# Fast Consensus Decoding over Translation Forests

**John DeNero**
Computer Science Division
University of California, Berkeley
denero@cs.berkeley.edu

**David Chiang** and **Kevin Knight**
Information Sciences Institute
University of Southern California
{chiang, knight}@isi.edu

## Abstract

The minimum Bayes risk (MBR) decoding objective improves BLEU scores for machine translation output relative to the standard Viterbi objective of maximizing model score. However, MBR targeting BLEU is prohibitively slow to optimize over $k$-best lists for large $k$. In this paper, we introduce and analyze an alternative to MBR that is equally effective at improving performance, yet is asymptotically faster — running 80 times faster than MBR in experiments with 1000-best lists. Furthermore, our fast decoding procedure can select output sentences based on distributions over entire forests of translations, in addition to $k$-best lists. We evaluate our procedure on translation forests from two large-scale, state-of-the-art hierarchical machine translation systems. Our forest-based decoding objective consistently outperforms $k$-best list MBR, giving improvements of up to 1.0 BLEU.

## 1 Introduction

In statistical machine translation, output translations are evaluated by their similarity to human reference translations, where similarity is most often measured by BLEU (Papineni et al., 2002). A decoding objective specifies how to derive final translations from a system's underlying statistical model. The Bayes optimal decoding objective is to minimize risk based on the similarity measure used for evaluation. The corresponding minimum Bayes risk (MBR) procedure maximizes the expected similarity score of a system's translations relative to the model's distribution over possible translations (Kumar and Byrne, 2004). Unfortunately, with a non-linear similarity measure like BLEU, we must resort to approximating the expected loss using a $k$-best list, which accounts for only a tiny fraction of a model's full posterior distribution. In this paper, we introduce a variant of the MBR decoding procedure that applies efficiently to translation forests. Instead of maximizing expected similarity, we express similarity in terms of features of sentences, and choose translations that are similar to expected feature values.

Our exposition begins with algorithms over $k$-best lists. A naïve algorithm for finding MBR translations computes the similarity between every pair of $k$ sentences, entailing $O(k^2)$ comparisons. We show that if the similarity measure is linear in features of a sentence, then computing expected similarity for all $k$ sentences requires only $k$ similarity evaluations. Specific instances of this general algorithm have recently been proposed for two linear similarity measures (Tromble et al., 2008; Zhang and Gildea, 2008).

However, the sentence similarity measures we want to optimize in MT are *not* linear functions, and so this fast algorithm for MBR does not apply. For this reason, we propose a new objective that retains the benefits of MBR, but can be optimized efficiently, even for non-linear similarity measures. In experiments using BLEU over 1000-best lists, we found that our objective provided benefits very similar to MBR, only much faster.

This same decoding objective can also be computed efficiently from forest-based expectations. Translation forests compactly encode distributions over much larger sets of derivations and arise naturally in chart-based decoding for a wide variety of hierarchical translation systems (Chiang, 2007; Galley et al., 2006; Mi et al., 2008; Venugopal et al., 2007). The resulting forest-based decoding procedure compares favorably in both complexity and performance to the recently proposed lattice-based MBR (Tromble et al., 2008).

The contributions of this paper include a linear-time algorithm for MBR using linear similarities, a linear-time alternative to MBR using non-linear similarity measures, and a forest-based extension to this procedure for similarities based on $n$-gram counts. In experiments, we show that our fast procedure is on average 80 times faster than MBR using 1000-best lists. We also show that using forests outperforms using $k$-best lists consistently across language pairs. Finally, in the first published multi-system experiments on consensus decoding for translation, we demonstrate that benefits can differ substantially across systems. In all, we show improvements of up to 1.0 BLEU from consensus approaches for state-of-the-art large-scale hierarchical translation systems.

## 2 Consensus Decoding Algorithms

Let $e$ be a candidate translation for a sentence $f$, where $e$ may stand for a sentence or its derivation as appropriate. Modern statistical machine translation systems take as input some $f$ and score each derivation $e$ according to a linear model of features: $\sum_i \lambda_i \cdot \theta_i(f, e)$. The standard Viterbi decoding objective is to find $e^* = \arg\max_e \lambda \cdot \theta(f, e)$.

For MBR decoding, we instead leverage a similarity measure $S(e; e')$ to choose a translation using the model's probability distribution $\mathrm{P}(e|f)$, which has support over a set of possible translations $E$. The Viterbi derivation $e^*$ is the mode of this distribution. MBR is meant to choose a translation that will be similar, on expectation, to any possible reference translation. To this end, MBR chooses $\tilde{e}$ that maximizes expected similarity to the sentences in $E$ under $\mathrm{P}(e|f)$:[1]

$$\begin{aligned} \tilde{e} &= \arg\max_e \mathbb{E}_{\mathrm{P}(e'|f)}\left[S(e; e')\right] \\ &= \arg\max_e \sum_{e' \in E} \mathrm{P}(e'|f) \cdot S(e; e') \end{aligned}$$

MBR can also be interpreted as a consensus decoding procedure: it chooses a translation similar to other high-posterior translations. Minimizing risk has been shown to improve performance for MT (Kumar and Byrne, 2004), as well as other language processing tasks (Goodman, 1996; Goel and Byrne, 2000; Kumar and Byrne, 2002; Titov and Henderson, 2006; Smith and Smith, 2007).

