Minimum Imputed Risk: Unsupervised Discriminative Training for Machine Translation

Zhifei Li*
Google Research
Mountain View, CA 94043, USA
zhifei.work@gmail.com

Jason Eisner
Johns Hopkins University
Baltimore, MD 21218, USA
eisner@jhu.edu

Ziyuan Wang, Sanjeev Khudanpur
Johns Hopkins University
Baltimore, MD 21218, USA
zwang40,khudanpur@jhu.edu

Brian Roark
Oregon Health & Science University
Beaverton, Oregon 97006, USA
roark@cslu.ogi.edu

Abstract

Discriminative training for machine translation has been well studied in the recent past. A limitation of the work to date is that it relies on the availability of high-quality in-domain bilingual text for supervised training. We present an unsupervised discriminative training framework to incorporate the usually plentiful target-language monolingual data by using a rough "reverse" translation system. Intuitively, our method strives to ensure that probabilistic "round-trip" translation from a target-language sentence to the source-language and back will have low expected loss. Theoretically, this may be justified as (discriminatively) minimizing an *imputed empirical risk*. Empirically, we demonstrate that augmenting supervised training with unsupervised data improves translation performance over the supervised case for both IWSLT and NIST tasks.

1 Introduction

Missing data is a common problem in statistics when fitting the parameters θ of a model. A common strategy is to attempt to impute, or "fill in," the missing data (Little and Rubin, 1987), as typified by the EM algorithm. In this paper we develop imputation techniques when θ is to be trained *discriminatively*.

We focus on machine translation (MT) as our example application. A Chinese-to-English machine translation system is given a Chinese sentence x and asked to predict its English translation y. This system employs statistical models $p_\theta(y \mid x)$ whose parameters θ are discriminatively trained using bilingual sentence pairs (x, y). But bilingual data for such supervised training may be relatively scarce for a particular language pair (e.g., Urdu-English), especially for some topics (e.g., technical manuals) or genres (e.g., blogs). So systems seek to exploit additional monolingual data, i.e., a corpus of English sentences y with no corresponding source-language sentences x, to improve estimation of θ. This is our missing data scenario.[1]

Discriminative training of the parameters θ of $p_\theta(y \mid x)$ using monolingual English data is a curious idea, since there is no Chinese input x to translate. We propose an unsupervised training approach, called *minimum imputed risk training*, which is conceptually straightforward: *First guess x (probabilistically) from the observed y using a reverse English-to-Chinese translation model $p_\phi(x \mid y)$. Then train the discriminative Chinese-to-English model $p_\theta(y \mid x)$ to do a good job at translating this imputed x back to y, as measured by a given performance metric.* Intuitively, our method strives to ensure that probabilistic "round-trip" translation from a target-language sentence to the source-language and back again will have low expected loss.

Our approach can be applied in an application scenario where we have (1) enough out-of-domain bilingual data to build two baseline translation systems, with parameters θ for the forward direction, and ϕ for the reverse direction; (2) a small amount

* Zhifei Li is currently working at Google Research, and this work was done while he was a PHD student at Johns Hopkins University.

[1] Contrast this with traditional semi-supervised training that looks to exploit "unlabeled" inputs x, with missing *outputs* y.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 920–929,
Edinburgh, Scotland, UK, July 27–31, 2011.

of in-domain bilingual development data to discriminatively tune a *small* number of parameters in ϕ; and (3) a large amount of in-domain English monolingual data.

The novelty here is to exploit (3) to *discriminatively* tune the parameters θ of all *translation* model components,[2] $p_\theta(y|x)$ and $p_\theta(y)$, not merely train a generative *language* model $p_\theta(y)$, as is the norm.

Following the theoretical development below, the empirical effectiveness of our approach is demonstrated by replacing a key *supervised* discriminative training step in the development of large MT systems — learning the log-linear combination of several component model scores (viewed as features) to optimize a performance metric (e.g. BLEU) on a set of (x, y) pairs — with our *unsupervised* discriminative training using only y. One may hence contrast our approach with the traditional *supervised* methods applied to the MT task such as minimum error rate training (Och, 2003; Macherey et al., 2008), the averaged Perceptron (Liang et al., 2006), maximum conditional likelihood (Blunsom et al., 2008), minimum risk (Smith and Eisner, 2006; Li and Eisner, 2009), and MIRA (Watanabe et al., 2007; Chiang et al., 2009).

We perform experiments using the open-source MT toolkit **Joshua** (Li et al., 2009a), and show that adding unsupervised data to the traditional supervised training setup improves performance.

