Improved Reordering for Phrase-Based Translation using Sparse Features

Colin Cherry
National Research Council Canada
Colin.Cherry@nrc-cnrc.gc.ca

Abstract

There have been many recent investigations into methods to tune SMT systems using large numbers of sparse features. However, there have not been nearly so many examples of helpful sparse features, especially for phrase-based systems. We use sparse features to address reordering, which is often considered a weak point of phrase-based translation. Using a hierarchical reordering model as our baseline, we show that simple features coupling phrase orientation to frequent words or word-clusters can improve translation quality, with boosts of up to 1.2 BLEU points in Chinese-English and 1.8 in Arabic-English. We compare this solution to a more traditional maximum entropy approach, where a probability model with similar features is trained on word-aligned bitext. We show that sparse decoder features outperform maximum entropy handily, indicating that there are major advantages to optimizing reordering features directly for BLEU with the decoder in the loop.

1 Introduction

With the growing adoption of tuning algorithms that can handle thousands of features (Chiang et al., 2008; Hopkins and May, 2011), SMT system designers now face a choice when incorporating new ideas into their translation models. Maximum likelihood models can be estimated from large word-aligned bitexts, creating a small number of highly informative decoder features; or the same ideas can be incorporated into the decoder's linear model directly. There are trade-offs to each approach. Maximum likelihood models can be estimated from millions of sentences of bitext, but optimize a mismatched objective, predicting events observed in word aligned bitext instead of optimizing translation quality. Sparse decoder features have the opposite problem; with the decoder in the loop, we can only tune on small development sets,[1] but a translation error metric directly informs training.

We investigate this trade-off in the context of reordering models for phrase-based decoding. Starting with the intuition that most lexicalized reordering models do not smooth their orientation distributions intelligently for low-frequency phrase-pairs, we design features that track the first and last words (or clusters) of the phrases in a pair. These features are incorporated into a maximum entropy reordering model, as well as sparse decoder features, to see which approach best complements the now-standard relative-frequency lexicalized reordering model.

We also view our work as an example of strong sparse features for phrase-based translation. Features from hierarchical and syntax-based translation (Chiang et al., 2009) do not easily transfer to the phrase-based paradigm, and most work that has looked at large feature counts in the context of phrase-based translation has focused on the learning method, and not the features themselves (Hopkins and May, 2011; Cherry and Foster, 2012; Gimpel and Smith, 2012). We show that by targeting reordering, large gains can be made with relatively simple features.

2 Background

Phrase-based machine translation constructs its target sentence from left-to-right, with each translation operation selecting a source phrase and appending its translation to the growing target sentence, until

[1] Some systems tune for BLEU on much larger sets (Simianer et al., 2012; He and Deng, 2012), but these require exceptional commitments of resources and time.

Proceedings of NAACL-HLT 2013, pages 22–31,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

all source words have been covered exactly once. The first phrase-based translation systems applied only a distortion penalty to model reordering (Koehn et al., 2003; Och and Ney, 2004). Any deviation from monotone translation is penalized, with a single linear weight determining how quickly the penalty grows.

2.1 Lexicalized Reordering

Implemented in a number of phrase-based decoders, the lexicalized reordering model (RM) uses word-aligned data to determine how each phrase-pair tends to be reordered during translation (Tillmann, 2004; Koehn et al., 2005; Koehn et al., 2007).

The core idea in this RM is to divide reordering events into three orientations that can be easily determined both during decoding and from word-aligned data. The orientations can be described in terms of the previously translated source phrase (*prev*) and the next source phrase to be translated (*next*):

- Monotone (M): *next* immediately follows *prev*.
- Swap (S): *prev* immediately follows *next*.
- Discontinuous (D): *next* and *prev* are not adjacent in the source.

Note that *prev* and *next* can be defined for constructing a translation from left-to-right or from right-to-left. Most decoders incorporate RMs for both directions; our discussion will generally only cover left-to-right, with the right-to-left case being implicit and symmetrical.

