Model Invertibility Regularization: Sequence Alignment With or Without Parallel Data

Tomer Levinboim[*] levinboim.1@nd.edu
Ashish Vaswani[†] avaswani@isi.edu
David Chiang[*] dchiang@nd.edu

[*]Dept. of Computer Science and Engineering
University of Notre Dame

[†]Information Sciences Institute
University of Southern California

Abstract

We present Model Invertibility Regularization (MIR), a method that jointly trains two directional sequence alignment models, one in each direction, and takes into account the invertibility of the alignment task. By coupling the two models through their parameters (as opposed to through their inferences, as in Liang et al.'s Alignment by Agreement (ABA), and Ganchev et al.'s Posterior Regularization (PostCAT)), our method seamlessly extends to all IBM-style word alignment models as well as to alignment *without* parallel data. Our proposed algorithm is mathematically sound and inherits its convergence guarantees from EM. We evaluate MIR on two tasks: (1) On word alignment, applying MIR on fertility based models we attain higher F-scores than ABA and PostCAT. (2) On Japanese-to-English back-transliteration without parallel data, applied to the decipherment model of Ravi and Knight, MIR learns sparser models that close the gap in whole-name error rate by 33% relative to a model trained on parallel data, and further, beats a previous approach by Mylonakis et al.

1 Introduction

The transfer of information between languages is a common natural language phenomenon that is intuitively invertible. For example, in transliteration, a source-language word is mapped to a target language's writing system under a sound preserving mapping (for example, "computer" to Japanese Romaji, "konpyutaa"). The original word should then be recoverable from its transliterated version. Similarly, in translation, the back-translation of the translation of a word is likely to be that same word itself.

In NLP, however, commonly-used generative models describing such phenomena are directional, only concerned with the transfer of source-language symbols to target-language symbols or vice versa, but not both directions. Left unchecked, independently training two such directional models (source-to-target and target-to-source) often yields two models that diverge from this invertibility intuition.

In word alignment, this can lead to disagreements between alignments inferred by a model trained in one direction and those inferred by a model trained in the reverse direction. To remedy this disparity (and other shortcomings), it is common to turn to alignment symmetrization techniques such as grow-diag-final-and (Koehn et al., 2003) which heuristically combines alignments from both directions.

Liang et al. (2006) suggest a more fundamental approach they call Alignment by Agreement (ABA), which jointly trains two word alignment models by maximizing their data-likelihoods along with a regularizer that rewards agreement between their alignment posteriors (computed over each parallel sentence pair). Although their EM-like optimization procedure is heuristic, it proves effective at jointly training bidirectional models. Ganchev et al. (2008) propose another approach for agreement between the directed models by adding constraints on the alignment posteriors. Unlike ABA, their optimization is exact, but it can be computationally expensive, requiring multiple forward-backward inferences in each E-step.

In this paper we develop a different approach for jointly training general bidirectional sequence alignment models called Model Invertibility Regularization, or MIR (Section 3). Our approach has two key benefits over ABA and PostCAT: First, MIR can

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 609–618,
Denver, Colorado, May 31 – June 5, 2015.

be applied to sequence alignment without parallel data. Second, a single implementation seamlessly extends to all IBM models, including the fertility based models. Furthermore, since MIR follows the MAP-EM framework, it inherits its desirable convergence guarantees.

The key idea facilitating the easy extension to complex models and to non-parallel data settings is in our regularizer, which operates on the model parameters as opposed to their inferences. Specifically, MIR was designed to reward model pairs whose translation tables respect the invertibility intuition.

We tested MIR against competitive baselines on two sequence alignment tasks: word alignment (with parallel data) and back-transliteration decipherment (without parallel data).

On Czech-English and Chinese-English word alignment (Section 5), restricted to the HMM model, MIR attains F- and B score improvements that are comparable to those of ABA and PostCAT. We further apply MIR beyond HMM, on the fertility-based IBM Models, showing further gains in F-score compared to the baseline, ABAandPostCAT. Interestingly, the HMM alignments obtained by ABA and MIR are qualitatively different, so that combining the two yields additive gains over each method by itself.

On English-Japanese back-transliteration decipherment (Section 6), we apply MIR to the cascade of wFSTs approach proposed by Ravi and Knight (2009). Using MIR, we are able to reduce the whole-name error-rate relative to a model trained on parallel data by 33%, as well as significantly outperform the joint model proposed by Mylonakis et al. (2007).