2 Supervised Discriminative Training via Minimization of Empirical Risk

Let us first review discriminative training in the *supervised* setting—as used in MERT (Och, 2003) and subsequent work.

One wishes to tune the parameters θ of some complex translation system $\delta_\theta(x)$. The function δ_θ, which translates Chinese x to English $y = \delta_\theta(x)$ need not be probabilistic. For example, θ may be the parameters of a scoring function used by δ, along with pruning and decoding heuristics, for extracting a high-scoring translation of x.

The goal of discriminative training is to minimize the expected loss of $\delta_\theta(\cdot)$, under a given task-specific **loss function** $\mathrm{L}(y', y)$ that measures how bad it would be to output y' when the correct output is y. For an MT system that is judged by the BLEU metric (Papineni et al., 2001), for instance, $\mathrm{L}(y', y)$ may be the negated BLEU score of y' w.r.t. y. To be precise, the goal[3] is to find θ with low **Bayes risk**,

$$\theta^* = \operatorname*{argmin}_\theta \sum_{x,y} p(x, y)\, \mathrm{L}(\delta_\theta(x), y) \qquad (1)$$

where $p(x, y)$ is the joint distribution of the input-output pairs.[4]

The true $p(x, y)$ is, of course, not known and, in practice, one typically minimizes **empirical risk** by replacing $p(x, y)$ above with the empirical distribution $\tilde{p}(x, y)$ given by a supervised training set $\{(x_i, y_i), i = 1, \ldots, N\}$. Therefore,

$$\begin{aligned} \theta^* &= \operatorname*{argmin}_\theta \sum_{x,y} \tilde{p}(x, y)\, \mathrm{L}(\delta_\theta(x), y) \\ &= \operatorname*{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), y_i). \qquad (2) \end{aligned}$$

The search for θ^* typically requires the use of numerical methods and some regularization.[5]

3 Unsupervised Discriminative Training with Missing Inputs

3.1 Minimization of Imputed Risk

We now turn to the unsupervised case, where we have training examples $\{y_i\}$ but not their corresponding inputs $\{x_i\}$. We cannot compute the summand $\mathrm{L}(\delta_\theta(x_i), y_i)$ for such i in (2), since $\delta_\theta(x_i)$ requires to know x_i. So we propose to replace

[2] Note that the extra monolingual data is used only for tuning the model weights, but not for inducing new phrases or rules.

[3] This goal is different from the minimum risk training of Li and Eisner (2009) in a subtle but important way. In both cases, θ^* minimizes *risk* or *expected loss*, but the expectation is w.r.t. different distributions: the expectation in Li and Eisner (2009) is under the conditional distribution $p(y \mid x)$, while the expectation in (1) is under the joint distribution $p(x, y)$.

[4] In the terminology of statistical decision theory, $p(x, y)$ is a distribution over states of nature. We seek a *decision rule* $\delta_\theta(x)$ that will incur low expected loss on *observations* x that are generated from unseen states of nature.

[5] To compensate for the shortcut of using the unsmoothed empirical distribution rather than a posterior estimate of $p(x, y)$ (Minka, 2000), it is common to add a regularization term $||\theta||_2^2$ in the objective of (2). The regularization term can prevent overfitting to a training set that is not large enough to learn all parameters.

$\mathrm{L}(\delta_\theta(x_i), y_i)$ with the expectation

$$\sum_x p_\phi(x \mid y_i)\, \mathrm{L}(\delta_\theta(x), y_i), \quad (3)$$

where $p_\phi(\cdot \mid \cdot)$ is a "reverse prediction model" that attempts to impute the missing x_i data. We call the resulting variant of (2) the minimization of *imputed empirical risk*, and say that

$$\theta^* = \operatorname*{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid y_i)\, \mathrm{L}(\delta_\theta(x), y_i) \quad (4)$$

is the estimate with the **minimum imputed risk**[6].

The minimum imputed risk objective of (4) could be evaluated by *brute force* as follows.

1. For each unsupervised example y_i, use the reverse prediction model $p_\phi(\cdot \mid y_i)$ to impute possible reverse translations $\mathcal{X}_i = \{x_{i1}, x_{i2}, \ldots\}$, and add each (x_{ij}, y_i) pair, weighted by $p_\phi(x_{ij} \mid y_i) \leq 1$, to an imputed training set .
2. Perform the supervised training of (2) on the *imputed* and *weighted* training data.

