Lattice Minimum Bayes-Risk Decoding for Statistical Machine Translation

Roy W. Tromble[1] and **Shankar Kumar**[2] and **Franz Och**[2] and **Wolfgang Macherey**[2]

[1]Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
royt@jhu.edu

[2] Google Inc.
1600 Amphitheatre Pkwy.
Mountain View, CA 94043, USA
{shankarkumar,och,wmach}@google.com

Abstract

We present Minimum Bayes-Risk (MBR) decoding over translation lattices that compactly encode a huge number of translation hypotheses. We describe conditions on the loss function that will enable efficient implementation of MBR decoders on lattices. We introduce an approximation to the BLEU score (Papineni et al., 2001) that satisfies these conditions. The MBR decoding under this approximate BLEU is realized using Weighted Finite State Automata. Our experiments show that the Lattice MBR decoder yields moderate, consistent gains in translation performance over N-best MBR decoding on Arabic-to-English, Chinese-to-English and English-to-Chinese translation tasks. We conduct a range of experiments to understand why Lattice MBR improves upon N-best MBR and study the impact of various parameters on MBR performance.

1 Introduction

Statistical language processing systems for speech recognition, machine translation or parsing typically employ the Maximum A Posteriori (MAP) decision rule which optimizes the 0-1 loss function. In contrast, these systems are evaluated using metrics based on string-edit distance (Word Error Rate), n-gram overlap (BLEU score (Papineni et al., 2001)), or precision/recall relative to human annotations. Minimum Bayes-Risk (MBR) decoding (Bickel and Doksum, 1977) aims to address this mismatch by selecting the hypothesis that minimizes the expected error in classification. Thus it directly incorporates the loss function into the decision criterion. The approach has been shown to give improvements over the MAP classifier in many areas of natural language processing including automatic speech recognition (Goel and Byrne, 2000), machine translation (Kumar and Byrne, 2004; Zhang and Gildea, 2008), bilingual word alignment (Kumar and Byrne, 2002), and parsing (Goodman, 1996; Titov and Henderson, 2006; Smith and Smith, 2007).

In statistical machine translation, MBR decoding is generally implemented by re-ranking an N-best list of translations produced by a first-pass decoder; this list typically contains between 100 and $10,000$ hypotheses. Kumar and Byrne (2004) show that MBR decoding gives optimal performance when the loss function is matched to the evaluation criterion; in particular, MBR under the sentence-level BLEU loss function (Papineni et al., 2001) gives gains on BLEU. This is despite the fact that the sentence-level BLEU loss function is an approximation to the exact corpus-level BLEU.

A different MBR inspired decoding approach is pursued in Zhang and Gildea (2008) for machine translation using Synchronous Context Free Grammars. A forest generated by an initial decoding pass is rescored using dynamic programming to maximize the expected count of synchronous constituents in the tree that corresponds to the translation. Since each constituent adds a new 4-gram to the existing translation, this approach approximately maximizes the expected BLEU.

In this paper we explore a different strategy to perform MBR decoding over *Translation Lattices* (Ueffing et al., 2002) that compactly encode a huge number of translation alternatives relative to an N-best list. This is a model-independent approach

in that the lattices could be produced by any statistical MT system — both phrase-based and syntax-based systems would work in this framework. We will introduce conditions on the loss functions that can be incorporated in Lattice MBR decoding. We describe an approximation to the BLEU score (Papineni et al., 2001) that will satisfy these conditions. Our Lattice MBR decoding is realized using Weighted Finite State Automata.

We expect Lattice MBR decoding to improve upon N-best MBR primarily because lattices contain many more candidate translations than the N-best list. This has been demonstrated in speech recognition (Goel and Byrne, 2000). We conduct a range of translation experiments to analyze lattice MBR and compare it with N-best MBR. An important aspect of our lattice MBR is the linear approximation to the BLEU score. We will show that MBR decoding under this score achieves a performance that is at least as good as the performance obtained under sentence-level BLEU score.

The rest of the paper is organized as follows. We review MBR decoding in Section 2 and give the formulation in terms of a gain function. In Section 3, we describe the conditions on the gain function for efficient decoding over a lattice. The implementation of lattice MBR with Weighted Finite State Automata is presented in Section 4. In Section 5, we introduce the corpus BLEU approximation that makes it possible to perform efficient lattice MBR decoding. An example of lattice MBR with a toy lattice is presented in Section 6. We present lattice MBR experiments in Section 7. A final discussion is presented in Section 8.

2 Minimum Bayes Risk Decoding

Minimum Bayes-Risk (MBR) decoding aims to find the candidate hypothesis that has the least expected loss under the probability model (Bickel and Doksum, 1977). We begin with a review of MBR decoding for Statistical Machine Translation (SMT).