The distribution $\mathrm{P}(e|f)$ can be induced from a translation system's features and weights by exponentiating with base $b$ to form a log-linear model:

$$P(e|f) = \frac{b^{\lambda \cdot \theta(f,e)}}{\sum_{e' \in E} b^{\lambda \cdot \theta(f,e')}}$$

We follow Ehling et al. (2007) in choosing $b$ using a held-out tuning set. For algorithms in this section, we assume that $E$ is a $k$-best list and $b$ has been chosen already, so $\mathrm{P}(e|f)$ is fully specified.

### 2.1 Minimum Bayes Risk over Sentence Pairs

Given any similarity measure $S$ and a $k$-best list $E$, the minimum Bayes risk translation can be found by computing the similarity between all pairs of sentences in $E$, as in Algorithm 1.

**Algorithm 1** MBR over Sentence Pairs

```
$A \leftarrow -\infty$
for $e \in E$ do
    $A_e \leftarrow 0$
    for $e' \in E$ do
        $A_e \leftarrow A_e + P(e'|f) \cdot S(e; e')$
    if $A_e > A$ then $A, \tilde{e} \leftarrow A_e, e$
return $\tilde{e}$
```

We can sometimes exit the inner *for* loop early, whenever $A_e$ can never become larger than $A$ (Ehling et al., 2007). Even with this shortcut, the running time of Algorithm 1 is $O(k^2 \cdot n)$, where $n$ is the maximum sentence length, assuming that $S(e; e')$ can be computed in $O(n)$ time.

### 2.2 Minimum Bayes Risk over Features

We now consider the case when $S(e; e')$ is a linear function of sentence features. Let $S(e; e')$ be a function of the form $\sum_j \omega_j(e) \cdot \phi_j(e')$, where $\phi_j(e')$ are real-valued features of $e'$, and $\omega_j(e)$ are sentence-specific weights on those features. Then, the MBR objective can be re-written as

$$\arg\max_{e \in E} \mathbb{E}_{\mathrm{P}(e'|f)}\left[S(e; e')\right]$$

$$\begin{aligned} &= \arg\max_e \sum_{e' \in E} P(e'|f) \cdot \sum_j \omega_j(e) \cdot \phi_j(e') \\ &= \arg\max_e \sum_j \omega_j(e) \left[\sum_{e' \in E} P(e'|f) \cdot \phi_j(e')\right] \\ &= \arg\max_e \sum_j \omega_j(e) \cdot \mathbb{E}_{\mathrm{P}(e'|f)}\left[\phi_j(e')\right]. \qquad (1) \end{aligned}$$

Equation 1 implies that we can find MBR translations by first computing all feature expectations, then applying $S$ only once for each $e$. Algorithm 2 proceduralizes this idea: lines 1-4 compute feature expectations, and lines 5-11 find the translation with highest $S$ relative to those expectations. The time complexity is $O(k \cdot n)$, assuming the number of non-zero features $\phi(e')$ and weights $\omega(e)$ grow linearly in sentence length $n$ and all features and weights can be computed in constant time.

[1] Typically, MBR is defined as $\arg\min_{e \in E} \mathbb{E}[L(e; e')]$ for some loss function $L$, for example $1 - \mathrm{BLEU}(e; e')$. These definitions are equivalent.

**Algorithm 2** MBR over Features

```
1:  $\bar{\phi} \leftarrow [0$ for $j \in J]$
2:  for $e' \in E$ do
3:      for $j \in J$ such that $\phi_j(e') \neq 0$ do
4:          $\bar{\phi}_j \leftarrow \bar{\phi}_j + P(e'|f) \cdot \phi_j(e')$
5:  $A \leftarrow -\infty$
6:  for $e \in E$ do
7:      $A_e \leftarrow 0$
8:      for $j \in J$ such that $\omega_j(e) \neq 0$ do
9:          $A_e \leftarrow A_e + \omega_j(e) \cdot \bar{\phi}_j$
10:     if $A_e > A$ then $A, \tilde{e} \leftarrow A_e, e$
11: return $\tilde{e}$
```

An example of a linear similarity measure is bag-of-words precision, which can be written as:

$$U(e;e') = \sum_{t \in T_1} \frac{\delta(e,t)}{|e|} \cdot \delta(e',t)$$

where $T_1$ is the set of unigrams in the language, and $\delta(e,t)$ is an indicator function that equals 1 if $t$ appears in $e$ and 0 otherwise. Figure 1 compares Algorithms 1 and 2 using $U(e;e')$. Other linear functions have been explored for MBR, including Taylor approximations to the logarithm of BLEU (Tromble et al., 2008) and counts of matching constituents (Zhang and Gildea, 2008), which are discussed further in Section 3.3.