2 Background

We are concerned with learning generative models that describe transformations of a source-language sequence $\mathbf{e} = (e_1, \ldots, e_I)$ to a target-language sequence $\mathbf{f} = (f_1, \ldots, f_J)$. We consider two different data scenarios.

In the parallel data setting, each sample in the observed data consists of a pair $(\mathbf{e}, \mathbf{f})$. The generative story assigns the following probability to the event that $\mathbf{f}$ arises from $\mathbf{e}$:

$$p(\mathbf{f} \mid \mathbf{e}; \Theta) = \sum_{\mathbf{a}} p(\mathbf{a}, \mathbf{f} \mid \mathbf{e}; \Theta) \qquad (1)$$

where Θ denotes the model parameters and $\mathbf{a}$ denotes a hidden variable that corresponds to unknown choices taken in the generative process.

In the non-parallel data setting, only the target sequence $\mathbf{f}$ is observed and the source sequence $\mathbf{e}$ is hidden. The model assigns the following probability to the observed data:

$$p(\mathbf{f}; \Theta) = \sum_{\mathbf{e}} p(\mathbf{e}) \sum_{\mathbf{a}} p(\mathbf{a}, \mathbf{f} \mid \mathbf{e}). \qquad (2)$$

That is, the sequence $\mathbf{f}$ can arise from any sequence $\mathbf{e}$ by first selecting $\mathbf{e} \sim p(\mathbf{e})$ and then proceeding according to the parallel-data generative story (Eq. 1).

Unsupervised training of such models entails maximizing the data log-likelihood $L(\Theta)$:

$$\arg\max_{\Theta} \; L(\Theta) = \arg\max_{\Theta} \sum_{x \in X} \log p(\mathbf{x}; \Theta)$$

where $X = \{(\mathbf{e}^n, \mathbf{f}^n)\}_n$ in the parallel data setting and $X = \{(\mathbf{f}^n)\}_n$ in the non-parallel data setting.

Although the structure of Θ is unspecified, in practice, most models that follow these generative stories contain a word translation table (t-table) denoted t, with each parameter $t(f \mid e)$ representing the conditional probability of mapping a given source symbol e to a target symbol f.

3 Model Invertibility Regularization

In this section we propose a method for jointly training two word alignment models, a source-to-target model Θ_1 and a target-to-source model Θ_2, by regularizing their parameters to respect the invertibility of the alignment task. We therefore name our method *Model Invertibility Regularization* (MIR).

3.1 Regularizer

Our regularizer operates on the t-table parameters t_1, t_2 of the two models, as follows: Let matrices T_1, T_2 denote the t-tables t_1, t_2 in matrix form and consider their multiplication $T = T_1 T_2$. The resulting matrix T is a stochastic square matrix of dimension $|V_1| \times |V_1|$ where $|V_1|$ denotes the size of the source-language vocabulary. Each entry T_{ij} represents the total probability mass mapped from source word e_i to source word e_j by first applying the source-to-target mapping T_1 and then the target-to-source mapping T_2.

In particular, each diagonal entry T_{ii} holds the probability of mapping a source symbol back onto itself, a quantity we intuitively believe should be high. We therefore (initially) consider maximizing the trace of T:

$$\mathrm{Tr}[T] = \sum_i T_{ii} = \sum_e \sum_f t_1(f \mid e)\, t_2(e \mid f).$$

We further note that $\mathrm{Tr}[T] = \mathrm{Tr}[T_1 T_2] = \mathrm{Tr}[T_2 T_1]$, so that the trace captures equally well how much the target symbols map onto themselves.

Since T is stochastic, setting it to the identity matrix I maximizes its trace. In other words, the more T_1 and T_2 behave as (pseudo-)inverses of each other, the higher the trace is. This exactly fits with our intuition regarding invertibility.

Unfortunately, the trace is not concave in both T_1 and T_2, a property which will become desirable in optimization. We therefore modify the trace regularizer by applying the entrywise square root operator on T_1, T_2 and denote the new term R:

$$\begin{aligned} R(t_1, t_2) &= \mathrm{Tr}\left[\sqrt{T_1}\,\sqrt{T_2}\right] \\ &= \sum_e \sum_f \sqrt{t_1(f \mid e)\, t_2(e \mid f)}. \end{aligned} \tag{3}$$

Note that R is maximized when $\sqrt{T_1}\,\sqrt{T_2} = I$.