Statistical MT (Brown et al., 1990; Och and Ney, 2004) can be described as a mapping of a word sequence F in the source language to a word sequence E in the target language; this mapping is produced by the MT decoder $\delta(F)$. If the reference translation E is known, the decoder performance can be measured by the loss function $L(E, \delta(F))$. Given such a loss function $L(E, E')$ between an automatic translation E' and the reference E, and an underlying probability model $P(E|F)$, the MBR decoder has the following form (Goel and Byrne, 2000; Kumar and Byrne, 2004):

$$\begin{aligned}\hat{E} &= \operatorname*{argmin}_{E' \in \mathcal{E}} R(E') \\ &= \operatorname*{argmin}_{E' \in \mathcal{E}} \sum_{E \in \mathcal{E}} L(E, E') P(E|F),\end{aligned}$$

where $R(E')$ denotes the Bayes risk of candidate translation E' under the loss function L.

If the loss function between any two hypotheses can be bounded: $L(E, E') \leq L_{max}$, the MBR decoder can be rewritten in terms of a gain function $G(E, E') = L_{max} - L(E, E')$:

$$\hat{E} = \operatorname*{argmax}_{E' \in \mathcal{E}} \sum_{E \in \mathcal{E}} G(E, E') P(E|F). \quad (1)$$

We are interested in performing MBR decoding under a sentence-level BLEU score (Papineni et al., 2001) which behaves like a gain function: it varies between 0 and 1, and a larger value reflects a higher similarity. We will therefore use Equation 1 as the MBR decoder.

We note that $\mathcal{E}$ represents the space of translations. For N-best MBR, this space $\mathcal{E}$ is the N-best list produced by a baseline decoder. We will investigate the use of a translation lattice for MBR decoding; in this case, $\mathcal{E}$ will represent the set of candidates encoded in the lattice.

In general, MBR decoding can use different spaces for hypothesis selection and risk computation: argmax and the sum in Equation 1 (Goel, 2001). As an example, the hypothesis could be selected from the N-best list while the risk is computed based on the entire lattice. Therefore, the MBR decoder can be more generally written as follows:

$$\hat{E} = \operatorname*{argmax}_{E' \in \mathcal{E}_h} \sum_{E \in \mathcal{E}_e} G(E, E') P(E|F), \quad (2)$$

where $\mathcal{E}_h$ refers to the *Hypothesis space* from where the translations are chosen, and $\mathcal{E}_e$ refers to the *Evidence space* that is used for computing the Bayes-risk. We will present experiments (Section 7) to show the relative importance of these two spaces.

3 Lattice MBR Decoding

We now present MBR decoding on translation lattices. A translation word lattice is a compact representation for very large N-best lists of translation hypotheses and their likelihoods. Formally, it is an acyclic Weighted Finite State Acceptor (WFSA) (Mohri, 2002) consisting of states and arcs representing transitions between states. Each arc is labeled with a word and a weight. Each path in the lattice, consisting of consecutive transitions beginning at the distinguished initial state and ending at a final state, expresses a candidate translation. Aggregation of the weights along the path[1] produces the weight of the path's candidate $H(E, F)$ according to the model. In our setting, this weight will imply the posterior probability of the translation E given the source sentence F:

$$P(E|F) = \frac{\exp\left(\alpha H(E,F)\right)}{\sum_{E' \in \mathcal{E}} \exp\left(\alpha H(E',F)\right)}. \quad (3)$$

The scaling factor $\alpha \in [0, \infty)$ flattens the distribution when $\alpha < 1$, and sharpens it when $\alpha > 1$.

Because a lattice may represent a number of candidates exponential in the size of its state set, it is often impractical to compute the MBR decoder (Equation 1) directly. However, if we can express the gain function G as a sum of *local* gain functions g_i, then we now show that Equation 1 can be refactored and the MBR decoder can be computed efficiently. We loosely call a gain function local if it can be applied to all paths in the lattice via WFSA intersection (Mohri, 2002) without significantly multiplying the number of states.

In this paper, we are primarily concerned with local gain functions that weight n-grams. Let $\mathcal{N} = \{w_1, \ldots, w_{|\mathcal{N}|}\}$ be the set of n-grams and let a local gain function $g_w : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$, for $w \in \mathcal{N}$, be as follows:

$$g_w(E, E') = \theta_w \#_w(E') \delta_w(E), \quad (4)$$

where θ_w is a constant, $\#_w(E')$ is the number of times that w occurs in E', and $\delta_w(E)$ is 1 if $w \in E$ and 0 otherwise. That is, g_w is θ_w times the number of occurrences of w in E', or zero if w does not occur in E. We first assume that the overall gain function $G(E, E')$ can then be written as a sum of local gain functions and a constant θ_0 times the length of the hypothesis E'.

$$\begin{aligned} G(E,E') &= \theta_0|E'| + \sum_{w \in \mathcal{N}} g_w(E,E') \quad (5) \\ &= \theta_0|E'| + \sum_{w \in \mathcal{N}} \theta_w \#_w(E') \delta_w(E) \end{aligned}$$