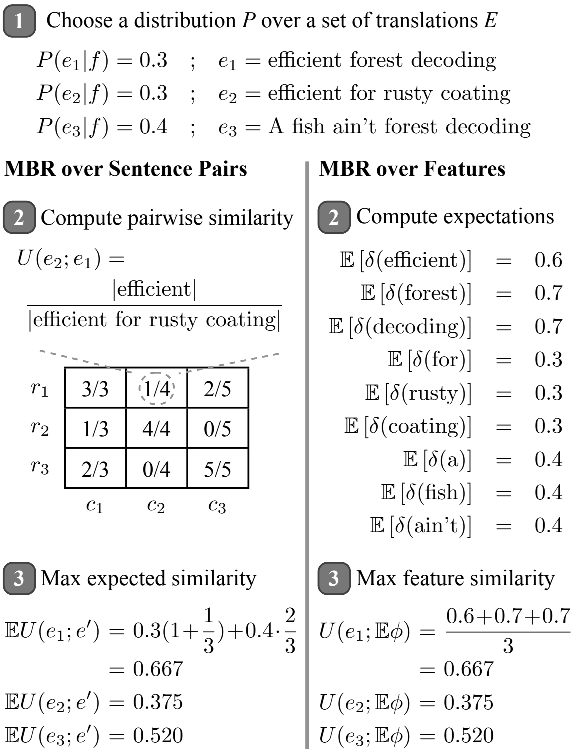
### 2.3 Fast Consensus Decoding using Non-Linear Similarity Measures

Most similarity measures of interest for machine translation are not linear, and so Algorithm 2 does not apply. Computing MBR even with simple non-linear measures such as BLEU, NIST or bag-of-words F1 seems to require $O(k^2)$ computation time. However, these measures are all functions of features of $e'$. That is, they can be expressed as $S(e;\phi(e'))$ for a feature mapping $\phi : E \rightarrow R^n$.

For example, we can express $\text{BLEU}(e;e') =$

$$\exp\left[\left(1 - \frac{|e'|}{|e|}\right)_{-} + \frac{1}{4}\sum_{n=1}^{4} \ln \frac{\sum_{t \in T_n} \min(c(e,t), c(e',t))}{\sum_{t \in T_n} c(e,t)}\right]$$

In this expression, $\text{BLEU}(e;e')$ references $e'$ only via its $n$-gram count features $c(e',t)$.[2]



1 Choose a distribution $P$ over a set of translations $E$

$P(e_1|f) = 0.3$ ; $e_1 =$ efficient forest decoding
$P(e_2|f) = 0.3$ ; $e_2 =$ efficient for rusty coating
$P(e_3|f) = 0.4$ ; $e_3 =$ A fish ain't forest decoding

**MBR over Sentence Pairs**

2 Compute pairwise similarity

$$U(e_2;e_1) = \frac{|\text{efficient}|}{|\text{efficient for rusty coating}|}$$

| | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|
| $r_1$ | 3/3 | 1/4 | 2/5 |
| $r_2$ | 1/3 | 4/4 | 0/5 |
| $r_3$ | 2/3 | 0/4 | 5/5 |

3 Max expected similarity

$$\mathbb{E}U(e_1;e') = 0.3(1+\frac{1}{3})+0.4\cdot\frac{2}{3} = 0.667$$
$$\mathbb{E}U(e_2;e') = 0.375$$
$$\mathbb{E}U(e_3;e') = 0.520$$

**MBR over Features**

2 Compute expectations

$$\mathbb{E}[\delta(\text{efficient})] = 0.6$$
$$\mathbb{E}[\delta(\text{forest})] = 0.7$$
$$\mathbb{E}[\delta(\text{decoding})] = 0.7$$
$$\mathbb{E}[\delta(\text{for})] = 0.3$$
$$\mathbb{E}[\delta(\text{rusty})] = 0.3$$
$$\mathbb{E}[\delta(\text{coating})] = 0.3$$
$$\mathbb{E}[\delta(\text{a})] = 0.4$$
$$\mathbb{E}[\delta(\text{fish})] = 0.4$$
$$\mathbb{E}[\delta(\text{ain't})] = 0.4$$

3 Max feature similarity

$$U(e_1;\mathbb{E}\phi) = \frac{0.6+0.7+0.7}{3} = 0.667$$
$$U(e_2;\mathbb{E}\phi) = 0.375$$
$$U(e_3;\mathbb{E}\phi) = 0.520$$

Figure 1: For the linear similarity measure $U(e;e')$, which computes unigram precision, the MBR translation can be found by iterating either over sentence pairs (Algorithm 1) or over features (Algorithm 2). These two algorithms take the same input (step 1), but diverge in their consensus computations (steps 2 & 3). However, they produce identical results for $U$ and any other linear similarity measure.

Following the structure of Equation 1, we can choose a translation $e$ based on the feature expectations of $e'$. In particular, we can choose

$$\tilde{e} = \arg\max_{e \in E} S(e; \mathbb{E}_{P(e'|f)}[\phi(e')]). \quad (2)$$

This objective differs from MBR, but has a similar consensus-building structure. We have simply moved the expectation inside the similarity function, just as we did in Equation 1. This new objective can be optimized by Algorithm 3, a procedure that runs in $O(k \cdot n)$ time if the count of non-zero features in $e'$ and the computation time of $S(e;\phi(e'))$ are both linear in sentence length $n$.