The second step means that we must use δ_θ to forward-translate each imputed x_{ij}, evaluate the loss of the translations y'_{ij} against the corresponding true translation y_i, and choose the θ that minimizes the weighted sum of these losses (i.e., the empirical risk when the empirical distribution $\tilde{p}(x, y)$ is derived from the imputed training set). Specific to our MT task, this tries to ensure that probabilistic "round-trip" translation, from the target-language sentence y_i to the source-language and back again, will have a low expected loss.[7]

The trouble with this method is that the reverse model p_ϕ generates a weighted lattice or hypergraph $\mathcal{X}_i$ encoding exponentially many translations of y_i, and it is computationally infeasible to forward-translate *each* $x_{ij} \in \mathcal{X}_i$. We therefore investigate several approximations to (4) in Section 3.4.

[6] One may exploit both supervised data $\{(x_i, y_i)\}$ and unsupervised data $\{y_j\}$ to perform semi-supervised training via an interpolation of (2) and (4). We will do so in our experiments.

[7] Our approach may be applied to other tasks as well. For example, in a speech recognition task, δ_θ is a speech recognizer that produces text, whereas p_ϕ is a speech *synthesizer* that must produce a distribution over audio (or at least over acoustic features or phone sequences) (Huang et al., 2010).

3.2 The Reverse Prediction Model p_ϕ

A *crucial* ingredient in (4) is the reverse prediction model $p_\phi(\cdot|\cdot)$ that attempts to impute the missing x_i. We will train this model in advance, doing the best job we can from available data, including any out-of-domain bilingual data as well as any in-domain monolingual data[8] x.

In the MT setting, δ_θ and p_ϕ may have similar parameterization. One translates Chinese to English; the other translates English to Chinese.

Yet the setup is not quite symmetric. Whereas δ_θ is a translation *system* that aims to produce a *single*, *low-loss* translation, the reverse version p_ϕ is rather a probabilistic *model*. It is supposed to give an accurate probability distribution over possible values x_{ij} of the missing input sentence x_i. All of these values are taken into account in (4), regardless of the loss that they would incur if they were evaluated for translation quality relative to the missing x_i.

Thus, ϕ does not need to be trained to minimize the risk itself (so there is no circularity). Ideally, it should be trained to match the underlying conditional distribution of x given y, by achieving a low conditional cross-entropy

$$\mathrm{H}(X \mid Y) = -\sum_{x,y} p(x, y) \log p_\phi(x \mid y). \quad (5)$$

In practice, ϕ is trained by (empirically) minimizing $-\frac{1}{M}\sum_{j=1}^{N} \log p_\phi(x_j \mid y_j) + \frac{1}{2\sigma^2}\|\phi\|_2^2$ on some bilingual data, with the regularization coefficient σ^2 tuned on held out data.

It may be tolerable for p_ϕ to impute mediocre translations x_{ij}. All that is necessary is that the (forward) translations generated from the imputed x_{ij} "simulate" the competing hypotheses that we would see when translating the correct Chinese input x_i.

3.3 The Forward Translation System δ_θ and The Loss Function $\mathbf{L}(\delta_\theta(x_i), y_i)$

The minimum empirical risk objective of (2) is quite general and various popular supervised training methods (Lafferty et al., 2001; Collins, 2002; Och, 2003; Crammer et al., 2006; Smith and Eisner,

[8] In a translation task from x to y, one usually does not make use of in-domain monolingual data x. But we *can* exploit x to train a language model $p_\phi(x)$ for the reverse translation system, which will make the imputed x_{ij} look like true Chinese inputs.

2006) can be formalized in this framework by choosing different functions for δ_θ and $\mathrm{L}(\delta_\theta(x_i), y_i)$. The generality of (2) extends to our minimum imputed risk objective of (4). Below, we specify the δ_θ and $\mathrm{L}(\delta_\theta(x_i), y_i)$ we considered in our investigation.

3.3.1 Deterministic Decoding

A simple translation rule would define

$$\delta_\theta(x) = \operatorname*{argmax}_y p_\theta(y \mid x) \tag{6}$$

If this $\delta_\theta(x)$ is used together with a loss function $\mathrm{L}(\delta_\theta(x_i), y_i)$ that is the negated BLEU score[9], our minimum imputed risk objective of (4) is equivalent to MERT (Och, 2003) *on the imputed training data*.

However, this would not yield a differentiable objective function. Infinitesimal changes to θ could result in discrete changes to the winning output string $\delta_\theta(x)$ in (6), and hence to the loss $\mathrm{L}(\delta_\theta(x), y_i)$. Och (2003) developed a specialized line search to perform the optimization, which is not scalable when the number of model parameters θ is large.