Given a gain function of this form, we can rewrite the risk (sum in Equation 1) as follows

$$\begin{aligned} &\sum_{E \in \mathcal{E}} G(E,E') P(E|F) \\ &= \sum_{E \in \mathcal{E}} \Big(\theta_0|E'| + \sum_{w \in \mathcal{N}} \theta_w \#_w(E') \delta_w(E)\Big) P(E|F) \\ &= \theta_0|E'| + \sum_{w \in \mathcal{N}} \theta_w \#_w(E') \sum_{E \in \mathcal{E}_w} P(E|F), \end{aligned}$$

where $\mathcal{E}_w = \{E \in \mathcal{E} | \delta_w(E) > 0\}$ represents the paths of the lattice containing the n-gram w at least once. The MBR decoder on lattices (Equation 1) can therefore be written as

$$\hat{E} = \operatorname*{argmax}_{E' \in \mathcal{E}} \Big\{\theta_0|E'| + \sum_{w \in \mathcal{N}} \theta_w \#_w(E') p(w|\mathcal{E})\Big\}. \quad (6)$$

Here $p(w|\mathcal{E}) = \sum_{E \in \mathcal{E}_w} P(E|F)$ is the posterior probability of the n-gram w in the lattice. We have thus replaced a summation over a possibly exponential number of items ($E \in \mathcal{E}$) with a summation over the number of n-grams that occur in $\mathcal{E}$, which is at worst polynomial in the number of edges in the lattice that defines $\mathcal{E}$. We compute the posterior probability of each n-gram w as:

$$p(w|\mathcal{E}) = \sum_{E \in \mathcal{E}_w} P(E|F) = \frac{Z(\mathcal{E}_w)}{Z(\mathcal{E})}, \quad (7)$$

where $Z(\mathcal{E}) = \sum_{E' \in \mathcal{E}} \exp(\alpha H(E', F))$ (denominator in Equation 3) and $Z(\mathcal{E}_w) = \sum_{E' \in \mathcal{E}_w} \exp(\alpha H(E', F))$. $Z(\mathcal{E})$ and $Z(\mathcal{E}_w)$ represent the sums[2] of weights of all paths in the lattices $\mathcal{E}_w$ and $\mathcal{E}$ respectively.

4 WFSA MBR Computations

We now show how the Lattice MBR Decision Rule (Equation 6) can be implemented using Weighted Finite State Automata (Mohri, 1997). There are four steps involved in decoding starting from weighted finite-state automata representing the candidate outputs of a translation system. We will describe these

[1] using the log semiring's *extend* operator

[2] in the log semiring, where $\log +(x, y) = \log(e^x + e^y)$ is the *collect* operator (Mohri, 2002)

steps in the setting where the evidence lattice $\mathcal{E}_e$ may be different from the hypothesis lattice $\mathcal{E}_h$ (Equation 2).

1. Extract the set of n-grams that occur in the evidence lattice $\mathcal{E}_e$. For the usual BLEU score, n ranges from one to four.

2. Compute the posterior probability $p(w|\mathcal{E})$ of each of these n-grams.

3. Intersect each n-gram w, with an appropriate weight (from Equation 6), to an initially unweighted copy of the hypothesis lattice $\mathcal{E}_h$.

4. Find the best path in the resulting automaton.

Computing the set of n-grams $\mathcal{N}$ that occur in a finite automaton requires a traversal, in topological order, of all the arcs in the automaton. Because the lattice is acyclic, this is possible. Each state q in the automaton has a corresponding set of n-grams $\mathcal{N}_q$ ending there.

1. For each state q, $\mathcal{N}_q$ is initialized to $\{\epsilon\}$, the set containing the empty n-gram.

2. Each arc in the automaton extends each of its source state's n-grams by its word label, and adds the resulting n-grams to the set of its target state. (ϵ arcs do not extend n-grams, but transfer them unchanged.) n-grams longer than the desired order are discarded.

3. $\mathcal{N}$ is the union over all states q of $\mathcal{N}_q$.

Given an n-gram, w, we construct an automaton matching any path containing the n-gram, and intersect that automaton with the lattice to find the set of paths containing the n-gram ($\mathcal{E}_w$ in Equation 7). Suppose $\mathcal{E}$ represent the weighted lattice, we compute[3]: $\mathcal{E}_w = \mathcal{E} \cap (\overline{w}\ w\ \Sigma^*)$, where $\overline{w} = (\overline{\Sigma^*\ w\ \Sigma^*})$ is the language that contains all strings that do not contain the n-gram w. The posterior probability $p(w|\mathcal{E})$ of n-gram w can be computed as a ratio of the total weights of paths in $\mathcal{E}_w$ to the total weights of paths in the original lattice (Equation 7).

For each n-gram $w \in \mathcal{N}$, we then construct an automaton that accepts an input E with weight equal to the product of the number of times the n-gram occurs in the input ($\#_w(E)$), the n-gram factor θ_w from Equation 6, and the posterior probability $p(w|\mathcal{E})$. The automaton corresponds to the weighted regular expression (Karttunen et al., 1996): $\bar{w}(w/(\theta_w p(w|\mathcal{E}))\ \bar{w})^*$.