Concavity of R in both t_1, t_2 (or equivalently T_1, T_2) follows by observing that it is a sum of concave functions – each term in the summation is a geometric mean, which is concave in its parameters.

3.2 Joint Objective Function

We apply `MIR` in two data scenarios: In the parallel data setting, we observe N sequence pairs $\{\mathbf{x}_1^n\}_n = \{(\mathbf{e}^n, \mathbf{f}^n)\}_n$ or, equivalently, $\{\mathbf{x}_2^n\}_n = \{(\mathbf{f}^n, \mathbf{e}^n)\}_n$.

In the non-parallel setting, two monolingual datasets are observed: N_1 source sequences $\{\mathbf{x}_1^n\}_n = \{\mathbf{e}^n\}_n$ and N_2 target sequences $\{\mathbf{x}_2^n\}_n = \{\mathbf{f}^n\}_n$.

The probability of the nth sample under the kth model Θ_k (for $k \in \{1, 2\}$) is denoted $p_k(\mathbf{x}_k^n; \Theta_k)$. Specifically, in the parallel data setting, the probability of $\mathbf{x}_k^n$ under its model is:[1]

$$\begin{aligned} p_1(\mathbf{x}_1^n; \Theta_1) &= p(\mathbf{f}^n \mid \mathbf{e}^n; \Theta_1) \\ p_2(\mathbf{x}_2^n; \Theta_2) &= p(\mathbf{e}^n \mid \mathbf{f}^n; \Theta_2) \end{aligned}$$

[1]This slight notational abuse helps represent both data scenarios succinctly.

whereas in the non-parallel data setting, the probability is defined as:

$$\begin{aligned} p_1(\mathbf{x}_1^n; \Theta_1) &= p(\mathbf{f}^n; \Theta_1) \\ p_2(\mathbf{x}_2^n; \Theta_2) &= p(\mathbf{e}^n; \Theta_2). \end{aligned}$$

Using the above definitions and the `MIR` regularizer R (Eq. 3), we formulate an optimization program for maximizing the regularized log-likelihoods of the observed data:

$$\max_{\Theta_1, \Theta_2} \; \lambda R(t_1, t_2) + \sum_{k \in \{1,2\}} \sum_{n=1}^{N_k} \log p_k(\mathbf{x}_k^n; \Theta_k) \tag{4}$$

where $\lambda \geq 0$ is a tunable hyperparameter (note that, in the parallel case, $N = N_1 = N_2$).

We defer discussion on the relationship and merits of our approach with respect to `ABA` (Liang et al., 2006) and `PostCAT` (Ganchev et al., 2008) to Section 4.

3.3 Optimization Procedure

Using our concave regularizer, `MIR` optimization (Eq. 4) neatly falls under the MAP-EM framework (Dempster et al., 1977) and inherits the convergence properties of the underlying algorithms. MAP-EM follows the same structure as standard EM: The E-step remains identical to the standard E-step, while the M-step maximizes the complete-data log-likelihood plus the regularization term. In the case of `MIR`, the E-step can be carried out independently for each model. The only extra work is in the M-step, which optimizes a single (concave) objective function.

Specifically, let $\mathbf{z}_n$ denote the missing data, where, in the parallel data setting, only the alignment is missing ($\mathbf{z}_k^n = \mathbf{a}_k^n$) and in the non-parallel data setting, both alignment and source symbol are missing ($\mathbf{z}_1^n = (\mathbf{a}_1^n, \mathbf{e}^n), \mathbf{z}_2^n = (\mathbf{a}_2^n, \mathbf{f}^n)$).

In the E-step, each model Θ_k (for $k \in \{1, 2\}$) is held fixed and its posterior distribution over the missing data $\mathbf{z}_k^n$ is computed per each observation, $\mathbf{x}_k^n$:

$$q_k(\mathbf{z}_k^n, \mathbf{x}_k^n) := p_k(\mathbf{z}_k^n \mid \mathbf{x}_k^n; \Theta_k).$$

In the M-step, the computed posteriors are used to define a convex optimization program that max-

imizes the regularized sum of expected complete-data log-likelihoods:

$$\max_{\Theta_1,\Theta_2} \lambda R(t_1, t_2) + \sum_{k\in\{1,2\}} \sum_{n=1}^{N_k} q_k(\mathbf{z}_k^n, \mathbf{x}_k^n) \log p_k(\mathbf{x}_k^n, \mathbf{z}_k^n)$$

where n ranges over the appropriate sample set.