3.3.2 Randomized Decoding

Instead of using the argmax of (6), we assume *during training* that $\delta_\theta(x)$ is itself random, i.e. the MT system *randomly* outputs a translation y with probability $p_\theta(y \mid x)$. As a result, we will modify our objective function of (4) to take yet another expectation over the unknown y. Specifically, we will replace $\mathrm{L}(\delta_\theta(x), y_i)$ in (4) with

$$\sum_y p_\theta(y \mid x)\, \mathrm{L}(y, y_i). \tag{7}$$

Now, the minimum imputed empirical risk objective of (4) becomes

$$\theta^* = \operatorname*{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x,y} p_\phi(x \mid y_i)\, p_\theta(y \mid x)\, \mathrm{L}(y, y_i) \tag{8}$$

If the loss function $\mathrm{L}(y, y_i)$ is a negated BLEU, this is equivalent to performing minimum-risk training described by (Smith and Eisner, 2006; Li and Eisner, 2009) *on the imputed data*.[10]

[9] One can manipulate the loss function to support other methods that use deterministic decoding, such as Perceptron (Collins, 2002) and MIRA (Crammer et al., 2006).

[10] Again, one may manipulate the loss function to support other probabilistic methods that use randomized decoding, such as CRFs (Lafferty et al., 2001).

The objective function in (8) is now differentiable, since each coefficient $p_\theta(y \mid x)$ is a differentiable function of θ, and thus amenable to optimization by gradient-based methods; we use the L-BFGS algorithm (Liu et al., 1989) in our experiments. We perform experiments with the syntax-based MT system **Joshua** (Li et al., 2009a), which implements dynamic programming algorithms for second-order expectation semirings (Li and Eisner, 2009) to efficiently compute the gradients needed for optimizing (8).

3.4 Approximating $p_\phi(x \mid y_i)$

As mentioned at the end of Section 3.1, it is computationally infeasible to forward-translate *each* of the imputed reverse translations x_{ij}. We propose four approximations that are computationally feasible. Each may be regarded as a different approximation of $p_\phi(x \mid y_i)$ in equations (4) or (8).

k-best. For each y_i, add to the imputed training set only the k most probable translations $\{x_{i1}, \ldots x_{ik}\}$ according to $p_\phi(x \mid y_i)$. (These can be extracted from $\mathcal{X}_i$ using standard algorithms (Huang and Chiang, 2005).) Rescale their probabilities to sum to 1.

Sampling. For each y_i, add to the training set k independent samples $\{x_{i1}, \ldots x_{ik}\}$ from the distribution $p_\phi(x \mid y_i)$, each with weight $1/k$. (These can be sampled from $\mathcal{X}_i$ using standard algorithms (Johnson et al., 2007).) This method is known in the literature as *multiple imputation* (Rubin, 1987).

Lattice. [11] Under certain special cases it is be possible to compute the expected loss in (3) exactly via dynamic programming. Although $\mathcal{X}_i$ does contain exponentially many translations, it may use a "packed" representation in which these translations share structure. This representation may furthermore enable sharing work in forward-translation, so as to efficiently translate the entire set $\mathcal{X}_i$ and obtain a distribution over translations y. Finally, the expected loss under that distribution, as required by equation (3), may also be efficiently computable.

All this turns out to be possible if (a) the posterior distribution $p_\phi(x \mid y_i)$ is represented by an *un-*

[11] The lattice approximation is presented here as a theoretical contribution, and we do not empirically evaluate it since its implementation requires extensive engineering effort that is beyond the main scope of this paper.

ambiguous weighted finite-state automaton $\mathcal{X}_i$, (b) the forward translation system δ_θ is structured in a certain way as a weighted synchronous context-free grammar, and (c) the loss function decomposes in a certain way. We omit the details of the construction as beyond the scope of this paper.

In our experimental setting described below, (b) is true (using **Joshua**), and (c) is true (since we use a loss function presented by Tromble et al. (2008) that is an approximation to BLEU and is decomposable). While (a) is not true in our setting because $\mathcal{X}_i$ is a hypergraph (which is ambiguous), Li et al. (2009b) show how to *approximate* a hypergraph representation of $p_\phi(x \mid y_i)$ by an unambiguous WFSA. One could then apply the construction to this WFSA[12], obtaining an approximation to (3).

Rule-level Composition. Intuitively, the reason why the structure-sharing in the hypergraph $\mathcal{X}_i$ (generated by the reverse system) cannot be exploited during forward translating is that when the forward Hiero system translates a string $x_i \in \mathcal{X}_i$, it must parse it into recursive phrases.