We successively intersect each of these automata with an automaton that begins as an unweighted copy of the lattice $\mathcal{E}_h$. This automaton must also incorporate the factor θ_0 of each word. This can be accomplished by intersecting the unweighted lattice with the automaton accepting $(\Sigma/\theta_0)^*$. The resulting MBR automaton computes the total expected gain of each path. A path in this automaton that corresponds to the word sequence E' has cost: $\theta_0|E'| + \sum_{w\in\mathcal{N}} \theta_w \#_w(E) p(w|\mathcal{E})$ (expression within the curly brackets in Equation 6).

Finally, we extract the best path from the resulting automaton[4], giving the lattice MBR candidate translation according to the gain function (Equation 6).

5 Linear Corpus BLEU

Our Lattice MBR formulation relies on the decomposition of the overall gain function as a sum of local gain functions (Equation 5). We here describe a linear approximation to the log(BLEU score) (Papineni et al., 2001) which allows such a decomposition. This will enable us to rewrite the log(BLEU) as a linear function of n-gram matches and the hypothesis length. Our strategy will be to use a first order Taylor-series approximation to what we call the corpus log(BLEU) gain: the change in corpus log(BLEU) contributed by the sentence relative to not including that sentence in the corpus.

Let r be the reference length of the corpus, c_0 the candidate length, and $\{c_n|1 \leq n \leq 4\}$ the number of n-gram matches. Then, the corpus BLEU score $B(r, c_0, c_n)$ can be defined as follows (Papineni et al., 2001):

$$\begin{aligned}\log B &= \min\left(0, 1 - \frac{r}{c_0}\right) + \frac{1}{4}\sum_{n=1}^{4} \log \frac{c_n}{c_0 - \Delta_n}, \\ &\approx \min\left(0, 1 - \frac{r}{c_0}\right) + \frac{1}{4}\sum_{n=1}^{4} \log \frac{c_n}{c_0},\end{aligned}$$

where we have ignored Δ_n, the difference between the number of words in the candidate and the num-

[3] in the log semiring (Mohri, 2002)

[4] in the $(\max, +)$ semiring (Mohri, 2002)

ber of n-grams. If L is the average sentence length in the corpus, $\Delta_n \approx (n-1)\frac{c_0}{L}$.

The corpus log(BLEU) gain is defined as the change in log(BLEU) when a new sentence's (E') statistics are added to the corpus statistics:

$$G = \log B' - \log B,$$

where the counts in B' are those of B plus those for the current sentence. We will assume that the brevity penalty (first term in the above approximation) does not change when adding the new sentence. In experiments not reported here, we found that taking into account the brevity penalty at the sentence level can cause large fluctuations in lattice MBR performance on different test sets. We therefore treat only c_ns as variables.

The corpus log BLEU gain is approximated by a first-order vector Taylor series expansion about the initial values of c_n.

$$G \approx \sum_{n=0}^{N} (c'_n - c_n) \left.\frac{\partial \log B'}{\partial c'_n}\right|_{c'_n = c_n}, \quad (8)$$

where the partial derivatives are given by

$$\begin{aligned} \frac{\partial \log B}{\partial c_0} &= \frac{-1}{c_0}, \\ \frac{\partial \log B}{\partial c_n} &= \frac{1}{4c_n}. \end{aligned} \quad (9)$$

Substituting the derivatives in Equation 8 gives

$$G = \Delta \log B \approx -\frac{\Delta c_0}{c_0} + \frac{1}{4}\sum_{n=1}^{4} \frac{\Delta c_n}{c_n}, \quad (10)$$

where each $\Delta c_n = c'_n - c_n$ counts the statistic in the sentence of interest, rather than the corpus as a whole. This score is therefore a linear function in counts of words Δc_0 and n-gram matches Δc_n. Our approach ignores the count clipping present in the exact BLEU score where a correct n-gram present once in the reference but several times in the hypothesis will be counted only once as correct. Such an approach is also followed in Dreyer et al. (2007).

Using the above first-order approximation to gain in log corpus BLEU, Equation 9 implies that θ_0, θ_w from Section 3 would have the following values:

$$\begin{aligned} \theta_0 &= \frac{-1}{c_0} \\ \theta_w &= \frac{1}{4c_{|w|}}. \end{aligned} \quad (11)$$

5.1 N-gram Factors

We now describe how the n-gram factors (Equation 11) are computed. The factors depend on a set of n-gram matches and counts (c_n; $n \in \{0, 1, 2, 3, 4\}$). These factors could be obtained from a decoding run on a development set. However, doing so could make the performance of lattice MBR very sensitive to the actual BLEU scores on a particular run. We would like to avoid such a dependence and instead, obtain a set of parameters which can be estimated from multiple decoding runs without MBR. To achieve this, we make use of the properties of n-gram matches. It is known that the average n-gram precisions decay approximately exponentially with n (Papineni et al., 2001). We now assume that the number of matches of each n-gram is a constant ratio r times the matches of the corresponding $n-1$ gram.

