# Probabilistic Word Alignment under the $L_0$-norm

**Thomas Schoenemann**
Center for Mathematical Sciences
Lund University, Sweden

## Abstract

This paper makes two contributions to the area of single-word based word alignment for bilingual sentence pairs. Firstly, it integrates the – seemingly rather different – works of (Bodrumlu et al., 2009) and the standard probabilistic ones into a single framework.

Secondly, we present two algorithms to optimize the arising task. The first is an iterative scheme similar to Viterbi training, able to handle large tasks. The second is based on the inexact solution of an integer program. While it can handle only small corpora, it allows more insight into the quality of the model and the performance of the iterative scheme.

Finally, we present an alternative way to handle prior dictionary knowledge and discuss connections to computing IBM-3 Viterbi alignments.

## 1 Introduction

The training of *single word based* translation models (Brown et al., 1993b; Vogel et al., 1996) is an essential building block for most state-of-the-art translation systems. Indeed, even more refined translation models (Wang and Waibel, 1998; Sumita et al., 2004; Deng and Byrne, 2005; Fraser and Marcu, 2007a) are initialized by the parameters of single word based ones. The exception is here the joint approach of Marcu and Wong (2002), but its refinement by Birch et al. (2006) again relies on the well-known IBM models.

Traditionally (Brown et al., 1993b; Al-Onaizan et al., 1999) single word based models are trained by the EM-algorithm, which has the advantageous property that the collection of counts can be decomposed over the sentences. Refinements that also allow symmetrized models are based on bipartite graph matching (Matusov et al., 2004; Taskar et al., 2005) or quadratic assignment problems (Lacoste-Julien et al., 2006). Recently, Bodrumlu et al. (2009) proposed the first method that treats a non-decomposable problem by handling all sentence pairs at once and via integer linear programming. Their (non-probabilistic) approach finds dictionaries with a minimal number of entries. However, the approach does not include a position model.

In this work we combine the two strategies into a single framework. That is, the dictionary sparsity objective of Bodrumlu et al. will become a *regularity term* in our framework. It is combined with the maximal alignment probability of every sentence pair, where we consider the models IBM-1, IBM-2 and HMM. This allows us to write dictionary sparsity as the (non-convex) $L_0$ norm of the dictionary parameters of the respective models.

For *supervised* training, regularity terms are quite common, e.g. (Taskar et al., 2005; Lacoste-Julien et al., 2006). For the unsupervised problem addressed in this paper they have recently been introduced in the form of posterior constraints (Ganchev et al., 2010). In related fields of NLP lately Dirichlet priors have been investigated, e.g. (Johnson, 2007).

We present two strategies to handle the objective function addressed in this paper. One of these schemes relies, like (Germann et al., 2004; Lacoste-Julien et al., 2006; DeNero and Klein, 2008; Bodrumlu et al., 2009), on integer linear programming



*Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 172–180,
Portland, Oregon, USA, 23–24 June 2011. 

(see e.g. (Schrijver, 1986; Achterberg, 2007)), but due to the large-scale nature of our problem we solve only the LP-relaxation, followed by successive strengthening. For the latter, we develop our own, exponentially large set of cuts and show that it can be handled as a polynomially sized system, though in practice this is too inefficient.

## 2 The Models

Before we introduce our objective function we give a brief description of the (standard) models we consider. In all cases, one is given a set of bilingual sentence pairs containing a foreign language and English. The models formalize the probability of obtaining the foreign sentence from a given English sentence, by considering *hidden* variables called alignments:

$$p_{\mathbf{d},\mathbf{l}}(\mathbf{f}_s|\mathbf{e}_s) = \sum_{\mathbf{a}_s} p_{\mathbf{d},\mathbf{l}}(\mathbf{f}_s, \mathbf{a}_s|\mathbf{e}_s) \ .$$

Here, the subscripts $\mathbf{d}$ and $\mathbf{l}$ denote two sets of parameters: whereas the set $\mathbf{l}$ defines the probability of an alignment without knowing any of the sentences, $\mathbf{d}$ describes the translational probability given an alignment and a source sentence.

For a source (English) sentence of length $I$ and a target (foreign) sentence of length $J$, the set of admissible alignments is generally the set of subsets of $\{1,\ldots,I\} \times \{1,\ldots,J\}$. However, for computational reasons the considered models only allow restricted alignments, where each target word may align to at most one source word. Any such alignment is expressed as a vector $\mathbf{a}_1^J \in \{0,\ldots,I\}^J$.