Operationally, for models Θ_k that can be encoded as wFSTs (such as the IBM1, IBM2 and HMM word alignment models), the E-step can be carried out efficiently and exactly using dynamic programming (Eisner, 2002). Other models resort to approximation techniques – for example, the fertility-based word alignment models apply hill-climbing and sampling heuristics in order to efficiently estimate the posteriors (Brown et al., 1993)

From the computed posteriors q_k we collect expected counts for each event, used to construct the M-step optimization objective. Since the `MIR` regularizer couples only the t-table parameters, the update rule for any remaining parameter is left unchanged (that is, one can use the usual closed-form count-and-divide solution).

Now, let $\mathsf{C}_1^{e,f}$ and $\mathsf{C}_2^{e,f}$ denote the expected counts for the t-table parameters. That is, $\mathsf{C}_k^{e,f}$ denotes the expected number of times a source-symbol type e is seen aligned to a target-symbol type f according to the posterior q_k. In the M-step, we maximize the following objective with respect to t_1 and t_2:

$$\arg\max_{t_1,t_2} \sum_{e,f} \mathsf{C}_1^{e,f} \log t_1(f \mid e) + \sum_{e,f} \mathsf{C}_2^{e,f} \log t_2(e \mid f) + \lambda R(t_1, t_2) \quad (5)$$

which can be efficiently solved using convex programming techniques due to the concavity of R and the complete-data log-likelihoods in both t_1 and t_2.

In our implementation, we applied Projected Gradient Descent (Bertsekas, 1999; Schoenemann, 2011), where at each step, the parameters are updated in the direction of the M-step objective gradient at (t_1, t_2) and then projected back onto the probability simplex. We used simple stopping conditions based on objective function value convergence and a bounded number of iterations.

4 Baselines

4.1 Parallel Data Baseline: `ABA` and `PostCAT`

Our approach is most similar to Alignment by Agreement (Liang et al., 2006) which uses a single joint objective for two word alignment models. The difference between our objective (Eq. 4) and theirs lies in their proposed regularizer, which rewards the per-sample agreement of the two models' alignment posteriors:

$$\sum_n \log \sum_{\mathbf{z}} p_1(\mathbf{z} \mid \mathbf{x}^n) \cdot p_2(\mathbf{z} \mid \mathbf{x}^n)$$

where $\mathbf{x}^n = (\mathbf{e}^n, \mathbf{f}^n)$ and where $\mathbf{z}$ ranges over the possible alignments between $\mathbf{e}^n$ and $\mathbf{f}^n$ (practically, only over 1-to-1 alignments, since each model is only capable of producing one-to-many alignments).

Liang et al. (2006) note that proper EM optimization of their regularized joint objective leads to an intractable E-step. Unable to exactly and efficiently compute alignment posteriors, they resort to a product-of-marginals heuristic which breaks EM's convergence guarantees, but has a closed-form solution and works well in practice.

`MIR` regularization has both theoretical and practical advantages compared to `ABA`, which make our method more convenient and broadly applicable:

1. By regularizing for posterior agreement, `ABA` is restricted to a parallel data setting, whereas `MIR` can be applied even without parallel data.

2. The posteriors of more advanced word alignment models (such as fertility-based models) do not correspond to alignments, and furthermore, are already estimated with approximation techniques. Thus, even if we somehow adapt `ABA`'s product-of-marginals heuristic to such models, we run the risk of estimating highly inaccurate posteriors (specifically, zero-valued posteriors). In contrast, `MIR` extends to all IBM-style word alignment models and does not add heuristics. The M-step computation can be done exactly and efficiently with convex optimization.

3. `MIR` provides the same theoretical convergence guarantees as the underlying algorithms.

Ganchev et al. (2008) propose `PostCAT` which uses Posterior Regularization (Ganchev et al., 2010)

to enforce posterior agreement between the two models. Specifically, they add a KL-projection step after the E-step of the EM algorithm which returns the posterior $q(z \mid x)$ closest in KL-Divergence to an E-step posterior, but which also upholds certain constraints. The particular constraints they suggest encode alignment agreement *in expectation* between the two models' posteriors. For details, the reader can refer to (Ganchev et al., 2008).