This fast consensus decoding procedure shares the same structure as linear MBR: first we compute feature expectations, then we choose the sentence that is most similar to those expectations. In fact, Algorithm 2 is a special case of Algorithm 3. Lines 7-9 of the former and line 7 of the latter are equivalent for linear $S(e;e')$. Thus, for any linear similarity measure, Algorithm 3 is an algorithm for minimum Bayes risk decoding.

[2] The length penalty $\left(1 - \frac{|e'|}{|e|}\right)_{-}$ is also a function of $n$-gram counts: $|e'| = \sum_{t \in T_1} c(e',t)$. The negative part operator $(\cdot)_{-}$ is equivalent to $\min(\cdot, 0)$.

**Algorithm 3** Fast Consensus Decoding

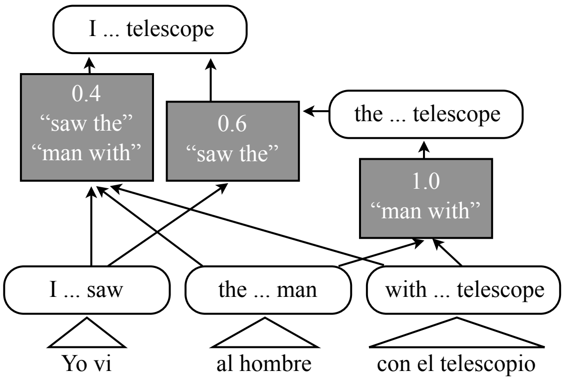
```
$\bar{\phi} \leftarrow [0$ **for** $j \in J]$
**for** $e' \in E$ **do**
    **for** $j \in J$ **such that** $\phi_j(e') \neq 0$ **do**
        $\bar{\phi}_j \leftarrow \bar{\phi}_j + P(e'|f) \cdot \phi_j(e')$
$A \leftarrow -\infty$
**for** $e \in E$ **do**
    $A_e \leftarrow S(e; \bar{\phi})$
    **if** $A_e > A$ **then** $A, \tilde{e} \leftarrow A_e, e$
**return** $\tilde{e}$
```

As described, Algorithm 3 can use any similarity measure that is defined in terms of real-valued features of $e'$. There are some nuances of this procedure, however. First, the precise form of $S(e; \phi(e'))$ will affect the output, but $S(e; \mathbb{E}[\phi(e')])$ is often an input point for which a sentence similarity measure $S$ was not originally defined. For example, our definition of BLEU above will have integer valued $\phi(e')$ for any real sentence $e'$, but $\mathbb{E}[\phi(e')]$ will not be integer valued. As a result, we are extending the domain of BLEU beyond its original intent. One could imagine different feature-based expressions that also produce BLEU scores for real sentences, but produce different values for fractional features. Some care must be taken to define $S(e; \phi(e'))$ to extend naturally from integer-valued to real-valued features.

Second, while any similarity measure can in principle be expressed as $S(e; \phi(e'))$ for a sufficiently rich feature space, fast consensus decoding will not apply effectively to all functions. For instance, we cannot naturally use functions that include alignments or matchings between $e$ and $e'$, such as METEOR (Agarwal and Lavie, 2007) and TER (Snover et al., 2006). Though these functions can in principle be expressed in terms of features of $e'$ (for instance with indicator features for whole sentences), fast consensus decoding will only be effective if different sentences share many features, so that the feature expectations effectively capture trends in the underlying distribution.

## 3 Computing Feature Expectations

We now turn our focus to efficiently computing feature expectations, in service of our fast consensus decoding procedure. Computing feature expectations from $k$-best lists is trivial, but $k$-best lists capture very little of the underlying model's posterior distribution. In place of $k$-best lists, compact encodings of translation distributions have proven effective for MBR (Zhang and Gildea, 2008; Tromble et al., 2008). In this section, we consider BLEU in particular, for which the relevant features $\phi(e)$ are $n$-gram counts up to length $n = 4$. We show how to compute expectations of these counts efficiently from translation forests.



$$\begin{aligned} \mathbb{E}\left[c(e, \text{“man with”})\right] &= \sum_h P(h|f) \cdot c(h, \text{“man with”}) \\ &= 0.4 \cdot 1 + (0.6 \cdot 1.0) \cdot 1 \end{aligned}$$

Figure 2: This translation forest for a Spanish sentence encodes two English parse trees. Hyper-edges (boxes) are annotated with normalized transition probabilities, as well as the bigrams produced by each rule application. The expected count of the bigram "man with" is the sum of posterior probabilities of the two hyper-edges that produce it. In this example, we normalized inside scores at all nodes to 1 for clarity.