### 2.1 Considered models

For a source sentence $\mathbf{e}^s = e_1^I$ and a target sentence $\mathbf{f}^s = f_1^J$, the considered models all factor as follows:

$$p_{\mathbf{d},\mathbf{l}}(\mathbf{f}^s, \mathbf{a}^s|\mathbf{e}^s) \ = \ \prod_{j=1}^{J} p_{\mathbf{d}}(f_j|e_{a_j}) \cdot \ p_{\mathbf{l}}(a_j|a_{j-1}, j, I) \tag{1}$$

In all cases, the translational probability is non-parametric, i.e. $\mathbf{d}$ contains one parameter for every co-occurring pair of source and target words. Since the model is probabilistic, the parameters of all $f$ for a given $e$ have to sum up to one.

With respect to the alignment probability, the models differ. For the IBM-1 the set $\mathbf{l}$ is actually empty, so $p_{\mathbf{l}}(a_j|a_{j-1}, j, I) = 1/(I+1)$. The IBM-2 models[1] $p(a_j|j, I)$, with a respective set of parameters. Finally, the HMM models $p(a_j|a_{j-1}, I)$.

It is common to further reduce the alignment parameters. In this paper we consider a nonparametric distribution for the IBM-2, but both a nonparametric and a parametric one for the HMM. In contrast to GIZA++, we have a parameter for every possible difference, i.e. we do not group differences with absolutes greater than 5. Also, commonly one uses a distribution $p(i|i', I) = r(i - i')/\sum_{i''=1}^{I} r(i'' - i')$, but for technical reasons, we drop the denominator and instead constrain the $r(\cdot)$-parameters to sum to 1. In future work we hope to implement both the normalization and the grouping of bins.

### 2.2 Word Alignment

Originally the considered models were used for the actual translation problem. Hence, the parameters $\mathbf{d}$ and $\mathbf{l}$ had to be inferred from a training corpus, which was based on maximizing the probability

$$\max_{\mathbf{d},\mathbf{l}} \prod_s \sum_{\mathbf{a}} p_{\mathbf{d},\mathbf{l}}(\mathbf{f}^s, \mathbf{a}^s|\mathbf{e}^s) \ . \tag{2}$$

Today the major application of the models lies in *word alignment*. Instead of estimating continuous parameters, one is now faced with the discrete optimization problem of assigning a single alignment to every sentence pair in the corpus. This lead to the recent innovative work of (Bodrumlu et al., 2009) where the alignments are the only unknown quantities.

Nevertheless, the use of probabilistic models remains attractive, in particular since they contribute statistics of likely alignments. In this work, we combine the two concepts into the criterion

$$\min_{\mathbf{d},\mathbf{l}} -\log\left[\prod_s \max_{\mathbf{a}^s} p_{\mathbf{d},\mathbf{l}}(\mathbf{f}^s, \mathbf{a}^s|\mathbf{e}^s)\right] \ + \ \lambda \left\|\mathbf{d}\right\|_0 \ ,$$

where $\lambda \geq 0$ is a weighting parameter and we now estimate a single alignment for every sentence.

The second term denotes the $L_0$-norm of the translational parameters, i.e. the number of non-zero

[1] The original work considered a dependence on $I$ and $J$, but it is common to drop $J$.

parameters. Since we only consider a single alignment per sentence, this term is equivalent to Bodrumlu et al.'s objective function. Minimizing the first term is closely related to the common criterion (2). For parameter estimation it is known as the *maximum approximation*, but for word alignment it is a perfectly valid model.

For the IBM-1 model the first term alone results in a convex, but not strictly convex minimization problem[2]. However, EM-like iterative methods generally do not reach the minimum: they are doing block coordinate descent (Bertsekas, 1999, chap. 2.7) which generally gives the optimum only for *strictly* convex functions. Indeed, our experiments showed a strong dependence on initialization and lead to heavily local solutions.

In the following we present two strategies to minimize the new objective. We start with an iterative method that also handles the regularity term.