If the unigram precision is p, we can obtain the n-gram factors ($n \in \{1, 2, 3, 4\}$) (Equation 11) as a function of the parameters p and r, and the number of unigram tokens T:

$$\begin{aligned} \theta_0 &= \frac{-1}{T} \\ \theta_n &= \frac{1}{4Tp \times r^{n-1}} \end{aligned} \quad (12)$$

We set p and r to the average values of unigram precision and precision ratio across multiple development sets. Substituting the above factors in Equation 6, we find that the MBR decision does not depend on T; therefore any value of T can be used.

6 An Example

Figure 1 shows a toy lattice and the final MBR automaton (Section 4) for BLEU with a maximum n-gram order of 2. We note that the MBR hypothesis ($bcde$) has a higher decoder cost relative to the MAP hypothesis ($abde$). However, $bcde$ gets a higher expected gain (Equation 6) than $abde$ since it shares more n-grams with the Rank-3 hypothesis ($bcda$). This illustrates how a lattice can help select MBR translations that can differ from the MAP translation.

7 Experiments

We now present experiments to evaluate MBR decoding on lattices under the linear corpus BLEU

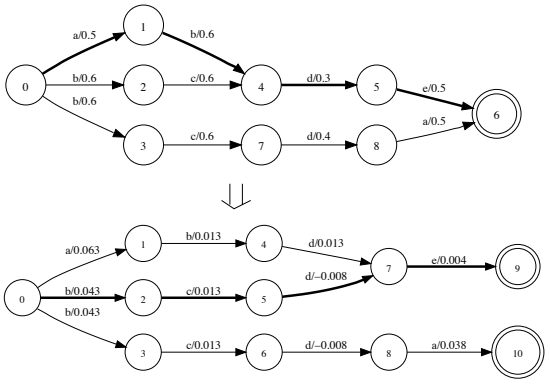

Figure 1: An example translation lattice with decoder costs (top) and its MBR Automaton for BLEU-2 (bottom). The bold path in the top is the MAP hypothesis and the bold path in the bottom is the MBR hypothesis. The precision parameters in Equation 12 are set to: $T = 10, p = 0.85, r = 0.72$.

Dataset	# of sentences		
	aren	zhen	enzh
dev1	1353	1788	1664
dev2	663	919	919
blind	1360	1357	1859

Table 1: Statistics over the development and test sets.

gain. We start with a description of the data sets and the SMT system.

7.1 Development and Blind Test Sets

We present our experiments on the constrained data track of the NIST 2008 Arabic-to-English (aren), Chinese-to-English (zhen), and English-to-Chinese (enzh) machine translation tasks.[5] In all language pairs, the parallel and monolingual data consists of all the allowed training sets in the constrained track.

For each language pair, we use two development sets: one for Minimum Error Rate Training (Och, 2003; Macherey et al., 2008), and the other for tuning the scale factor for MBR decoding. Our development sets consists of the NIST 2004/2003 evaluation sets for both aren and zhen, and NIST 2006 (NIST portion)/2003 evaluation sets for enzh. We report results on NIST 2008 which is our blind test set. Statistics computed over these data sets are reported in Table 1.

[5] http://www.nist.gov/speech/tests/mt/

7.2 MT System Description

Our phrase-based statistical MT system is similar to the alignment template system described in Och and Ney (2004). The system is trained on parallel corpora allowed in the constrained track. We first perform sentence and sub-sentence chunk alignment on the parallel documents. We then train word alignment models (Och and Ney, 2003) using 6 Model-1 iterations and 6 HMM iterations. An additional 2 iterations of Model-4 are performed for zhen and enzh pairs. Word Alignments in both source-to-target and target-to-source directions are obtained using the Maximum A-Posteriori (MAP) framework (Matusov et al., 2004). An inventory of phrase-pairs up to length 5 is then extracted from the union of source-target and target-source alignments. Several feature functions are then computed over the phrase-pairs. 5-gram word language models are trained on the allowed monolingual corpora. Minimum Error Rate Training under BLEU is used for estimating approximately 20 feature function weights over the dev1 development set.

Translation is performed using a standard dynamic programming beam-search decoder (Och and Ney, 2004) using two decoding passes. The first decoder pass generates either a lattice or an N-best list. MBR decoding is performed in the second pass. The MBR scaling parameter (α in Equation 3) is tuned on the dev2 development set.

7.3 Translation Results

We next report translation results from lattice MBR decoding. All results will be presented on the NIST 2008 evaluation sets. We report results using the NIST implementation of the BLEU score which computes the brevity penalty using the shortest reference translation for each segment (NIST, 2002 2008). The BLEU scores are reported at the word-level for aren and zhen but at the character level for enzh. We measure statistical significance using 95% confidence intervals computed with paired bootstrap resampling (Koehn, 2004). In all tables, systems in a column show statistically significant differences unless marked with an asterisk.

We first compare lattice MBR to N-best MBR decoding and MAP decoding (Table 2). In these experiments, we hold the likelihood scaling factor α a

	BLEU(%)		
	aren	zhen	enzh
MAP	43.7	27.9	41.4
N-best MBR	43.9	28.3*	42.0
Lattice MBR	44.9	28.5*	42.6

Table 2: Lattice MBR, N-best MBR & MAP decoding. On zhen, Lattice MBR and N-best MBR do not show statistically significant differences.