But the structure-sharing within the hypergraph of $\mathcal{X}_i$ has already parsed x_i into recursive phrases, in a way determined by the reverse Hiero system; each translation phrase (or rule) corresponding to a hyperedge. To exploit structure-sharing, we can use a forward translation system that decomposes according to that existing parse of x_i. We can do that by considering *only* forward translations that respect the hypergraph structure of $\mathcal{X}_i$. The simplest way to do this is to require complete isomorphism of the SCFG trees used for the reverse and forward translations. In other words, this does round-trip imputation (i.e., from y to x, and then to y') at the rule level. This is essentially the approach taken by Li et al. (2010).

3.5 The Log-Linear Model p_θ

We have not yet specified the form of p_θ. Following much work in MT, we begin with a linear model

$$\text{score}(x, y) = \theta \cdot f(x, y) = \sum_k \theta_k f_k(x, y) \quad (9)$$

where $f(x, y)$ is a feature vector indexed by k. Our deterministic *test-time* translation system δ_θ simply outputs the highest-scoring y for fixed x. At *training time*, our randomized decoder (Section 3.3.2) uses the Boltzmann distribution (here a log-linear model)

$$p_\theta(y \mid x) = \frac{e^{\gamma \cdot \text{score}(x,y)}}{Z(x)} = \frac{e^{\gamma \cdot \text{score}(x,y)}}{\sum_{y'} e^{\gamma \cdot \text{score}(x,y')}} \quad (10)$$

The scaling factor γ controls the sharpness of the training-time distribution, i.e., the degree to which the randomized decoder favors the highest-scoring y. For large γ, our training objective approaches the imputed risk of the deterministic test-time system while remaining differentiable.

In a task like MT, in addition to the input x and output y, we often need to introduce a *latent* variable d to represent the hidden derivation that relates x to y. A derivation d represents a particular *phrase segmentation* in a phrase-based MT system (Koehn et al., 2003) and a *derivation tree* in a typical syntax-based system (Galley et al., 2006; Chiang, 2007). We change our model to assign scores not to an (x, y) pair but to the detailed derivation d; in particular, now the function f that extracts a feature vector can look at all of d. We replace y by d in (9)–(10), and finally define $p_\theta(y|x)$ by marginalizing out d,

$$p_\theta(y \mid x) = \sum_{d \in \mathrm{D}(x,y)} p_\theta(d \mid x) \quad (11)$$

where $\mathrm{D}(x, y)$ represents the set of derivations that yield x and y.

4 Minimum Imputed Risk vs. EM

The notion of imputing missing data is familiar from other settings (Little and Rubin, 1987), particularly the expectation maximization (EM) algorithm, a widely used generative approach. So it is instructive to compare EM with minimum imputed risk.

One can estimate θ by maximizing the log-likelihood of the data $\{(x_i, y_i), i = 1, \ldots, N\}$ as

$$\operatorname*{argmax}_\theta \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i, y_i). \quad (12)$$

If the x_i's are missing, EM tries to iteratively maximize the marginal probability:

$$\operatorname*{argmax}_\theta \frac{1}{N} \sum_{i=1}^{N} \log \sum_x p_\theta(x, y_i). \quad (13)$$

[12] Note that the forward translation of a WFSA is tractable by using a lattice-based decoder such as that by Dyer et al. (2008).

The E-step of each iteration comprises computing $\sum_x p_{\theta_t}(x \mid y_i) \log p_\theta(x, y_i)$, the *expected* log-likelihood of the complete data, where $p_{\theta_t}(x \mid y_i)$ is the conditional part of $p_{\theta_t}(x, y_i)$ under the current iterate θ_t, and the M-step comprises maximizing it:

$$\theta_{t+1} = \operatorname*{argmax}_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_{\theta_t}(x \mid y_i) \log p_\theta(x, y_i). \quad (14)$$

Notice that if we replace $p_{\theta_t}(x|y_i)$ with $p_\phi(x \mid y_i)$ in the equation above, and admit negated log-likelihood as a loss function, then the EM update (14) becomes identical to (4). In other words, the minimum imputed risk approach of Section 3.1 differs from EM in (i) using an externally-provided and static p_ϕ, instead of refining it at each iteration based on the current p_{θ_t}, and (ii) using a specific loss function, namely negated log-likelihood.

So why not simply use the maximum-likelihood (EM) training procedure for MT? One reason is that it is not discriminative: the loss function (e.g. negated BLEU) is ignored during training.