## 3 An Iterative Scheme

To derive our algorithms, we first switch the minimum and maximum in the objective and obtain

$$\min_{\{\mathbf{a}^s\}} \min_{\mathbf{d},\mathbf{l}} -\sum_s \log\left[p_{\mathbf{d},\mathbf{l}}(\mathbf{f}^s, \mathbf{a}^s|\mathbf{e}^s)\right] \;+\; \lambda \left\|\mathbf{d}\right\|_0 \;,$$

where the notation$\{\mathbf{a}^s\}$ denotes the alignments of all sentence pairs. Ignoring the $L_0$-term for the moment, we now make use of a result of (Vicente et al., 2009) in their recent work on histogram-based image segmentation: for any *given* set of alignments, the inner minimization over the parameters is solved by relative frequencies. When plugging this solution into the functional, one gets a model that does not decompose over the sentences, but one that is still reasonably easy to handle.

Before we get into details, we observe that this minimizer is valid even when including the $L_0$ term: if two words are never linked, both terms will set the respective parameter to 0. If they are linked, however, then setting this parameter to 0 would make the first term infinite. All non-zero parameters are treated equally by the $L_0$ term, so the restriction to relative frequencies does not change the optimal value. In fact, this can be extended to *weighted* $L_0$ terms, and later on we exploit this to derive an alternative way to handle a dictionary prior. Note that the same principle could also be applied to the work of (Bodrumlu et al., 2009).

[2]This is due to taking the maximizing alignment. Summing over all alignments is strictly convex.

### 3.1 Mathematical Formulation

We detail our scheme for the IBM-1 model, the extension to other models is easy. For given alignments we introduce the counts

$$N_{f,e}(\{\mathbf{a}^s\}) = \sum_s \sum_j \delta(f, f_j) \cdot \delta(e, e_{a_j})$$

$$N_e(\{\mathbf{a}^s\}) = \sum_s \sum_j \delta(e, e_{a_j}) \;\;,$$

where $\delta(\cdot,\cdot)$ is the Kronecker-delta. The optimal translation parameters are then given by $N_{f,e}(\{\mathbf{a}^s\})/N_e(\{\mathbf{a}^s\})$, and plugging this into the first term in the objective gives (up to a constant)

$$\min_{\{\mathbf{a}^s\}} \sum_{f,e} -N_{f,e}(\{\mathbf{a}^s\}) \log\left(\frac{N_{f,e}(\{\mathbf{a}^s\})}{N_e(\{\mathbf{a}^s\})}\right) \;\;.$$

The second term is simply $\lambda \sum_{f,e} \|N_{f,e}(\{\mathbf{a}^s\})\|_0$, and since $N(e) = \sum_f N(f,e)$, in total we get

$$\begin{aligned}\min_{\{\mathbf{a}^s\}} \sum_{f,e} & -N_{f,e}(\{\mathbf{a}^s\}) \log\left(N_{f,e}(\{\mathbf{a}^s\})\right) \\ & + \sum_e N_e(\{\mathbf{a}^s\}) \log\left(N_e(\{\mathbf{a}^s\})\right) \;\;. \\ & + \lambda \sum_{f,e} \|N_{f,e}(\{\mathbf{a}^s\})\|_0 \end{aligned} \tag{3}$$

In essence we are now dealing with the function $x\,\log(x)$, where its value for 0 is defined as 0.

### 3.2 Algorithm

For the new objective, we were able to entirely get rid of the model parameters, leaving only alignment variables. Nevertheless, the algorithm we present maintains these parameters, and it requires an initial choice. While we initialize the alignment parameters uniformly, for the translation parameters we use co-occurrence counts. This performed much better than a uniform initialization. The algorithm, called AM (for alternating minimization), now iterates two steps:

1. Starting from the current setting of $\mathbf{d}$ and $\mathbf{l}$, derive Viterbi alignments for all sentence pairs. E.g. for the IBM-1 we set $a_j^s = \arg\max_i d(f_j|e_i)$. For the IBM-2 the term is similar, while for the HMM one can use dynamic programming.

   Note that this step does not consider the $L_0$-term. This term can however not increase.

2. Run the Iterated Conditional Modes (Besag, 1986), i.e. go sequentially over all alignment variables and set them to their optimal value when keeping the others fixed.

   Here, we need to keep track of the current alignment counts. In every step we need to compute the objective cost associated to a count that increases by 1, and (another) one that decreases by 1. For the IBM-2 we need to consider the alignment counts, too, and for the HMM usually two alignment terms are affected. In case of 0-alignments there can be more than two. We presently do not consider these cases and hence do not find the exact optimum there.