Similarly to ABA, with their suggested alignment agreement constraints PostCAT cannot be applied without parallel data and it is unclear how to extend it to fertility based models (however, it does seems possible to apply other constraints using the general posterior regularization framework).

We compare MIR against ABA and PostCAT in Section 5.

4.2 Non-Parallel Data Baseline: bi-EM

Mylonakis et al. (2007) cast the two directional models as a single joint model by reparameterization and normalization. That is, both directional models, consisting of a t-table only, are reparameterized as:

$$t_1(f \mid e) = \frac{\beta_{e,f}}{\sum_f \beta_{e,f}} \qquad t_2(e \mid f) = \frac{\beta_{e,f}}{\sum_e \beta_{e,f}} \tag{6}$$

They then maximize the likelihood of observed *monolingual* sequences from both languages:

$$\max_{\beta} L_1(\{\mathbf{f}^n\};\beta) + L_2(\{\mathbf{e}^n\};\beta) \tag{7}$$

where, for example:

$$\begin{aligned} L_1(\{\mathbf{f}^n\};\beta) &= \log \prod_n p(\mathbf{f}^n) \\ &= \log \prod_n \sum_{\mathbf{e}} p(\mathbf{f}^n \mid \mathbf{e}) p(\mathbf{e}) \\ &= \log \prod_n \sum_{\mathbf{e}} p(\mathbf{e}) \prod_m t_1(f_m^n \mid \mathbf{e}) \end{aligned}$$

Here, $p(\mathbf{e})$ denotes the probability of $\mathbf{e}$ according to a fixed source language model.

Once training of β is complete, we can decode an observed target sequence $\mathbf{f}$, by casting β back in terms of t_1 and apply the Viterbi decoding algorithm.

To solve for β in Eq. 7, Mylonakis et al. (2007) propose bi-EM, an iterative EM-style algorithm. The objective function in their M-step is not concave, hinting that a closed-form solution for the maximizer is unlikely. The probability estimate that they use in the M-step appears to maximize an approximation of their M-step objective which omits the normalization factors in Eq. 7.

Nevertheless, bi-EM attains improved results compared to standard EM on both POS-tagging and monotone noun sequence translation without parallel data. We compare MIR against bi-EM in Sec. 6.

5 Experiments with Parallel Data

In this section, we compare MIR against standard EM training and ABA on Czech-English and Chinese-English word alignment and translation.

5.1 Implementation and Code

For ABA[2] and PostCAT[3] training we used the authors' implementation, which supports the HMM model. Vanilla EM training was done using GIZA++,[4] which supports all IBM models as well as HMM. Our method MIR was implemented on top of GIZA++.[5]

5.2 Data

We used the following parallel data to train the word alignment models:

Chinese-English: 287K sentence pairs from the NIST 2009 Open MT Evaluation constrained task consisting of 5.3M and 6.6M tokens, respectively.

Czech-English: 85K sentence pairs from the News Commentary corpus, consisting of 1.6M and 1.8M tokens, respectively.

Sentence length was restricted to at most 40 tokens.

5.3 Word Alignment Experiments

We obtained HMM alignments by running either 5 or 10 iterations (optimized on a held-out validation set) of both IBM Model 1 and HMM. We obtained IBM Model 4 alignments by continuing with 5 iterations of IBM Model 3 and 10 iterations of IBM

[2] http://cs.stanford.edu/~pliang/software/cross-em-aligner-1.3.zip

[3] http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html

[4] http://code.google.com/p/giza-pp/

[5] https://github.com/vaswani/MIR_ALIGNMENT

Method	Chi-Eng **Align F1**	Cze-Eng **Align F1**
EM-HMM	64.6	65.0
PostCAT-HMM	69.8	69.6
ABA-HMM	70.8	70.4
MIR-HMM	70.9	69.6
EM-IBM4	68.4	67.3
MIR-IBM4	72.9	70.7

Table 1: Word alignment F1 scores.

Model 4. We then extracted symmetrized alignments in the following manner: For all HMM models, we used the *posterior decoding* technique described in Liang et al. (2006) as implemented by each package. For IBM Model 4, we used the standard grow-diag-final-and (gdfa) symmetrization heuristic (Koehn et al., 2003).

We tuned MIR's λ parameter to maximize alignment F-score on a validation set of 460 hand-aligned Czech-English and 1102 Chinese-English sentences.