A second reason is that training good *joint* models $p_\theta(x, y)$ is computationally expensive. Contemporary MT makes heavy use of log-linear probability models, which allow the system designer to inject phrase tables, linguistic intuitions, or prior knowledge through a careful choice of features. Computing the objective function of (14) in closed form is difficult if p_θ is an arbitrary log-linear model, because the joint probability $p_\theta(x_i, y_i)$ is then defined as a ratio whose denominator Z_θ involves a sum over all possible sentence pairs (x, y) of any length.

By contrast, our discriminative framework will only require us to work with conditional models. While conditional probabilities such as $p_\phi(x \mid y)$ and $p_\theta(y \mid x)$ are also ratios, computing their denominators only requires us to sum over a packed forest of possible translations of a given y or x.[13]

In summary, EM would impute missing data using $p_\theta(x \mid y)$ and predict outputs using $p_\theta(y \mid x)$, both being conditional forms of the same joint model $p_\theta(x, y)$. Our minimum imputed risk training method is similar, but it instead uses a pair of separately parameterized, separately trained models $p_\phi(x \mid y)$ and $p_\theta(y \mid x)$. By sticking to conditional models, we can efficiently use more sophisticated model features, and we can incorporate the loss function when we train θ, which should improve both efficiency and accuracy at test time.

[13] Analogously, discriminative CRFs have become more popular than generative HMMs because they permit efficient training even with a wide variety of log-linear features (Lafferty et al., 2001).

5 Experimental Results

We report results on Chinese-to-English translation tasks using **Joshua** (Li et al., 2009a), an open-source implementation of Hiero (Chiang, 2007).

5.1 Baseline Systems

5.1.1 IWSLT Task

We train both reverse and forward baseline systems. The translation models are built using the corpus for the IWSLT 2005 Chinese to English translation task (Eck and Hori, 2005), which comprises 40,000 pairs of transcribed utterances in the travel domain. We use a *5-gram* language model with modified Kneser-Ney smoothing (Chen and Goodman, 1998), trained on the English (resp. Chinese) side of the bitext. We use a standard training pipeline and pruning settings recommended by (Chiang, 2007).

5.1.2 NIST Task

For the NIST task, the TM is trained on about 1M parallel sentence pairs (about 28M words in each language), which are sub-sampled from corpora distributed by LDC for the NIST MT evaluation using a sampling method implemented in **Joshua**. We also used a 5-gram language model, trained on a data set consisting of a 130M words in English Gigaword (LDC2007T07) and the bitext's English side.

5.2 Feature Functions

We use two classes of features f_k for discriminative training of p_θ as defined in (9).

5.2.1 Regular Hiero Features

We include ten features that are standard in Hiero (Chiang, 2007). In particular, these include one baseline language model feature, three baseline translation models, one word penalty feature, three features to count how many rules with an arity of

zero/one/two are used in a derivation, and two features to count how many times the unary and binary glue rules in Hiero are used in a derivation.

5.2.2 Target-rule Bigram Features

In this paper, we do not attempt to discriminatively tune a separate parameter for each bilingual rule in the Hiero grammar. Instead, we train several hundred features that generalize across these rules.

For each bilingual rule, we extract bigram features over the target-side symbols (including non-terminals and terminals). For example, if a bilingual rule's target-side is "*on the* X_1 *issue of* X_2" where X_1 and X_2 are non-terminals (with a position index), we extract the bigram features *on the*, *the* X, X *issue*, *issue of*, and *of* X. (Note that the position index of a non-terminal is ignored in the feature.) Moreover, for the terminal symbols, we will use their dominant POS tags (instead of the symbol itself). For example, the feature *the* X becomes *DT* X. We use 541 such bigram features for IWSLT task (and 1023 such features for NIST task) that fire frequently.

5.3 Data Sets for Discriminative Training

5.3.1 IWSLT Task

In addition to the 40,000 sentence pairs used to train the baseline generative models (which are used to compute the features f_k), we use three bilingual data sets listed in Table 1, also from IWSLT, for discriminative training: one to train the reverse model p_ϕ (which uses only the 10 standard Hiero features as described in Section 5.2.1),[14] one to train the forward model δ_θ (which uses both classes of features described in Section 5.2, i.e., 551 features in total), and one for test.

Note that the reverse model ϕ is always trained using the supervised data of Dev_ϕ, while the forward model θ may be trained in a supervised or semi-supervised manner, as we will show below.