   Afterwards, reestimate the parameters $\mathbf{d}$ and $\mathbf{l}$ from the final counts.

## 4 Integer Linear Programming

The above algorithm is fast and can handle large corpora. However, it still gets stuck in local minima, and there is no way of telling how close to the optimum one got.

This motivates the second algorithm where we cast the objective function as an integer linear program (ILP). In practice it is too large to be solved exactly, so we solve its linear programming relaxation, followed by successive strengthening. Here we derive our own set of cuts. Now we also get a lower bound on the problem and obtain lower energy solutions in practice. But we can handle only small corpora.

We limit this method to the models IBM-1 and IBM-2. Handling an HMM would be possible, but its first order alignment model would introduce many more variables. Handling the IBM-3, based on (Ravi and Knight, 2010; Schoenemann, 2010) seems a more promising direction.

### 4.1 An ILP for the Regularized IBM-1

The basis of our ILP is the fact that the counts $N_{f,e}$ and $N_e$ can only assume a finite, a-priori known set of integral values, including 0. We introduce a binary variable $n_{f,e}^c \in \{0,1\}$ for each possible value $c$, where we want $n_{f,e}^c = 1$ if $N_{f,e}(\mathbf{a}^s) = c$, otherwise $n_{f,e}^c = 0$. This is analogous for the variables $n_e^c$ and $N_e(\mathbf{a}^s)$. Finally, since the counts depend on the alignments, we also need binary variables $x_{i,j}^s \in \{0,1\}$ that we want to be 1 if and only if $a_j^s = i$.

The cost function of (3) can now be written as a linear function in terms of the integer variables $n_{f,e}^c$ and $n_e^c$, with coefficients

$$w_{e,f}^c = -c\log(c) + \lambda\|c\|_0\,, \quad w_e^c = c\log(c)\ .$$

However, we need additional constraints. In particular we need to ensure that for a given $f$ and $e$ exactly one variable $n_{f,e}^c$ is 1. Equivalently we can postulate that the sum of these variables is one. We proceed analogous for each $e$ and the variables $n_e^c$.

Then, we need to ensure that for each source word in each sentence $\mathbf{f}^s$ an alignment is specified, i.e. that for each given $s$ and $j$ the variables $x_{i,j}^s$ sum to 1. Finally, the count variables have to reflect the counts induced by the alignment variables. For the counts $N_{f,e}$ this is expressed by

$$\sum_{s,i,j:f_j^s=f,e_i^s=e} x_{i,j}^s = \sum_c c\cdot n_{f,e}^c \quad \forall f,e\ ,$$

and likewise for the counts $N_e$.

Altogether, we arrive at the following system:

$$\min_{\{x_{i,j}^s\},\{n_{f,e}^c\},\{n_e^c\}} \sum_{e,c} w_e^c\, n_e^c \;+\; \sum_{f,e,c} w_{f,e}^c\, n_{f,e}^c$$

$$\text{s.t.} \qquad \sum_i x_{i,j}^s = 1 \quad \forall s,j$$

$$\sum_c n_{f,e}^c = 1 \quad \forall f,e$$

$$\sum_c n_e^c = 1 \quad \forall e$$

$$\sum_{s,i,j:f_j=f,e_i=e} x_{i,j}^s = \sum_c c\cdot n_{f,e}^c \quad \forall f,e$$

$$\sum_{s,i,j:e_i=e} x_{i,j}^s = \sum_c c\cdot n_e^c \quad \forall e$$

$$x_{i,j}^s \in \{0,1\},\ n_e^c \in \{0,1\},\ n_{e,f}^c \in \{0,1\}\ .$$

### 4.2 Handling the IBM-2

The above mentioned system can be easily adapted to the IBM-2 model. To this end, we introduce variables $n^c_{i,j,I} \in \{0,1\}$ to express how often source word $j$ is aligned to target word $i$ given that there are $I$ words in the target sentence. Note that the number of times source word $j$ is aligned given that the target sentence has $I$ words is known a-priori and does not depend on the alignment to be optimized. We denote it $C_{j,I}$. The cost function of the ILP is augmented by $\sum_{i,j,I,c} w^c_{i,j,I}\, n^c_{i,j,I}$ , with $w^c_{i,j,I} = c \log(c/C_{j,I})$. In addition we add the following constraints to the system:

$$\sum_{s:I^s=I} x^s_{i,j} = \sum_c c \cdot n^c_{i,j,I} \quad \forall i,j,I \quad .$$

## 5 Cutting Planes

Integer linear programming is an NP-hard problem (see e.g. (Schrijver, 1986)). While for problems with a few thousand variables it can often be solved exactly via the branch and cut method, in our setting none of the solvers we tried terminated. Already solving the linear programming relaxation requires up to 4 GB of memory for corpora with roughly 3000 sentence pairs.