### 3.1 Translation Forests

Translation forests compactly encode an exponential number of output translations for an input sentence, along with their model scores. Forests arise naturally in chart-based decoding procedures for many hierarchical translation systems (Chiang, 2007). Exploiting forests has proven a fruitful avenue of research in both parsing (Huang, 2008) and machine translation (Mi et al., 2008).

Formally, translation forests are weighted acyclic hyper-graphs. The nodes are states in the decoding process that include the span $(i, j)$ of the sentence to be translated, the grammar symbol $s$ over that span, and the left and right context words of the translation relevant for computing $n$-gram language model scores.[3] Each hyper-edge $h$ represents the application of a synchronous rule $r$ that combines nodes corresponding to non-terminals in

[3] Decoder states can include additional information as well, such as local configurations for dependency language model scoring.

$r$ into a node spanning the union of the child spans and perhaps some additional portion of the input sentence covered directly by $r$'s lexical items. The weight of $h$ is the incremental score contributed to all translations containing the rule application, including translation model features on $r$ and language model features that depend on both $r$ and the English contexts of the child nodes. Figure 2 depicts a forest.

Each $n$-gram that appears in a translation $e$ is associated with some $h$ in its derivation: the $h$ corresponding to the rule that produces the $n$-gram. Unigrams are produced by lexical rules, while higher-order $n$-grams can be produced either directly by lexical rules, or by combining constituents. The $n$-gram language model score of $e$ similarly decomposes over the $h$ in $e$ that produce $n$-grams.

### 3.2 Computing Expected N-Gram Counts

We can compute expected $n$-gram counts efficiently from a translation forest by appealing to the linearity of expectations. Let $\phi(e)$ be a vector of $n$-gram counts for a sentence $e$. Then, $\phi(e)$ is the sum of hyper-edge-specific $n$-gram count vectors $\phi(h)$ for all $h$ in $e$. Therefore, $\mathbb{E}[\phi(e)] = \sum_{h \in e} \mathbb{E}[\phi(h)]$.

To compute $n$-gram expectations for a hyperedge, we first compute the posterior probability of each $h$, conditioned on the input sentence $f$:

$$\mathrm{P}(h|f) = \left( \sum_{e:h \in e} b^{\lambda \cdot \theta(f,e)} \right) \left( \sum_{e} b^{\lambda \cdot \theta(f,e)} \right)^{-1},$$

where $e$ iterates over translations in the forest. We compute the numerator using the inside-outside algorithm, while the denominator is the inside score of the root node. Note that many possible derivations of $f$ are pruned from the forest during decoding, and so this posterior is approximate.

The expected $n$-gram count vector for a hyperedge is $\mathbb{E}[\phi(h)] = \mathrm{P}(h|f) \cdot \phi(h)$. Hence, after computing $P(h|f)$ for every $h$, we need only sum $\mathrm{P}(h|f) \cdot \phi(h)$ for all $h$ to compute $\mathbb{E}[\phi(e)]$. This entire procedure is a linear-time computation in the number of hyper-edges in the forest.

To complete forest-based fast consensus decoding, we then extract a $k$-best list of unique translations from the forest (Huang et al., 2006) and continue Algorithm 3 from line 5, which chooses the $\tilde{e}$ from the $k$-best list that maximizes $\mathrm{BLEU}(e; \mathbb{E}[\phi(e')])$.

### 3.3 Comparison to Related Work

Zhang and Gildea (2008) embed a consensus decoding procedure into a larger multi-pass decoding framework. They focus on inversion transduction grammars, but their ideas apply to richer models as well. They propose an MBR decoding objective of maximizing the expected number of matching constituent counts relative to the model's distribution. The corresponding constituent-matching similarity measure can be expressed as a linear function of features of $e'$, which are indicators of constituents. Expectations of constituent indicator features are the same as posterior constituent probabilities, which can be computed from a translation forest using the inside-outside algorithm. This forest-based MBR approach improved translation output relative to Viterbi translations.

Tromble et al. (2008) describe a similar approach using MBR with a linear similarity measure. They derive a first-order Taylor approximation to the logarithm of a slightly modified definition of corpus BLEU[4], which is linear in $n$-gram indicator features $\delta(e', t)$ of $e'$. These features are weighted by $n$-gram counts $c(e, t)$ and constants $\theta$ that are estimated from held-out data. The linear similarity measure takes the following form, where $T_n$ is the set of $n$-grams:

$$G(e; e') = \theta_0 |e| + \sum_{n=1}^{4} \sum_{t \in T_n} \theta_t \cdot c(e, t) \cdot \delta(e', t).$$

Using $G$, Tromble et al. (2008) extend MBR to word lattices, which improves performance over $k$-best list MBR.

Our approach differs from Tromble et al. (2008) primarily in that we propose decoding with an alternative to MBR using BLEU, while they propose decoding with MBR using a linear alternative to BLEU. The specifics of our approaches also differ in important ways.

First, word lattices are a subclass of forests that have only one source node for each edge (i.e., a graph, rather than a hyper-graph). While forests are more general, the techniques for computing posterior edge probabilities in lattices and forests are similar. One practical difference is that the forests needed for fast consensus decoding are

[4] The log-BLEU function must be modified slightly to yield a linear Taylor approximation: Tromble et al. (2008) replace the clipped $n$-gram count with the product of an $n$-gram count and an $n$-gram indicator function.

generated already by the decoder of a syntactic translation system.