As the decoder extends its hypothesis by translating a source phrase, we can assess the implied orientations to determine if the resulting reordering makes sense. This is done using the probability of an orientation given the phrase pair $pp = [src, tgt]$ extending the hypothesis:[2]

$$P(o|pp) \approx \frac{cnt(o, pp)}{\sum_o cnt(o, pp)} \qquad (1)$$

where $o \in \{M, S, D\}$, cnt uses simple heuristics on word-alignments to count phrase pairs and their orientations, and the $\approx$ symbol allows for smoothing. The log of this probability is easily folded into the linear models that guide modern decoders. Better performance is achieved by giving each orientation its own log-linear weight (Koehn et al., 2005).

2.2 Hierarchical Reordering

Introduced by Galley and Manning (2008), the hierarchical reordering model (HRM) also tracks statistics over orientations, but attempts to increase the consistency of orientation assignments. To do so, they remove the emphasis on the previously translated phrase ($prev$), and instead determine orientation using a compact representation of the full translation history, as represented by a shift-reduce stack. Each source span is shifted onto the stack as it is translated; if the new top is adjacent to the span below it, then a reduction merges the two.

Orientations are determined in terms of the top of this stack,[3] rather than the previously translated phrase *prev*. The resulting orientations are more consistent across different phrasal decompositions of the same translation, and more consistent with the statistics extracted from word aligned data. This results in a general improvement in performance. We assume the HRM as our baseline reordering model.

It is important to note that although our maximum entropy and sparse reordering solutions build on the HRM, the features in this paper can still be applied without a shift-reduce stack, by using the previously translated phrase where we use the top of the stack.

2.3 Maximum Entropy Reordering

One frequent observation regarding both the RM and the HRM is that the statistics used to grade orientations are very sparse. Each orientation prediction $P(o|pp)$ is conditioned on an entire phrase pair. Koehn et al. (2005) experiment with alternatives, such as conditioning on only the source or the target, but using the entire pair generally performs best. The vast majority of phrase pairs found in bitext with standard extraction heuristics are singletons (more than 92% in our Arabic-English bitext), and the corresponding $P(o|pp)$ estimates are based on a single observation. Because of these heavy tails, there have been several attempts to use maximum entropy to create more flexible distributions.

One straight-forward way to do so is to continue predicting orientations on phrases, but to use maxi-

[2] *pp* corresponds to the phrase pair translating *next* for the left-to-right model, and *prev* for right-to-left.

[3] In the case of the right-to-left model, an approximation of the top of the stack is used instead.

mum entropy to consider features of the phrase pair. This is the approach taken by Xiong et al. (2006); their maximum entropy model chooses between M and S orientations, which are the only two options available in their chart-based ITG decoder. Nguyen et al. (2009) build a similar model for a phrase-based HRM, using syntactic heads and constituent labels to create a rich feature set. They show gains over an HRM baseline, albeit on a small training set.

A related approach is to build a reordering model over words, which is evaluated at phrase boundaries at decoding time. Zens and Ney (2006) propose one such model, with jumps between words binned very coarsely according to their direction and distance, testing models that differentiate only left jumps from right, as well as the cross-product of {left, right} × {adjacent, discontinuous}. Their features consider word identity and automatically-induced clusters. Green et al. (2010) present a similar approach, with finer-grained distance bins, using word-identity and part-of-speech for features. Yahyaei and Monz (2010) also predict distance bins, but use much more context, opting to look at both sides of a reordering jump; they also experiment with hard constraints based on their model.

Tracking word-level reordering simplifies the extraction of complex models from word alignments; however, it is not clear if it is possible to enhance a word reordering model with the stack-based histories used by HRMs. In this work, we construct a phrase orientation maximum entropy model.

3 Methods

Our primary contribution is a comparison between the standard HRM and two feature-based alternatives. Since a major motivating concern is smoothing, we begin with a detailed description of our HRM baseline, followed by our maximum entropy HRM and our novel sparse reordering features.