Alignment F-scores are reported in Table 1. In particular, the best results were obtained by MIR, when applied to the fertility based IBM4 model - we obtained gains of +2.1% (Chinese-English) and +0.3% (Czech-English) compared to the best competitor.

5.4 MT Experiments

We ran MT experiments using the Moses (Koehn et al., 2007) phrase-based translation system.[6] The feature weights were trained discriminatively using MIRA (Chiang et al., 2008), and we used a 5-gram language model trained on the Xinhua portion of English Gigaword (LDC2007T07). All other parameters remained with their default settings. The development data used for discriminative training were: for Chinese-English, data from the NIST 2004 and NIST 2006 test sets; for Czech-English, 2051 sentences from the WMT 2010 shared task. We used case-insensitive IBM B (closest reference length) as our metric.

On both language pairs, ABA, PostCAT and MIR outperform their respective EM baseline with comparable gains overall. However, we noticed that ABA and MIR are not producing the same alignments.

[6] http://www.statmt.org/moses/

Method	Chi-Eng **NIST08**	Cze-Eng **WMT09**	Cze-Eng **WMT10**
EM-HMM	23.6	16.7	17.1
PostCAT-HMM	24.6	16.9	17.4
MIR-HMM	24.0	17.1	17.6
ABA-HMM	24.4	17.1	17.7
EM-IBM4	24.2	16.8	17.2
MIR-IBM4	24.6	17.2	17.5
ABA + MIR-HMM	25.1	17.4	17.9

Table 2: B scores. Combining ABA and MIR HMM alignments improves B score significantly over all other methods.

For example, by combining their HMM alignments (simply concatenating aligned bitexts) the total improvement reaches +1.5 B on the Chinese-to-English task, a statistically significant improvement ($p < 0.05$) according to a bootstrap resampling significance test (Koehn, 2004)). Table 5.4 summarizes our MT results.

6 Experiments without Parallel Data

Ravi and Knight (2009) consider the challenging task of learning a Japanese-English back-transliteration model without parallel data. The goal is to correctly decode a list of 100 US senator names written in katakana script, without having access to parallel data. In this section, we reproduce their decipherment experiment and show that applying MIR to their baseline model significantly outperforms both the baseline and the bi-EM method.

6.1 Phonetic-Based Japanese Decipherment

Ravi and Knight (2009) construct a English-to-Japanese transliteration model as a cascade of wFSTs (depicted in Figure 1, top). According to their generative story, any word in katakana is generated by re-writing an English word in its English phonetic representation, which is then transformed to a Japanese phonetic representation and finally re-written in katakana script. For example, the word "computer" is mapped to a sequence of 8 English phonemes (k, ah, m, p, y, uw, t, er), which is mapped to a sequence of 9 Japanese phonemes (K, O, N, P, Y, U, T, A, A) and finally to Katakana.

They apply their trained transliteration model to decode a list of 100 US senator names and report a

whole-name error-rate (WNER)[7] of 40% with parallel data (trained over 3.3k word pairs), compared to 73% WNER without parallel data (trained over 9.5k Japanese words only), demonstrating the weakness of methods that do not use parallel data.

6.2 Forward Pipeline

We reproduced the English-to-Japanese transliteration pipeline of Ravi and Knight (2009) by constructing each of the cascade wFSTs as follows:

1. A unigram language model (LM) of English terms, estimated over the top 40K most frequent capitalized words found in the Gigaword corpus (without smoothing).

2. An English pronunciation wFST from the CMU pronunciation dictionary.[8]

3. An English-to-Japanese phoneme mapping wFST that encodes a phoneme t-table t_1 which was designed according to the best setting reported by Ravi and Knight (2009). Specifically, t_1 is restricted to either 1-to-1 or 1-to-2 phoneme mappings and maintains consonant parity. See further details in their paper.

4. A hand-built Japanese pronunciation to Katakana wFST (Ravi and Knight, 2009).

6.3 Backward Pipeline

`MIR` requires a pipeline in the reverse direction, transliteration of Japanese to English. We constructed a unigram LM of Katakana terms over the top 25K most frequent Katakana words found in the Japanese `2005-2008-news` dictionary from the Leipzig corpora.[9]

The remaining required wFSTs were obtained by inverting the forward model wFSTs (that is, wFSTs 2,3,4 above), and the cascade was composed in the reverse direction. In particular, by inverting t_1, we obtained the Japanese-to-English t-table t_2 that allows only 2-to-1 or 1-to-1 phoneme mappings.