In all three data sets, each Chinese sentence x_i has 16 English reference translations, so each y_i is actually a *set* of 16 translations. When we impute data from y_i (in the semi-supervised scenario), we actually impute 16 different values of x_i, by using p_ϕ to separately reverse translate each sentence in y_i. This effectively adds 16 pairs of the form (x_i, y_i) to the training set (see section 3.4), where each x_i is a different input sentence (imputed) in each case, but y_i is always the original set of 16 references.

Data set	Purpose	# of sentences	
		Chinese	English
Dev_ϕ	training ϕ	503	503×16
Dev_θ	training θ	503*	503×16
Eval_θ	testing	506	506×16

Table 1: **IWSLT Data sets used for discriminative training/test**. Dev_ϕ is used for discriminatively training of the reverse model ϕ, Dev_θ is for the forward model, and Eval_θ is for testing. The star * for Dev_θ emphasizes that some of its Chinese side will not be used in the training (see Table 2 for details).

5.3.2 NIST Task

For the NIST task, we use MT03 set (having 919 sentences) to tune the component parameters in both the forward and reverse baseline systems. Additionally, we use the English side of MT04 (having 1788 sentences) to perform semi-supervised tuning of the forward model. The test sets are MT05 and MT06 (having 1082 and 1099 sentences, respectively). In all the data sets, each source sentence has four reference translations.

5.4 Main Results

We compare two training scenarios: supervised and semi-supervised. The supervised system ("Sup") carries out discriminative training on a bilingual data set. The semi-supervised system ("+Unsup") additionally uses some monolingual English text for discriminative training (where we impute one Chinese translation per English sentence).

Tables 2 and 3 report the results for the two tasks under two training scenarios. Clearly, adding unsupervised data improves over the supervised case, by at least 1.3 BLEU points in IWSLT and 0.5 BLEU in NIST.

5.5 Results for Analysis Purposes

Below, we will present more results on the IWSLT data set to help us understand the behavior of the

[14] Ideally, we should train ϕ to minimize the conditional cross-entropy (5) as suggested in section 3.2. In the present results, we trained ϕ discriminatively to minimize risk, purely for ease of implementation using well versed steps.

Training scenario	Test BLEU
Sup, (200, 200×16)	47.6
+Unsup, 101×16 Eng sentences	49.0
+Unsup, 202×16 Eng sentences	48.9
+Unsup, 303×16 Eng sentences	49.7*

Table 2: **BLEU scores for semi-supervised training for IWSLT task.** The supervised system ("Sup") is trained on a subset of Dev_θ containing 200 Chinese sentences and 200×16 English translations. "+Unsup" means that we include additional (monolingual) English sentences from Dev_θ for semi-supervised training; for each English sentence, we impute the 1-best Chinese translation. A star * indicates a result that is signicantly better than the "Sup" baseline (paired permutation test, $p < 0.05$).

Training scenario	Test BLEU	
	MT05	MT06
Sup, (919, 919×4)	32.4	30.6
+Unsup, 1788 Eng sentences	33.0*	31.1*

Table 3: **BLEU scores for semi-supervised training for NIST task.** The "Sup" system is trained on MT03, while the "+Unsup" system is trained with additional 1788 English sentences from MT04. (Note that while MT04 has 1788×4 English sentences as it has four sets of references, we only use one such set, for computational efficiency of discriminative training.) A star * indicates a result that is signicantly better than the "Sup" baseline (paired permutation test, $p < 0.05$).

methods proposed in this paper.

5.5.1 Imputation with Different Reverse Models

A critical component of our unsupervised method is the reverse translation model $p_\phi(x \mid y)$. We wonder how the performance of our unsupervised method changes when the quality of the reverse system varies. To study this question, we used two different reverse translation systems, one with a language model trained on the Chinese side of the bitext ("WLM"), and the other one without using such a Chinese LM ("NLM"). Table 4 (in the fully unsupervised case) shows that the imputed Chinese translations have a far lower BLEU score without the language model,[15] and that this costs us about 1 English BLEU point in the forward translations. Still, even with the worse imputation (in the case of "NLM"), our forward translations improve as we add more monolingual data.

Data size	Imputed-CN BLEU		Test-EN BLEU	
	WLM	NLM	WLM	NLM
101	11.8	3.0	48.5	46.7
202	11.7	3.2	48.9	47.6
303	13.4	3.5	48.8	47.9

Table 4: **BLEU scores for unsupervised training with/without using a language model in the reverse system.** A data size of 101 means that we use only the English sentences from a subset of Dev_θ containing 101 Chinese sentences and 101×16 English translations; for each English sentence we impute the 1-best Chinese translation. "WLM" means a Chinese language model is used in the reverse system, while "NLM" means no Chinese language model is used. In addition to reporting the BLEU score on Eval_θ, we also report "Imputed-CN BLEU", the BLEU score of the imputed Chinese sentences against their corresponding Chinese reference sentences.