3.1 Relative Frequency Model

The standard HRM uses relative frequencies to build smoothed maximum likelihood estimates of orientation probabilities. Orientation counts for phrase pairs are collected from bitext, using the method described by Galley and Manning (2008). The probability model $P(o|pp = [src, tgt])$ is estimated using recursive MAP smoothing:

$$P(o|pp) = \frac{cnt(o,pp) + \alpha_s P_s(o|src) + \alpha_t P_t(o|tgt)}{\sum_o cnt(o,pp) + \alpha_s + \alpha_t}$$

$$P_s(o|src) = \frac{\sum_{tgt} cnt(o,src,tgt) + \alpha_g P_g(o)}{\sum_{o,tgt} cnt(o,src,tgt) + \alpha_g}$$

$$P_t(o|tgt) = \frac{\sum_{src} cnt(o,src,tgt) + \alpha_g P_g(o)}{\sum_{o,src} cnt(o,src,tgt) + \alpha_g}$$

$$P_g(o) = \frac{\sum_{pp} cnt(o,pp) + \alpha_u/3}{\sum_{o,pp} cnt(o,pp) + \alpha_u} \qquad (2)$$

where the various α parameters can be tuned empirically. In practice, the model is not particularly sensitive to these parameters.[4]

3.2 Maximum Entropy Model

Next, we describe our implementation of a maximum entropy HRM. Our goal with this system is to benefit from modeling features of a phrase pair, while keeping the system architecture as simple and replicable as possible. To simplify training, we learn our model from the same orientation counts that power the relative-frequency HRM. To simplify decoder integration, we limit our feature space to information from a single phrase pair.

In a maximum entropy model, the probability of an orientation o given a phrase pair pp is given by a log-linear model:

$$P(o|pp) = \frac{\exp(w \cdot f(o,pp))}{\sum_{o'} \exp(w \cdot f(o',pp))} \qquad (3)$$

where $f(o, pp)$ returns features of a phrase-pair, and w is the learned weight vector. We build two models, one for left-to-right translation, and one for right-to-left. As with the relative frequency model, we limit our discussion to the left-to-right model, with the other direction being symmetrical.

We construct a training example for each unique phrase-pair type (as opposed to token) found in our bitext. We use the orientation counts observed for a phrase pair pp_i to construct its example weight: $c_i = \sum_o cnt(o, pp_i)$. The same counts are used to construct a target distribution $\tilde{P}(o|pp_i)$, using the

[4] We use a historically good setting of $\alpha_* = 10$ throughout.

Base:
$bias$; $src \wedge tgt$; src; tgt
$src.first$; $src.last$; $tgt.first$; $tgt.last$
$clust_{50}(src.first)$; $clust_{50}(src.last)$
$clust_{50}(tgt.first)$; $clust_{50}(tgt.last)$
× Orientation $\{M, S, D\}$

Table 1: Features for the Maximum Entropy HRM.

unsmoothed relative frequency estimate in Equation 1. We then train our weight vector to minimize:

$$\frac{1}{2}||w||^2+C\sum_i c_i \left[\begin{array}{l} \log\sum_o \exp\left(w \cdot f(o, pp_i)\right) \\ -\sum_o \tilde{P}(o|pp_i)\left(w \cdot f(o, pp_i)\right) \end{array} \right] \tag{4}$$

where C is a hyper-parameter that controls the amount of emphasis placed on minimizing loss versus regularizing w.[5] Note that this objective is a departure from previous work, which tends to create an example for each phrase-pair token, effectively assigning $\tilde{P}(o|pp) = 1$ to a single gold-standard orientation. Instead, our model attempts to reproduce the target distribution $\tilde{P}$ for the entire type, where the penalty c_i for missing this target is determined by the frequency of the phrase pair. The resulting model will tend to match unsmoothed relative frequency estimates for very frequent phrase pairs, and will smooth intelligently using features for less frequent phrase pairs.

All of the features returned by $f(o|pp)$ are derived from the phrase pair $pp = [src, tgt]$, with the goal of describing the phrase pair at a variety of granularities. Our features are described in Table 1, using the following notation: the operators *first* and *last* return the first and last words of phrases,[6] while the operator $clust_{50}$ maps a word onto its corresponding cluster from an automatically-induced, deterministic 50-word clustering provided by `mkcls` (Och, 1999). Our use of words at the corners of phrases (as opposed to the syntactic head, or the last aligned word) follows Xiong et al. (2006), while our use of word clusters follows Zens and Ney (2006). Each feature has the orientation o appended onto it.