Second, rather than use BLEU as a sentence-level similarity measure directly, Tromble et al. (2008) approximate corpus BLEU with $G$ above. The parameters $\theta$ of the approximation must be estimated on a held-out data set, while our approach requires no such estimation step.

Third, our approach is also simpler computationally. The features required to compute $G$ are indicators $\delta(e', t)$; the features relevant to us are counts $c(e', t)$. Tromble et al. (2008) compute expected feature values by intersecting the translation lattice with a lattices for each $n$-gram $t$. By contrast, expectations of $c(e', t)$ can all be computed with a single pass over the forest. This contrast implies a complexity difference. Let $H$ be the number of hyper-edges in the forest or lattice, and $T$ the number of $n$-grams that can potentially appear in a translation. Computing indicator expectations seems to require $O(H \cdot T)$ time because of automata intersections. Computing count expectations requires $O(H)$ time, because only a constant number of $n$-grams can be produced by each hyper-edge.

Our approaches also differ in the space of translations from which $\tilde{e}$ is chosen. A linear similarity measure like $G$ allows for efficient search over the lattice or forest, whereas fast consensus decoding restricts this search to a $k$-best list. However, Tromble et al. (2008) showed that most of the improvement from lattice-based consensus decoding comes from lattice-based expectations, not search: searching over lattices instead of $k$-best lists did not change results for two language pairs, and improved a third language pair by 0.3 BLEU. Thus, we do not consider our use of $k$-best lists to be a substantial liability of our approach.

Fast consensus decoding is also similar in character to the concurrently developed variational decoding approach of Li et al. (2009). Using BLEU, both approaches choose outputs that match expected $n$-gram counts from forests, though differ in the details. It is possible to define a similarity measure under which the two approaches are equivalent.[5]

## 4 Experimental Results

We evaluate these consensus decoding techniques on two different full-scale state-of-the-art hierarchical machine translation systems. Both systems were trained for 2008 GALE evaluations, in which they outperformed a phrase-based system trained on identical data.

### 4.1 Hiero: a Hierarchical MT Pipeline

Hiero is a hierarchical system that expresses its translation model as a synchronous context-free grammar (Chiang, 2007). No explicit syntactic information appears in the core model. A phrase discovery procedure over word-aligned sentence pairs provides rule frequency counts, which are normalized to estimate features on rules.

The grammar rules of Hiero all share a single non-terminal symbol $X$, and have at most two non-terminals and six total items (non-terminals and lexical items), for example:

$$\text{my } X_2 \text{ 's } X_1 \rightarrow X_1 \text{ de mi } X_2$$

We extracted the grammar from training data using standard parameters. Rules were allowed to span at most 15 words in the training data.

The log-linear model weights were trained using MIRA, a margin-based optimization procedure that accommodates many features (Crammer and Singer, 2003; Chiang et al., 2008). In addition to standard rule frequency features, we included the distortion and syntactic features described in Chiang et al. (2008).

### 4.2 SBMT: a Syntax-Based MT Pipeline

SBMT is a string-to-tree translation system with rich target-side syntactic information encoded in the translation model. The synchronous grammar rules are extracted from word aligned sentence pairs where the target sentence is annotated with a syntactic parse (Galley et al., 2004). Rules map source-side strings to target-side parse tree fragments, and non-terminal symbols correspond to target-side grammatical categories:

$$\text{(NP (NP (PRP\$ my) NN}_2 \text{ (POS 's)) NNS}_1) \rightarrow \\ \text{NNS}_1 \text{ de mi NN}_2$$

We extracted the grammar via an array of criteria (Galley et al., 2006; DeNeefe et al., 2007; Marcu et al., 2006). The model was trained using minimum error rate training for Arabic (Och, 2003) and MIRA for Chinese (Chiang et al., 2008).

[5] For example, decoding under a variational approximation to the model's posterior that decomposes over bigram probabilities is equivalent to fast consensus decoding with the similarity measure $B(e; e') = \prod_{t \in T_2} \left[ \frac{c(e', t)}{c(e', h(t))} \right]^{c(e,t)}$, where $h(t)$ is the unigram prefix of bigram $t$.

| Arabic-English | | |
|---|---|---|
| **Objective** | **Hiero** | **SBMT** |
| Min. Bayes Risk (Alg 1) | 2h 47m | 12h 42m |
| Fast Consensus (Alg 3) | 5m 49s | 5m 22s |
| *Speed Ratio* | *29* | *142* |

| Chinese-English | | |
|---|---|---|
| **Objective** | **Hiero** | **SBMT** |
| Min. Bayes Risk (Alg 1) | 10h 24m | 3h 52m |
| Fast Consensus (Alg 3) | 4m 52s | 6m 32s |
| *Speed Ratio* | *128* | *36* |

Table 1: Fast consensus decoding is orders of magnitude faster than MBR when using BLEU as a similarity measure. Times only include reranking, not $k$-best list extraction.