[7] The percentage of names where any error occurs anywhere in either the first or last name.

[8] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

[9] `http://corpora.uni-leipzig.de/`

6.4 Training Data

For training data, we used the top 50% most frequent terms from the monolingual data over which we constructed the LM wFSTs. This resulted in a set of 20K English terms (denoted `ENG`) and a set of 13K Japanese terms in Katakana (denoted `KTKN`).

Taking the entire set of monolingual terms led to poor baseline results, probably since uncommon English terms are not transliterated, and uncommon Katakana terms may be borrowed from languages other than English.

In any case, it is important to note that `ENG` and `KTKN` are unrelated, since both were collected over non-parallel corpora.

6.5 Training and Tuning

We train and tune 4 models:

`baseline`: the model proposed by Ravi and Knight (2009), which maximizes the likelihood (Eq. 2) of the observed Japanese terms `KTKN`.

`MIR`: Our bidirectional, regularized model, which maximizes the regularized likelihoods (Eq. 4) of both monolingual corpora `ENG`, `KTKN`.

`bi-EM`: The joint model proposed by Mylonakis et al. (2007), which maximizes the likelihoods (Eq. 7) of both monolingual corpora `ENG`, `KTKN`.

`Oracle`: As an upper bound, we train the model of Ravi and Knight (2009) as if it was given the correct English origin for each Japanese term. (over 4.2K *parallel* English-Japanese phoneme sequences).

We train each method for 15 EM iterations, while keeping the LM and pronunciation wFSTs fixed.

Training was done using the Carmel finite-state toolkit.[10] Specifically, `baseline` and `oracle` rely on Carmel exclusively, while for `MIR` and `bi-EM`, we manipulated Carmel to output the E-step posteriors, which we then used to construct and solve the M-step objective using our own implementation.

The different models were tuned over a development set consisting of 50 frequent Japanese terms and their English origin. For each method, we chose the so-called stretch-factor $\alpha \in \{1, 2, 3\}$ used to exponentiate the model parameters before decoding

[10] `http://www.isi.edu/licensed-sw/carmel/`

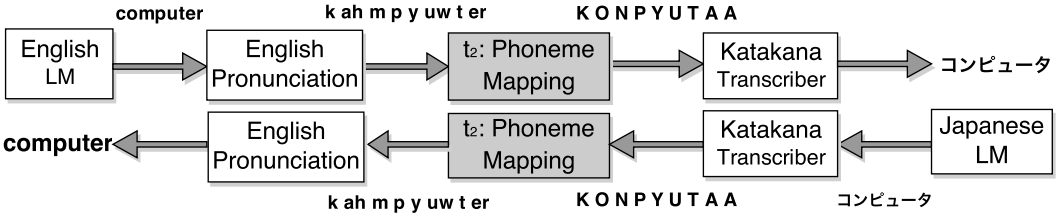

Figure 1: The transliteration generative story as a cascade of wFSTs. Each box represents a transducer. **Top:** transliteration of the word "computer" to Japanese Katakana. **Bottom:** the reverse process. `MIR` jointly trains the two cascades by maximizing the regularized data log-likelihood with respect to the two (shaded) phoneme mapping models t_1, t_2.

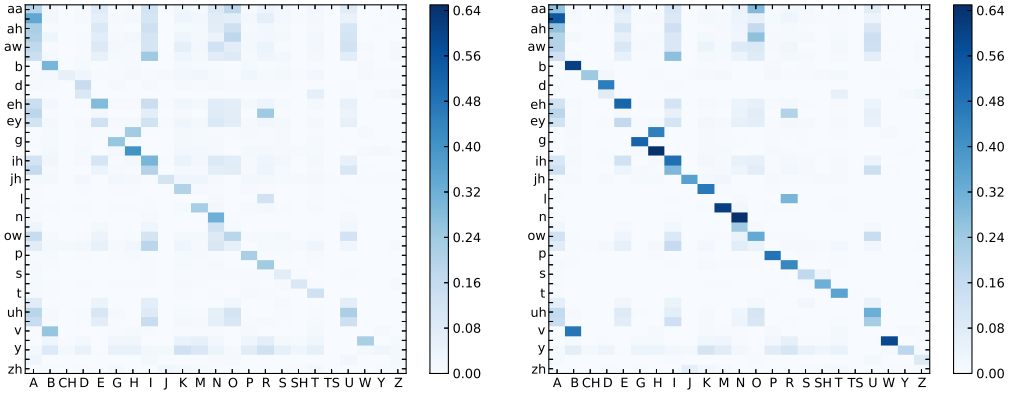

Figure 2: The 1-to-1 mapping submatrix of the t_1 transliteration table for independent training (left) and `MIR` (right). `MIR` learns sparser, peaked models compared to those learned by independent training.