To help scale and to encourage smoothing, we only allow features that occur in at least 5 phrase pair tokens. Furthermore, to deal with the huge number of extracted phrase pairs (our Arabic system extracts roughly 88M distinct phrase pair types), we subsample pairs that have been observed only once, keeping only 10% of them. This reduces the number of training examples from 88M to 19M.

Base:
$src.first$; $src.last$; $tgt.first$; $tgt.last$
$top.src.first$; $top.src.last$; $top.tgt.last$
$between\_words$
× Representation {80-words, 50-clusters, 20-clusters}
× Orientation $\{M, S, D\}$

Table 2: Features for the Sparse Feature HRM.

3.3 Sparse Reordering Features

The maximum entropy approach uses features to model the distribution of orientations found in word alignments. Alternatively, a number of recent tuning methods, such as MIRA (Chiang et al., 2008) or PRO (Hopkins and May, 2011), can handle thousands of features. These could be used to tune similar features to maximize BLEU directly.

Given the appropriate tuning architecture, the sparse feature approach is actually simpler in many ways than the maximum entropy approach. There is no need to scale to millions of training examples, and there is no question of how to integrate the trained model into the decoder. Instead, one simply implements the desired features in the decoder's feature API and then tunes as normal. The challenge is to design features so that the model can be learned from small tuning sets.

The standard approach for sparse feature design in SMT is to lexicalize only on extremely frequent words, such as the top-80 words from each language (Chiang et al., 2009; Hopkins and May, 2011). We take that approach here, but we also use deterministic clusters to represent words from both languages, as provided by `mkcls`. These clusters mirror parts-of-speech quite effectively (Blunsom and Cohn, 2011), without requiring linguistic resources. They should provide useful generalization for reordering decisions. Inspired by recent successes in semi-supervised learning (Koo et al., 2008;

[5] Preliminary experiments indicated that the model is robust to the choice of C; we use $C = 0.1$ throughout.

[6] *first* $=$ *last* for a single-word phrase

corpus	sentences	words (ar)	words (en)
train	1,490,514	46,403,734	47,109,486
dev	1,663	45,243	50,550
mt08	1,360	45,002	51,341
mt09	1,313	40,684	46,813

Table 3: Arabic-English Corpus. For English dev and test sets, word counts are averaged across 4 references.

corpus	sentences	words (ch)	words (en)
train	3,505,529	65,917,610	69,453,695
dev	1,894	48,384	53,584
mt06	1,664	39,694	47,143
mt08	1,357	33,701	40,893

Table 4: Chinese-English Corpus. For English dev and test sets, word counts are averaged across 4 references.

Lin and Wu, 2009), we cluster at two granularities (20 clusters and 50 clusters), and allow the discriminative tuner to determine how to best employ the various representations.

We add the sparse features in Table 2 to our decoder to help assess reordering decisions. As with the maximum entropy model, orientation is appended to each feature. Furthermore, each feature has a different version for each of our three word representations. Like the maximum entropy model, we describe the phrase pair being added to the hypothesis in terms of the first and last words of its phrases. Unlike the maximum entropy model, we make no attempt to use entire phrases or phrase-pairs as features, as they would be far too sparse for our small tuning sets. However, due to the sparse features' direct decoder integration, we have access to a fair amount of extra context. We represent the current top of the stack (top) using its first and last source words (accessible from the HRM stack), and its last target word (accessible using language model context). Furthermore, for discontinuous (D) orientations, we can include an indicator for each source word between the current top of the stack and the phrase being added.

Because the sparse feature HRM has no access to phrase-pair or monolingual phrase features, and because it completely ignores our large supply of word-aligned training data, we view it as complimentary to the relative frequency HRM. We always include both when tuning and decoding. Furthermore, we only include sparse features in the left-to-right translation direction, as the features already consider context (top) as well as the next phrase.