	BLEU(%)		
	aren	zhen	enzh
Lattice MBR, Lin. Corpus BLEU	44.2	28.1	42.2
N-best MBR, Sent. BLEU	43.9*	28.3*	42.0*
N-best MBR, Sent. Log BLEU	44.0*	28.3*	41.9*

Table 3: Lattice and N-best MBR (with Sentence BLEU/Sentence log BLEU) on a 1000-best list. In each column, entries with an asterisk do not show statistically significant differences.

constant; it is set to 0.2 for aren and enzh, and 0.1 for zhen. The translation lattices are pruned using Forward-Backward pruning (Sixtus and Ortmanns, 1999) so that the average numbers of arcs per word (lattice density) is 30. For N-best MBR, we use N-best lists of size 1000. To match the loss function, Lattice MBR is performed at the word level for aren/zhen and at the character level for enzh. Our lattice MBR is implemented using the Google OpenFst library.[6] In our experiments, p, r (Equation 12) have values of 0.85/0.72, 0.80/0.62, and 0.63/0.48 for aren, zhen, and enzh respectively.

We note that Lattice MBR provides gains of 0.2-1.0 BLEU points over N-best MBR, which in turn gives 0.2-0.6 BLEU points over MAP. These gains are obtained on top of a baseline system that has competitive performance relative to the results reported in the NIST 2008 Evaluation.[7] This demonstrates the effectiveness of lattice MBR decoding as a realization of MBR decoding which yields substantial gains over the N-best implementation.

The gains from lattice MBR over N-best MBR could be due to a combination of factors. These include: 1) better approximation of the corpus BLEU score, 2) larger hypothesis space, and 3) larger evidence space. We now present experiments to tease apart these factors.

Our first experiment restricts both the hypothesis and evidence spaces in lattice MBR to the 1000-best list (Table 3). We compare this to N-best MBR with: a) sentence-level BLEU, and b) sentence-level log BLEU.

The results show that when restricted to the 1000-best list, Lattice MBR performs slightly better than N-best MBR (with sentence BLEU) on aren/enzh while N-best MBR is better on zhen. We hypothesize that on aren/enzh, the linear corpus BLEU gain (Equation 10) is better correlated to the actual corpus BLEU than sentence-level BLEU while the opposite is true on zhen. N-best MBR gives similar results with either sentence BLEU or sentence log BLEU. This confirms that using a log BLEU score does not change the outcome of MBR decoding and further justifies our Taylor-series approximation of the log BLEU score.

		BLEU(%)		
Hyp Space	Evid Space	aren	zhen	enzh
Lattice	Lattice	44.9	28.5	42.6
1000-best	Lattice	44.6	28.5	42.6
Lattice	1000-best	44.1*	28.0*	42.1
1000-best	1000-best	44.2*	28.1*	42.2

Table 4: Lattice MBR with restrictions on hypothesis and evidence spaces. In each column, entries with an asterisk do not show statistically significant differences.

We next attempt to understand factors 2 and 3. To do that, we carry out lattice MBR when either the hypothesis or the evidence space in Equation 2 is restricted to 1000-best hypotheses (Table 4). For comparison, we also include results from lattice MBR when both hypothesis and evidence spaces are identical: either the full lattice or the 1000-best list (from Tables 2 and 3).

These results show that lattice MBR results are almost unchanged when the hypothesis space is restricted to a 1000-best list. However, when the evidence space is shrunk to a 1000-best list, there is a significant degradation in performance; these latter results are almost identical to the scenario when both evidence and hypothesis spaces are restricted to the 1000-best list. This experiment throws light on what makes lattice MBR effective over N-best MBR. Relative to the N-best list, the translation lattice provides a better estimate of the expected BLEU score. On the other hand, there are few hypotheses

[6] http://www.openfst.org/

[7] http://www.nist.gov/speech/tests/mt/2008/doc/mt08_official_results_v0.html

outside the 1000-best list which are selected by lattice MBR.

Finally, we show how the performance of lattice MBR changes as a function of the lattice density. The lattice density is the average number of arcs per word and can be varied using Forward-Backward pruning (Sixtus and Ortmanns, 1999). Figure 2 reports the average number of lattice paths and BLEU scores as a function of lattice density. The results show that Lattice MBR performance generally improves when the size of the lattice is increased. However, on zhen, there is a small drop beyond a density of 10. This could be due to low quality (low posterior probability) hypotheses that get included at the larger densities and result in a poorer estimate of the expected BLEU score. On aren and enzh, there are some gains beyond a lattice density of 30. These gains are relatively small and come at the expense of higher memory usage; we therefore work with a lattice density of 30 in all our experiments. We note that Lattice MBR is operating over lattices which are gigantic in comparison to the number of paths in an N-best list. At a lattice density of 30, the lattices in aren contain on an average about 10^{81} hypotheses!