So instead of looking for an exact solution, we make use of a few iterations of the cutting planes method (Gomory, 1958), where repeatedly an LP-relaxation is solved, then additionally valid inequalities, called *cuts*, are added to the system. Every round gives a tighter relaxation, i.e. a better lower bound on the optimal integral value.

After solving each LP-relaxation we derive an integral solution by starting the iterative method from section 3 from the fractional LP-solution. In the end we output the best found integral solution.

For deriving cuts we tried all the methods implemented in the COIN Cut Generation Library CGL[3], based on the solver Clp from the same project line. However, either the methods were very slow in producing cuts or they produced very few cuts only. So eventually we derived our own set of cuts that will now be presented. Note that they alone do not give an integral solution.

[3] `http://www.coin-or.org/projects/Cgl.xml`

### 5.1 A Set of Count-based Cuts

The derived ILP contains several constraints of the form

$$\sum_i y_i = \sum_c c \cdot z_c \quad , \tag{4}$$

where all variables are binary. Expressions of this kind arise wherever we need to ensure consistency between alignment variables and count variables. Our cuts aim at strengthening each of these equations individually.

Assume that equation (4) involves the variables $y_1, \ldots, y_K$ and hence also the variables $z_0, \ldots, z_K$. The trouble with the equation is that even if the left hand side is integral, the right-hand side is usually not. As an example, consider the case where $\sum_{i=1}^K y_i = 3$. Then the fractional assignment $z_0 = 1 - 3/K,\ z_K = 3/K$ and $z_c = 0$ for all other $c$ satisfies (4). Indeed, if the cost function for $z_c$ is concave in $c$, as is the function $-c\ \log(c)$ we use, this will be the optimal solution for the given left hand side.

Hence we want to enforce that for an integral value $k$ of the left hand side, all variables $z_c$ for $0 \le c < k$ are zero. This is ensured by the following system of inequalities that is exponentially large in $k$:

$$\sum_{i \in \mathcal{K}} y_i + \sum_{c=0}^{k-1} z_c \le k \tag{5}$$
$$\forall \mathcal{K} \subseteq \{1, \ldots, K\} : |\mathcal{K}| = k \ .$$

It turns out that this system can be formulated quite compactly.

### 5.2 Polynomial Formulation

We now formalize the result for the compact formulation of (5).

**Proposition 1** *The union of the systems (5) for all $k$ can be represented by polynomially many variables and linear constraints.*

**Proof**: we first observe that it suffices to enforce

$$\left[ \max_{\mathcal{K}:|\mathcal{K}|=k} \sum_{i \in \mathcal{K}} y_i \right] + \sum_{c=0}^{k-1} z_c \le k$$

for all $k$. These are polynomially many equations (one for each $k$), but each involves a maximization

over exponentially many sets. However, this maximization can be expressed by the auxiliary variables

$$\begin{aligned}\tau_l^k &:= \max_{\mathcal{K}\subseteq\{1,\ldots,l\}:|\mathcal{K}|\leq k} \sum_{i\in\mathcal{K}} y_i \\ &= \max\{\tau_{l-1}^{k-1} + y_l\,,\, \tau_{l-1}^k\}\end{aligned}$$

Now we only have to maximize over two linear expressions for each of the new, polynomially many, variables. We can enforce $\tau_l^k$ to be an upper bound on the maximum by postulating $\tau_l^k \geq \tau_{l-1}^{k-1} + y_l$ and $\tau_l^k \geq \tau_{l-1}^k$. Since the original maximum occurred on the left hand side of a less-than constraint, this upper bound will be tight wherever this is necessary. □

In practice the arising system is numerically hard to solve. Since usually only polynomially many cuts of the form (5) are actually violated during the cutting planes process, we add them in passes and get significantly lower running times. Moreover, each round of cuts gives a new opportunity to derive a heuristic integral solution.

