient	29.88
IBM-4	1 x 1	nondeficient	30.18
IBM-4	50 x 50	fully deficient	29.86
IBM-4	50 x 50	reduced deficient	30.14
IBM-4	50 x 50	nondeficient	29.90
IBM-5	50	nondeficient	29.84

Table 2: Evaluation of **phrase-based translation** from German to English with the obtained alignments (for 100.000 sentence pairs). Training is run in both directions and the resulting alignments are combined via `diag-grow-final-and`. The table shows no clear superiority of any method. In fact, the IBM-4 is not superior to the IBM-3 and the HMM is about equal to the IBM-3. We think that one needs to handle larger corpora to get clearer insights.

6 Conclusion

We have shown that the word alignment models IBM-3 and IBM-4 can be turned into nondeficient variants, an important aim of probabilistic modeling for word alignment.

Here we have exploited that the models are proper applications of the chain rule of probabilities, where deficiency is only introduced by ignoring parts of the history for the distortion factors in the factorization. By proper renormalization the desired nondeficient variants are obtained.

The arising models are trained via expectation maximization. In the E-step we use hillclimbing to get a likely alignment (ideally the Viterbi alignment). While this cannot be handled fully incrementally, it is still fast enough in practice. The M-step energies are non-concave and have no (known) closed-form solutions. They are handled via projected gradient ascent.

For the IBM-3 nondeficiency clearly improves alignment accuracy. For the IBM-4 we get improved accuracies in roughly half the cases, both with and without word classes. The IBM-5 performs surprisingly well, it is often best and hence much better than its reputation. An evaluation of phrase based translation showed no clear insights.

Nevertheless, we think that nondeficiency in fertility based models is an important issue, and that at the very least our paper is of theoretical value. The implementations are publicly available in RegAligner 1.2.

References

M. Bansal, C. Quirk, and R. Moore. 2011. Gappy phrasal alignment by agreement. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, Oregon, June.

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.

F. Cromières and S. Kurohashi. 2009. An alignment algorithm using Belief Propagation and a structure-based distortion model. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Athens, Greece, April.

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

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.

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

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.

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.

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.

S. Ravi and K. Knight. 2010. Does GIZA++ make search errors? *Computational Linguistics*, 36(3).

T. Schoenemann. 2010. Computing optimal alignments for the IBM-3 translation model. In *Conference on Computational Natural Language Learning (CoNLL)*, Uppsala, Sweden, July.

T. Schoenemann. 2011. Regularizing mono- and bi-word models for word alignment. In *International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, November.

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.

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.

R. Udupa and H.K. Maji. 2006. Computational complexity of statistical machine translation. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy, April.

A. Vaswani, L. Huang, and D. Chiang. 2012. Smaller alignment models for better translations: Unsupervised word alignment with the l_0-norm. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Korea, 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.