7.4 Lattice MBR Scale Factor

We next examine the role of the scale factor α in lattice MBR decoding. The MBR scale factor determines the flatness of the posterior distribution (Equation 3). It is chosen using a grid search on the dev2 set (Table 1). Figure 3 shows the variation in BLEU scores on eval08 as this parameter is varied. The results show that it is important to tune this factor. The optimal scale factor is identical for all three language pairs. In experiments not reported in this paper, we have found that the optimal scaling factor on a moderately sized development set carries over to unseen test sets.

7.5 Maximum n-gram Order

Lattice MBR Decoding (Equation 6) involves computing a posterior probability for each n-gram in the lattice. We would like to speed up the Lattice MBR computation (Section 4) by restricting the maximum order of the n-grams in the procedure. The results (Table 5) show that on aren, there is no degradation if we limit the maximum order of the n-grams to 3. However, on zhen/enzh, there is improvement by considering 4-grams. We can therefore reduce Lattice MBR computations in aren.

	BLEU(%)		
Max n-gram order	aren	zhen	enzh
1	38.7	26.8	40.0
2	44.1	27.4	42.2
3	44.9	28.0	42.4
4	44.9	28.5	42.6

Table 5: Lattice MBR as a function of max n-gram order.

8 Discussion

We have presented a procedure for performing Minimum Bayes-Risk Decoding on translation lattices. This is a significant development in that the MBR decoder operates over a very large number of translations. In contrast, the current N-best implementation of MBR can be scaled to, at most, a few thousands of hypotheses. If the number of hypotheses is greater than, say 20,000, the N-best MBR becomes computationally expensive. The lattice MBR technique is efficient when performed over enormous number of hypotheses (up to 10^{80}) since it takes advantage of the compact structure of the lattice. Lattice MBR gives consistent improvements in translation performance over N-best MBR decoding, which is used in many state-of-the-art research translation systems. Moreover, we see gains on three different language pairs.

There are two potential reasons why Lattice MBR decoding could outperform N-best MBR: a larger hypothesis space from which translations could be selected or a larger evidence space for computing the expected loss. Our experiments show that the main improvement comes from the larger evidence space: a larger set of translations in the lattice provides a better estimate of the expected BLEU score. In other words, the lattice provides a better posterior distribution over translation hypotheses relative to an N-best list. This is a novel insight into the workings of MBR decoding. We believe this could be possibly employed when designing discriminative training approaches for machine translation. More generally, we have found a component in machine translation where the posterior distribution over hypotheses plays a crucial role.

We have shown the effect of the MBR scaling fac-

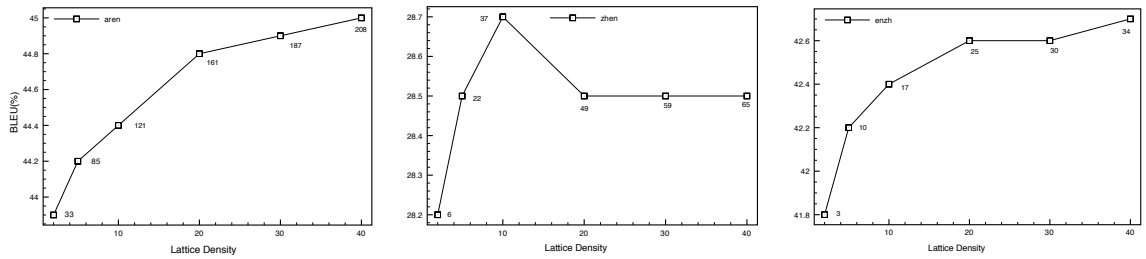

Figure 2: Lattice MBR vs. lattice density: aren/zhen/enzh. Each point also shows the $\log_e$(Avg. # of paths).

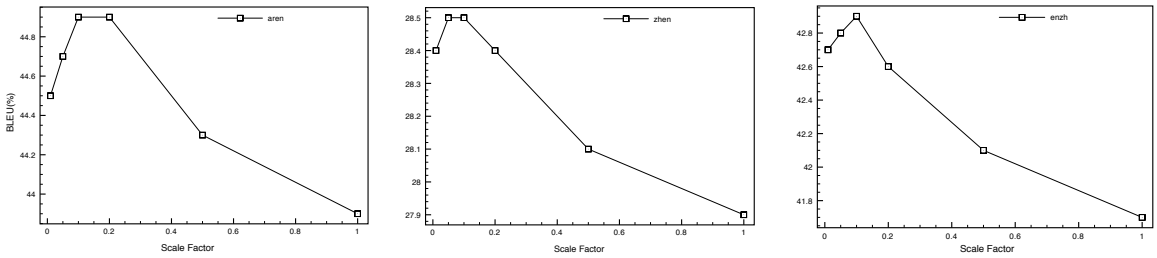

Figure 3: Lattice MBR with various scale factors α: aren/zhen/enzh.

tor on the performance of lattice MBR. The scale factor determines the flatness of the posterior distribution over translation hypotheses. A scale of 0.0 means a uniform distribution while 1.0 implies that there is no scaling. This is an important parameter that needs to be tuned on a development set. There has been prior work in MBR speech recognition and machine translation (Goel and Byrne, 2000; Ehling et al., 2007) which has shown the need for tuning this factor. Our MT system parameters are trained with Minimum Error Rate Training which assigns a very high posterior probability to the MAP translation. As a result, it is necessary to flatten the probability distribution so that MBR decoding can select hypotheses other than the MAP hypothesis.