### 5.3 Backward Cuts

We call the cuts we have derived above *forward cuts* as they focus on variables that are smaller than $k$. If we could be sure that the left-hand side of (4) was always integral, they should indeed suffice. In practice this is not the case, so we now also derive *backward cuts* where we focus on all variables that are larger than $k$, with the following reasoning: once we know that at least $K - k$ variables $y_i$ are inactive (i.e. $y_i = 0$), we can conclude that all $z_c$ with $c > k$ must be zero, too. This can be expressed by the set of inequalities

$$\sum_{i\in\mathcal{K}}(1 - y_i) + \sum_{c=k+1}^{K} z_c \leq K - k$$
$$\forall\mathcal{K} \subseteq \{1,\ldots,K\} : |\mathcal{K}| = K - k\ ,$$

or equivalently

$$\sum_{i\in\mathcal{K}} -y_i + \sum_{c=k+1}^{K} z_c \leq 0 \quad \forall\mathcal{K} : |\mathcal{K}| = K - k\ .$$

### 5.4 Other Applications

A related constraint system arises in recent work (Ravi and Knight, 2010; Schoenemann, 2010) on computing IBM-3 Viterbi alignments. We implemented[4] the polynomial formulation of the above forward cuts for this system, and got mild speed-ups (224 instead of 237 minutes for the Hansards task reported in the second paper). With an additionally improved fertility exclusion stage[5] this is reduced to 176 minutes.

## 6 Experiments

We evaluate the proposed strategies on both small scale and (where applicable) large scale tasks. We compare to standard EM with sums over alignments, where for the IBM-1 and the HMM we use GIZA++. In addition, we evaluate several variants (our implementations) of the HMM, with non-parametric and parametric alignment models. Note that for the non-parametric variant we estimate distributions for the first aligned position, for the parametric all initial positions are equally likely. For the IBM-2 we consider the non-parametric variant and hence our implementation. We also evaluate our schemes on the task without regularization.

All experiments in this work were executed on a 3 GHz Core 2 Duo machine with 8 GB of memory, where up to 4 GB were actually used. The iterative scheme was run for 15 iterations, where it was nearly converged. This setting was also used for our own EM-implementations. Solely for GIZA++ we used the standard setting of 5 iterations, and the implemented smoothing process. For the IBM-2 and HMM we follow the standard strategy to first train an IBM-1 with the same objective function.

### 6.1 Large Scale Tasks

We consider two well-known corpora with publicly available gold alignments, and run both translation directions for each of them. The first task is the Canadian Hansards task (French and English) with roughly 1 million sentences. The second task is Europarl Spanish-English, where we take the first 500000 sentences. Our iterative scheme runs in

[4] based on code available at `www.maths.lth.se/matematiklth/personal/tosch/download.html`.

[5] In (Schoenemann, 2010) we stated that the difference between $c_{if}^y$ and the contribution of $i$ to the bound has to exceed $u - l_3$. This can be tightened if one additionally adds the cost of the best $f$ alignments to $i$ to the cost $c_{if}^y$.

| Canadian Hansards | | |
|---|---|---|
| | Fr → En | En → Fr |
| HMM (Giza++) | 0.918 | 0.918 |
| par. HMM (our EM) | 0.887 | 0.896 |
| par. HMM (Viterbi) | 0.873 | 0.897 |
| par. HMM + $L_0$ | 0.891 | 0.907 |
| nonpar. HMM (our EM) | 0.873 | 0.911 |
| nonpar. HMM (Viterbi) | 0.881 | 0.909 |
| nonpar. HMM + $L_0$ | 0.902 | 0.917 |

| Europarl | | |
|---|---|---|
| | Es → En | En → Es |
| HMM (Giza++) | 0.764 | 0.754 |
| nonpar. HMM (our EM) | 0.738 | 0.733 |
| nonpar. HMM (Viterbi) | 0.726 | 0.716 |
| nonpar. HMM + $L_0$ | 0.729 | 0.73 |

Table 1: For large corpora, the proposed scheme outperforms Viterbi training and sometimes even our EM.

roughly 5 hours (with room for improvements), using 2.5 GB memory. We found that an $L_0$-weight of $\lambda = 5.0$ performs very well. Hence, we will use this for all our experiments.

We compare to the standard GIZA++ implementation and our own HMM implementations with EM. Here we ran 15 iterations for IBM-1 and HMM each.