### 4.3 Data Conditions

We evaluated on both Chinese-English and Arabic-English translation tasks. Both Arabic-English systems were trained on 220 million words of word-aligned parallel text. For the Chinese-English experiments, we used 260 million words of word-aligned parallel text; the hierarchical system used all of this data, and the syntax-based system used a 65-million word subset. All four systems used two language models: one trained from the combined English sides of both parallel texts, and another, larger, language model trained on 2 billion words of English text (1 billion for Chinese-English SBMT).

All systems were tuned on held-out data (1994 sentences for Arabic-English, 2010 sentences for Chinese-English) and tested on another dataset (2118 sentences for Arabic-English, 1994 sentences for Chinese-English). These datasets were drawn from the NIST 2004 and 2005 evaluation data, plus some additional data from the GALE program. There was no overlap at the segment or document level between the tuning and test sets.

We tuned $b$, the base of the log-linear model, to optimize consensus decoding performance. Interestingly, we found that tuning $b$ on the same dataset used for tuning $\lambda$ was as effective as tuning $b$ on an additional held-out dataset.

### 4.4 Results over $K$-Best Lists

Taking expectations over 1000-best lists[6] and using BLEU[7] as a similarity measure, both MBR and our variant provided consistent small gains of 0.0–0.2 BLEU. Algorithms 1 and 3 gave the same small BLEU improvements in each data condition up to three significant figures.

The two algorithms differed greatly in speed, as shown in Table 1. For Algorithm 1, we terminated the computation of $\mathbb{E}[BLEU(e;e')]$ for each $e$ whenever $e$ could not become the maximal hypothesis. MBR speed depended on how often this shortcut applied, which varied by language and system. Despite this optimization, our new Algorithm 3 was an average of 80 times faster across systems and language pairs.

| Arabic-English | | | |
|---|---|---|---|
| **Expectations** | **Similarity** | **Hiero** | **SBMT** |
| Baseline | - | 52.0 | 53.9 |
| $10^4$-best | BLEU | 52.2 | 53.9 |
| Forest | BLEU | **53.0** | 54.0 |
| Forest | Linear $G$ | 52.3 | 54.0 |

| Chinese-English | | | |
|---|---|---|---|
| **Expectations** | **Similarity** | **Hiero** | **SBMT** |
| Baseline | - | 37.8 | 40.6 |
| $10^4$-best | BLEU | 38.0 | 40.7 |
| Forest | BLEU | **38.2** | 40.8 |
| Forest | Linear $G$ | 38.1 | 40.8 |

Table 2: Translation performance improves when computing expected sentences from translation forests rather than $10^4$-best lists, which in turn improve over Viterbi translations. We also contrasted forest-based consensus decoding with BLEU and its linear approximation, $G$. Both similarity measures are effective, but BLEU outperforms $G$.

### 4.5 Results for Forest-Based Decoding

Table 2 contrasts Algorithm 3 over $10^4$-best lists and forests. Computing $\mathbb{E}[\phi(e')]$ from a translation forest rather than a $10^4$-best list improved Hiero by an additional 0.8 BLEU (1.0 over the baseline). Forest-based expectations always outperformed $k$-best lists, but curiously the magnitude of benefit was not consistent across systems. We believe the difference is in part due to more aggressive forest pruning within the SBMT decoder.

For forest-based decoding, we compared two similarity measures: BLEU and its linear Taylor approximation $G$ from section 3.3.[8] Table 2 shows

[6] We ensured that $k$-best lists contained no duplicates.

[7] To prevent zero similarity scores, we also used a standard smoothed version of BLEU that added 1 to the numerator and denominator of all $n$-gram precisions. Performance results were identical to standard BLEU.

[8] We did not estimate the $\theta$ parameters of $G$ ourselves; instead we used the parameters listed in Tromble et al. (2008), which were also estimated for GALE data. We also approximated $\mathbb{E}[\delta(e',t)]$ with a clipped expected count

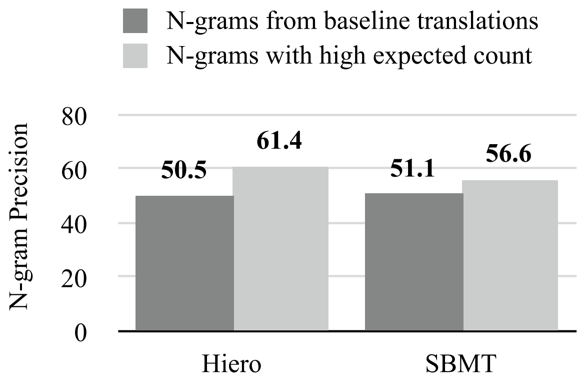


Figure 3: $N$-grams with high expected count are more likely to appear in the reference translation that $n$-grams in the translation model's Viterbi translation, $e^*$. Above, we compare the precision, relative to reference translations, of sets of $n$-grams chosen in two ways. The left bar is the precision of the $n$-grams in $e^*$. The right bar is the precision of $n$-grams with $\mathbb{E}[c(e,t)] > \rho$. To justify this comparison, we chose $\rho$ so that both methods of choosing $n$-grams gave the same $n$-gram recall: the fraction of $n$-grams in reference translations that also appeared in $e^*$ or had $\mathbb{E}[c(e,t)] > \rho$.

that both similarities were effective, but BLEU outperformed its linear approximation.