Our Lattice MBR implementation is made possible due to the linear approximation of the BLEU score. This linearization technique has been applied elsewhere when working with BLEU: Smith and Eisner (2006) approximate the expectation of log BLEU score. In both cases, a linear metric makes it easier to compute the expectation. While we have applied lattice MBR decoding to the approximate BLEU score, we note that our procedure (Section 3) is applicable to other gain functions which can be decomposed as a sum of local gain functions. In particular, our framework might be useful with translation metrics such as TER (Snover et al., 2006) or METEOR (Lavie and Agarwal, 2007).

In contrast to a phrase-based SMT system, a syntax based SMT system (e.g. Zollmann and Venugopal (2006)) can generate a hypergraph that represents a generalized translation lattice with words and hidden tree structures. We believe that our lattice MBR framework can be extended to such hypergraphs with loss functions that take into account both BLEU scores as well as parse tree structures.

Lattice and Forest based search and training procedures are not yet common in statistical machine translation. However, they are promising because the search space of translations is much larger than the typical N-best list (Mi et al., 2008). We hope that our approach will provide some insight into the design of lattice-based search procedures along with the use of non-linear, global loss functions such as BLEU.

References

P. J. Bickel and K. A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected topics*. Holden-Day Inc., Oakland, CA, USA.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J . Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S.

Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

M. Dreyer, K. Hall, and S. Khudanpur. 2007. Comparing Reordering Constraints for SMT Using Efficient BLEU Oracle Computation. In *SSST, NAACL-HLT 2007*, pages 103–110, Rochester, NY, USA, April.

N. Ehling, R. Zens, and H. Ney. 2007. Minimum Bayes Risk Decoding for BLEU. In *ACL 2007*, pages 101–104, Prague, Czech Republic, June.

V. Goel and W. Byrne. 2000. Minimum Bayes-Risk Automatic Speech Recognition. *Computer Speech and Language*, 14(2):115–135.

V. Goel. 2001. *Minimum Bayes-Risk Automatic Speech Recognition*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD, USA.

J. Goodman. 1996. Parsing Algorithms and Metrics. In *ACL*, pages 177–183, Santa Cruz, CA, USA.

L. Karttunen, J-p. Chanod, G. Grefenstette, and A. Schiller. 1996. Regular Expressions for Language Engineering. *Natural Language Engineering*, 2:305–328.

P. Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP*, Barcelona, Spain.

S. Kumar and W. Byrne. 2002. Minimum Bayes-Risk word alignments of bilingual texts. In *EMNLP*, pages 140–147, Philadelphia, PA, USA.

S. Kumar and W. Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *HLT-NAACL*, pages 169–176, Boston, MA, USA.

A. Lavie and A. Agarwal. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *SMT Workshop, ACL*, pages 228–231, Prague, Czech Republic.

W. Macherey, F. Och, I. Thayer, and J. Uszkoreit. 2008. Lattice-based Minimum Error Rate Training for Statistical Machine Translation. In *EMNLP*, Honolulu, Hawaii, USA.

E. Matusov, R. Zens, and H. Ney. 2004. Symmetric Word Alignments for Statistical Machine Translation. In *COLING*, Geneva, Switzerland.

H. Mi, L. Huang, and Q. Liu. 2008. Forest-Based Translation. In *ACL*, Columbus, OH, USA.

M. Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(3).

M. Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350.

NIST. 2002-2008. The NIST Machine Translation Evaluations. http://www.nist.gov/speech/tests/mt/.

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

F. Och and H. Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417 – 449.

F. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *ACL*, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Division.

A. Sixtus and S. Ortmanns. 1999. High Quality Word Graphs Using Forward-Backward Pruning. In *ICASSP*, Phoenix, AZ, USA.

D. Smith and J. Eisner. 2006. Minimum Risk Annealing for Training Log-Linear Models. In *ACL*, Sydney, Australia.

D. Smith and N. Smith. 2007. Probabilistic models of nonprojective dependency trees. In *EMNLP-CoNLL*, Prague, Czech Republic.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *AMTA*, Boston, MA, USA.

I. Titov and J. Henderson. 2006. Loss Minimization in Parse Reranking. In *EMNLP*, Sydney, Australia.

N. Ueffing, F. Och, and H. Ney. 2002. Generation of Word Graphs in Statistical Machine Translation. In *EMNLP*, Philadelphia, PA, USA.

H. Zhang and D. Gildea. 2008. Efficient Multi-pass Decoding for Synchronous Context Free Grammars. In *ACL*, Columbus, OH, USA.

A. Zollmann and A. Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In *HLT-NAACL*, New York, NY, USA.