As shown in Table 1 adding the $L_0$ term improves the standard Viterbi training. Our method also sometimes beats the simple EM implementation but not GIZA++. This may be due to the special parametric model of GIZA++, its smoothing process or the lower number of iterations. Our deficient parametric model is inferior for the Hansards task, so we did not run it for Europarl.

### 6.2 Small Scale Tasks

To evaluate the ILP strategy we consider four small scale tasks released by the European Corpus Initiative[6]. See (Schoenemann, 2010) for the corpus statistics. We report weighted f-measures (Fraser and Marcu, 2007b) on gold alignments (sure and possible) specified by one annotator, for 144 and 110 sentences respectively. The number of cut rounds was selected so that the execution times remained below 2 days for all tasks. This was 50 rounds for the IBM-1 and 2 rounds for the IBM-2. In fact, with these numbers the Avalanche task is processed in little less than a day.

We tested a number of LP solvers and found that most of them need several hours to solve the root relaxation. This is different for the commercial solver FICO Xpress, which only needs around 15 minutes. However, it is slower in processing the subsequent cut iterations. Hence, for the IBM-1 we use the open source Clp[7].

The resulting f-measures for the tested strategies are given in Table 2. In all cases, adding the $L_0$ term greatly improves the standard Viterbi training. Moreover, for the small scale tasks, the parametric HMM is clearly the best choice when using the $L_0$ penalty. In the majority of cases the ILP strategy performs better than the iterative scheme. In fact, it *always* found the lower energy solution. The most extreme difference we observed for the IBM-2 on the UBS English to German task: here AM finds an energy of 318147, where the ILP gives 303674.

Finally, Table 3 evaluates the effectiveness of the cut strategy exemplarily on one of the tasks. Clearly, the gaps are reduced significantly compared to the LP-relaxation. However, except for the IBM-1 (which is convex for $\lambda = 0$) the lower bounds are still quite loose.

### 6.3 Handling Dictionary Knowledge

The presented $L_0$ regularity is easily modified to include dictionary knowledge[8]. To this end, we introduce a *weighted* $L_0$-norm: whenever a pair of source and target words is listed in the dictionary, the entry is not penalized. All remaining entries are penalized by $\lambda$.

Note that this is different from the standard way (Brown et al., 1993a) of handling dictionary knowledge, which appends the dictionary to the corpus (with a proper weighting factor). We tried both schemes with several weighting factors, then chose the best-performing for the UBS task. For the UBS German to English task we get an accuracy of 0.445, which beats GIZA++ both with (0.438) and without (0.398) dictionary knowledge. In the reverse direction both schemes profit from the extra knowledge,

[6] http://www.elsnet.org/eci.html

[7] http://www.coin-or.org/projects/Clp.xml

[8] Our data are based on www.dict.info/uddl.php and www.ilovelanguages.com/idp and the stemming algorithms at snowball.tartarus.org.

| Avalanche French → German | | | |
|---|---|---|---|
| Model | EM | AM | ILP |
| IBM-1 | 0.603 | 0.619 | 0.591 |
| IBM-1 + $L_0$ | – | 0.64 | 0.625 |
| IBM-2 | 0.568 | 0.632 | 0.60 |
| IBM-2 + $L_0$ | – | 0.680 | 0.636 |
| par. HMM | 0.752 | 0.621 | – |
| par. HMM + $L_0$ | – | 0.779 | – |
| nonpar. HMM | 0.752 | 0.655 | – |
| nonpar. HMM + $L_0$ | – | 0.714 | – |

| Avalanche German → French | | | |
|---|---|---|---|
| Model | EM | AM | ILP |
| IBM-1 | 0.494 | 0.485 | 0.507 |
| IBM-1 + $L_0$ | – | 0.497 | 0.488 |
| IBM-2 | 0.428 | 0.459 | 0.526 |
| IBM-2 + $L_0$ | – | 0.483 | 0.55 |
| par. HMM | 0.606 | 0.49 | – |
| par. HMM + $L_0$ | – | 0.592 | – |
| nonpar. HMM | 0.582 | 0.501 | – |
| nonpar. HMM + $L_0$ | – | 0.537 | – |