(see Ravi and Knight (2009)), our model's hyperparameter $\lambda \in \{1, 2, 3, 4\}$, and the number of iterations (up to 15) to minimize WNER on the development set.

We decoded Japanese terms using the Viterbi algorithm, applied on the selected t_1 model (using Eq. 6 to convert the `bi-EM` model β back to to t_1). Finally, note that `ABA` training and symmetrization decoding heuristics are inapplicable, since they rely on parallel data.

6.6 Senator Name Decoding Results

We compiled our own test set, consisting of 100 US senator names (first and last), and compared the performance of the four algorithms. Table 3 reports WNER, average normalized edit distance (NED) and the number of model parameters (t_1) with value greater than 0.01 as an indication of sparsity. Figure 2 further compares the 1-to-1 portions of the best model learned by the `baseline` method with the best model learned by `MIR`, showing the difference in parameter sparsity.

	WNER	NED	$t_1 > 0.01$
`baseline`	67%	23.2	649
`bi-EM`	66%	21.8	600
`MIR`	**59%**	**17.3**	**421**
`Oracle`	43%	10.8	152

Table 3: `MIR` reduces error rates (WNER, NED) and learns sparser models (number of t_1 parameters greater than 0.01) compared to the other models.

Using `MIR`, we obtained significant reduction in error rates, closing the gap between the `baseline` method and `Oracle`, which was trained on parallel data, by 33% in WNER and nearly 50% in NED. This error reduction clearly demonstrates the efficacy of `MIR` in the non-parallel data setting.

7 Conclusion

We presented Model Invertibility Regularization (MIR), an unsupervised method for jointly training bidirectional sequence alignment models with or without parallel data. Our formulation is based on the simple observation that the alignment tasks at hand are inherently invertible and encourages the translation tables in both models to behave like pseudo-inverses of each other.

We derived an efficient MAP-EM algorithm and demonstrated our method's effectiveness on two different alignment tasks. On word alignment, applying MIR on the IBM4 model yielded the highest F scores and the resulting B scores were comparable to that of Alignment by Agreement (Liang et al., 2006) and PostCAT (Ganchev et al., 2008). Our best MT results (up to +1.5 B improvement) were obtained by combining alignments from both MIR and ABA, indicating that the two methods learn complementary alignments. On Japanese-English back-transliteration with no parallel data, we obtained a significant error reduction over two baseline methods (Ravi and Knight, 2009; Mylonakis et al., 2007).

As future work, we plan to apply MIR on large-scale MT decipherment (Ravi and Knight, 2011; Dou and Knight, 2013), where, so far, only a single directional model has been used. Another promising direction is to encourage invertibility not only between words, but between their senses and synonyms.

Acknowledgements

We would like to thank Markos Mylonakis and Khalil Sima'an for their help in understanding the derivation of their bi-EM method. This work was partially supported by DARPA grants DOI/NBC D12AP00225 and HR0011-12-C-0014 and a Google Faculty Research Award to Chiang.

References

Dimitri P. Bertsekas. 1999. *Nonlinear Programming*. Athena Scientific.

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

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of EMNLP*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Computational Linguistics*, 39(4):1–38.

Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *EMNLP*, pages 1668–1676. ACL.

Jason Eisner. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kuzman Ganchev, João V. Graça, and Ben Taskar. 2008. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio, June. Association for Computational Linguistics.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL*.

Markos Mylonakis, Khalil Sima'an, and Rebecca Hwa. 2007. Unsupervised estimation for noisy-channel models. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 665–672, New York, NY, USA. ACM.

Sujith Ravi and Kevin Knight. 2009. Learning phoneme mappings for transliteration without parallel data. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 37–45, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 12–21, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Schoenemann. 2011. Probabilistic word alignment under the L_0-norm. In *Proceedings of CoNLL.*