5.5.2 Imputation with Different k-best Sizes

In all the experiments so far, we used the reverse translation system to impute only a single Chinese translation for each English monolingual sentence. This is the 1-best approximation of section 3.4.

Table 5 shows (in the fully unsupervised case) that the performance does not change much as k increases.[16] This may be because that the 5-best sentences are likely to be quite similar to one another (May and Knight, 2006). Imputing a longer k-best list, a sample, or a lattice for x_i (see section 3.4) might achieve more diversity in the training inputs, which might make the system more robust.

6 Conclusions

In this paper, we present an unsupervised discriminative training method that works with missing inputs. The key idea in our method is to use a reverse model to impute the missing input from the observed output. The training will then forward translate the imputed input, and choose the parameters of the forward model such that the imputed risk (i.e.,

[15] The BLEU scores are low even *with* the language model because only one Chinese reference is available for scoring.

[16] In the present experiments, however, we simply weighted all k imputed translations equally, rather than in proportion to their posterior probabilities as suggested in Section 3.4.

Training scenario	Test BLEU
Unsup, k=1	48.5
Unsup, k=2	48.4
Unsup, k=3	48.9
Unsup, k=4	48.5
Unsup, k=5	48.4

Table 5: **BLEU scores for unsupervised training with different k-best sizes.** We use 101×16 monolingual English sentences, and for each English sentence we impute the k-best Chinese translations using the reverse system.

the expected loss of the forward translations with respect to the observed output) is minimized. This matches the intuition that the probabilistic "round-trip" translation from the target-language sentence to the source-language and back should have low expected loss.

We applied our method to two Chinese to English machine translation tasks (i.e. IWSLT and NIST). We showed that augmenting supervised data with unsupervised data improved performance over the supervised case (for both tasks).

Our discriminative model used only a small amount of training data and relatively few features. In future work, we plan to test our method in settings where there are large amounts of monolingual training data (enabling many discriminative features). Also, our experiments here were performed on a language pair (i.e., Chinese to English) that has quite rich bilingual resources in the domain of the test data. In future work, we plan to consider low-resource test domains and language pairs like Urdu-English, where bilingual data for novel domains is sparse.

Acknowledgements

This work was partially supported by NSF Grants No IIS-0963898 and No IIS-0964102 and the DARPA GALE Program. The authors thank Markus Dreyer, Damianos Karakos and Jason Smith for insightful discussions.

References

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *ACL*, pages 200–208.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *NAACL*, pages 218–226.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *ACL*, pages 1012–1020.

Matthias Eck and Chiori Hori. 2005. Overview of the iwslt 2005 evaluation campaign. In *In IWSLT*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL*, pages 961–968.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *IWPT*, pages 53–64.

Jui-Ting Huang, Xiao Li, and Alex Acero. 2010. Discriminative training methods for language models using conditional entropy criteria. In *ICASSP*.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *NAACL*, pages 139–146.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*, pages 48–54.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *EMNLP*, pages 40–51.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Jonathan Weese, and Omar. Zaidan. 2009a. Joshua: An open source toolkit for parsing-based machine translation. In *WMT09*, pages 26–30.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009b. Variational decoding for statistical machine translation. In *ACL*, pages 593–601.

Zhifei Li, Ziyuan Wang, Sanjeev Khudanpur, and Jason Eisner. 2010. Unsupervised discriminative language

model training for machine translation using simulated confusion sets. In *COLING*, pages 556–664.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *ACL*, pages 761–768.

R. J. A. Little and D. B. Rubin. 1987. *Statistical Analysis with Missing Data*. J. Wiley & Sons, New York.

Dong C. Liu, Jorge Nocedal, and Dong C. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *EMNLP*, pages 725–734.

Jonathan May and Kevin Knight. 2006. A better n-best list: practical determinization of weighted finite tree automata. In *NAACL*, pages 351–358.

Thomas Minka. 2000. Empirical risk minimization is an incomplete inductive principle. In *MIT Media Lab note*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

D. B. Rubin. 1987. *Multiple Imputation for Nonresponse in Surveys*. J. Wiley & Sons, New York.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *ACL*, pages 787–794.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum-Bayes-risk decoding for statistical machine translation. In *EMNLP*, pages 620–629.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *EMNLP-CoNLL*, pages 764–773.