| UBS German → English | | | |
|---|---|---|---|
| Model | EM | AM | ILP |
| IBM-1 | 0.381 | 0.359 | 0.335 |
| IBM-1 + $L_0$ | – | 0.350 | 0.442 |
| IBM-2 | 0.315 | 0.324 | 0.340 |
| IBM-2 + $L_0$ | – | 0.383 | 0.462 |
| par. HMM | 0.398 | 0.229 | – |
| par. HMM + $L_0$ | – | 0.383 | – |
| nonpar. HMM | 0.421 | 0.29 | – |
| nonpar. HMM + $L_0$ | – | 0.371 | – |

| UBS English → German | | | |
|---|---|---|---|
| Model | EM | AM | ILP |
| IBM-1 | 0.515 | 0.435 | 0.489 |
| IBM-1 + $L_0$ | – | 0.444 | 0.504 |
| IBM-2 | 0.417 | 0.40 | 0.435 |
| IBM-2 + $L_0$ | – | 0.52 | 0.571 |
| par. HMM | 0.625 | 0.404 | – |
| par. HMM + $L_0$ | – | 0.537 | – |
| nonpar. HMM | 0.623 | 0.436 | – |
| nonpar. HMM + $L_0$ | – | 0.524 | – |

Table 2: Alignment accuracy (weighted f-measure) for different algorithms. We use a dictionary penalty of $\lambda = 5$ and the standard EM (GIZA++ for IBM-1 and parametric HMM, our implementation otherwise) training scheme with 5 iterations for each model.

| UBS English → German | | | |
|---|---|---|---|
| | $L_0$-weight | IBM-1 | IBM-2 |
| root relaxation | 0.0 | 1.098 | 7.697 |
| after cut rounds | 0.0 | 1.081 | 5.67 |
| root relaxation | 5.0 | 1.16 | 2.76 |
| after cut rounds | 5.0 | 1.107 | 2.36 |

Table 3: Ratios of the best known integer solution and the best known lower bounds for all considered tasks.

but GIZA++ remains the clear winner. Applying the same weights to the above mentioned Hansards task slightly improved GIZA++, whereas it slightly worsened the performance of our scheme in the one direction and slightly improved it in the other. We intend to investigate this more thoroughly in the future.

## 7 Discussion

In this paper we have shown that an $L_0$ prior on the dictionary parameters greatly improves Viterbi training. A simple iterative scheme often nearly matches our EM-implementation of the HMM.

We have also derived two algorithms to deal with the new objective. A simple iterative scheme gives quite accurate results on large scale tasks. On small scale tasks our inexact ILP strategy shows that the iterative scheme does not find the optimum in practice, a point that may well carry over to other models trained with the maximum approximation. This strategy also provides lower bounds, but at present they are quite loose.

Moreover, we have presented an alternative way of handling dictionary knowledge. Finally, we have discussed connections to computing IBM-3 Viterbi alignments, where we got mild speed-ups.

In future work we intend to investigate the effect of the generated alignments on the translation quality of phrase-based approaches. We also want to explore strategies to determine the regularity weight. Finally, we want to handle a non-deficient parametric HMM.

**Acknowledgements.** We thank Ben Taskar and João Graça for helpful discussions. This work was funded by the European Research Council (GlobalVision grant no. 209480).

## References


T. Achterberg. 2007. *Constraint Integer Programming*. Ph.D. thesis, Zuse Institut, TU Berlin, Germany, July.

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/`.

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

J. Besag. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302.

A. Birch, C. Callison-Burch, and M. Osborne. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Conference of the Association for Machine Translation in the Americas (AMTA)*, Cambridge, Massachusetts, August.

T. Bodrumlu, K. Knight, and S. Ravi. 2009. A new objective function for word alignment. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing (ILP)*, Boulder, Colorado, June.

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, M.J. Goldsmith, J. Hajic, R.L. Mercer, and S. Mohanty. 1993a. But dictionaries are data too. In *HLT workshop on Human Language Technology*.

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

J. DeNero and D. Klein. 2008. The complexity of phrase alignment problems. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio, 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.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2004. Fast decoding and optimal decoding for machine translation. *Artificial Intelligence*, 154(1–2), April.

R.E. Gomory. 1958. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278.

M. Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, June.

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.

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.

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.

A. Schrijver. 1986. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons.

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.

S. Vicente, V.N. Kolmogorov, and C. Rother. 2009. Joint optimization of segmentation and appearance models. In *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, September.

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.