### 4.6 Analysis

Forest-based consensus decoding leverages information about the correct translation from the entire forest. In particular, consensus decoding with BLEU chooses translations using $n$-gram count expectations $\mathbb{E}[c(e,t)]$. Improvements in translation quality should therefore be directly attributable to information in these expected counts.

We endeavored to test the hypothesis that expected $n$-gram counts under the forest distribution carry more predictive information than the baseline Viterbi derivation $e^*$, which is the mode of the distribution. To this end, we first tested the predictive accuracy of the $n$-grams proposed by $e^*$: the fraction of the $n$-grams in $e^*$ that appear in a reference translation. We compared this $n$-gram precision to a similar measure of predictive accuracy for expected $n$-gram counts: the fraction of the $n$-grams $t$ with $\mathbb{E}[c(e,t)] \geq \rho$ that appear in a reference. To make these two precisions comparable, we chose $\rho$ such that the recall of reference $n$-grams was equal. Figure 3 shows that computing $n$-gram expectations—which sum over translations—improves the model's ability to predict which $n$-grams will appear in the reference. We attribute gains from fast consensus decoding to this increased predictive accuracy.

**Reference translation:**
Mubarak said that he received a telephone call from Sharon in which he said he was "ready (to resume negotiations) but the Palestinians are hesitant."

**Baseline translation:**
Mubarak said he had received a telephone call from Sharon told him he was ready to resume talks with the Palestinians.

**Fast forest-based consensus translation:**
Mubarak said that he had received a telephone call from Sharon told him that he "was ready to resume the negotiations) , *but the Palestinians are hesitant*."

Figure 4: Three translations of an example Arabic sentence: its human-generated reference, the translation with the highest model score under Hiero (Viterbi), and the translation chosen by forest-based consensus decoding. The consensus translation reconstructs content lost in the Viterbi translation.

Examining the translations chosen by fast consensus decoding, we found that gains in BLEU often arose from improved lexical choice. However, in our hierarchical systems, consensus decoding did occasionally trigger large reordering. We also found examples where the translation quality improved by recovering content that was missing from the baseline translation, as in Figure 4.

## 5 Conclusion

We have demonstrated substantial speed increases in $k$-best consensus decoding through a new procedure inspired by MBR under linear similarity measures. To further improve this approach, we computed expected $n$-gram counts from translation forests instead of $k$-best lists. Fast consensus decoding using forest-based $n$-gram expectations and BLEU as a similarity measure yielded consistent improvements over MBR with $k$-best lists, yet required only simple computations that scale linearly with the size of the translation forest.

The space of similarity measures is large and relatively unexplored, and the feature expectations that can be computed from forests extend beyond $n$-gram counts. Therefore, future work may show additional benefits from fast consensus decoding.

## Acknowledgements

This work was supported under DARPA GALE, Contract No. HR0011-06-C-0022.

$\min(1, \mathbb{E}[c(e',t)])$. Assuming an $n$-gram appears at most once per sentence, these expressions are equivalent, and this assumption holds for most $n$-grams.

## References

Abhaya Agarwal and Alon Lavie. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Workshop on Statistical Machine Translation for the Association of Computational Linguistics*.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Steve DeNeefe, Kevin Knight, Wei Wang, and Daniel Marcu. 2007. What can syntax-based MT learn from phrase-based MT? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and CoNLL*.

Nicola Ehling, Richard Zens, and Hermann Ney. 2007. Minimum Bayes risk decoding for BLEU. In *Proceedings of the Association for Computational Linguistics: Short Paper Track*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of HLT: the North American Chapter of the Association for Computational Linguistics*.

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 *Proceedings of the Association for Computational Linguistics*.

Vaibhava Goel and William Byrne. 2000. Minimum Bayes-risk automatic speech recognition. In *Computer, Speech and Language*.

Joshua Goodman. 1996. Parsing algorithms and metrics. In *Proceedings of the Association for Computational Linguistics*.

Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the Association for Machine Translation in the Americas*.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the Association for Computational Linguistics*.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proceedings of the Association for Computational Linguistics and IJCNLP*.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of the Association for Computational Linguistics*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*.

David Smith and Noah Smith. 2007. Probabilistic models of nonprojective dependency trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and CoNLL*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Ivan Titov and James Henderson. 2006. Loss minimization in parse reranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Roy Tromble, Shankar Kumar, Franz Josef Och, and Wolfgang Macherey. 2008. Lattice minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. 2007. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Proceedings of HLT: the North American Association for Computational Linguistics Conference*.

Hao Zhang and Daniel Gildea. 2008. Efficient multi-pass decoding for synchronous context free grammars. In *Proceedings of the Association for Computational Linguistics*.