4 Experimental Design

We test our reordering models in Arabic to English and Chinese to English translation tasks. Both systems are trained on data from the NIST 2012 MT evaluation; the Arabic system is summarized in Table 3 and the Chinese in Table 4. The Arabic system's development set is the NIST mt06 test set, and its test sets are mt08 and mt09. The Chinese system's development set is taken from the NIST mt05 evaluation set, augmented with some material reserved from our NIST training corpora in order to better cover newsgroup and weblog domains. Its test sets are mt06 and mt08.

4.1 Baseline System

For both language pairs, word alignment is performed by GIZA++ (Och and Ney, 2003), with 5 iterations of Model 1, HMM, Model 3 and Model 4. Phrases are extracted with a length limit of 7 from alignments symmetrized using grow-diag-final-and (Koehn et al., 2003). Conditional phrase probabilities in both directions are estimated from relative frequencies, and from lexical probabilities (Zens and Ney, 2004). 4-gram language models are estimated from the target side of the bitext with Kneser-Ney smoothing. Relative frequency and maximum entropy RMs are represented with six features, with separate weights for M, S and D in both directions (Koehn et al., 2007). HRM orientations are determined using an unrestricted shift-reduce parser (Cherry et al., 2012). We also employ a standard distortion penalty incorporating the minimum completion cost described by Moore and Quirk (2007). Our multi-stack phrase-based decoder is quite similar to Moses (Koehn et al., 2007).

For all systems, parameters are tuned with a batch-lattice variant of hope-fear MIRA (Chiang et al., 2008; Cherry and Foster, 2012). Preliminary experiments suggested that the sparse reordering features have a larger impact when tuned with lattices as opposed to n-best lists.

4.2 Evaluation

We report lower-cased BLEU (Papineni et al., 2002), evaluated using the same English tokenization used in training. For our primary results, we perform random replications of parameter tuning, as suggested by Clark et al. (2011). Each replication uses a different random seed to determine the order in which MIRA visits tuning sentences. We test for significance using Clark et al.'s MultEval tool, which uses a stratified approximate randomization test to account for multiple replications.

5 Results

We begin with a comparison of the reordering models described in this paper: the hierarchical reordering model (**HRM**), the maximum entropy HRM (**Maxent HRM**) and our sparse reordering features (**Sparse HRM**). Results are shown in Table 5.

Our three primary points of comparison have been tested with 5 replications. We report BLEU scores averaged across replications as well as standard deviations, which indicate optimizer stability. We also provide unreplicated results for two systems, one using only the distortion penalty (**No RM**), and one using a non-hierarchical reordering model (**RM**). These demonstrate that our baseline already has quite mature reordering capabilities.

The Maxent HRM has very little effect on translation performance. We found this surprising; we expected large gains from improving the reordering distributions of low-frequency phrase-pairs. See §5.1 for further exploration of this result.

The Sparse HRM, on the other hand, performs very well. It produces significant BLEU score improvements on all test sets, with improvements ranging between 1 and 1.8 BLEU points. Even with millions of training sentences for our HRM, there is a large benefit in building HRM-like features that are tuned to optimize the decoder's BLEU score on small tuning sets. We examine the impact of subsets of these features in §5.2.

The test sets' standard deviations increase from 0.1 under the baseline to 0.3 under the Sparse HRM for Chinese-English, indicating a decrease in optimizer stability. With so many features trained on so few sentences, this is not necessarily surprising. Fortunately, looking at the actual replications (not shown), we confirmed that if a replication produced low scores in one test, it also produced low scores in the other. This means that one should be able to outperform the average case by using a dev-test set to select among replications.

Base:
$src.first$; $src.last$; $tgt.first$; $tgt.last$
× Representation {80-words, 50-clusters}
× Orientation $\{M, S, D\}$

Table 6: Intersection of Maxent & Sparse HRM features.

5.1 Maximum Entropy Analysis

The next two sections examine our two solutions in detail, starting with the Maxent HRM. To avoid excessive demands on our computing resources, all experiments report tuning with a single replication with the same seed. We select Arabic-English for our analysis, as this pair has high optimizer stability and fast decoding speeds.

Why does the Maxent HRM help so little? We begin by investigating some design decisions. One possibility is that our subsampling of frequency-1 training pairs (see §3.2) harmed performance. To test the impact of this decision, we train a Maxent HRM without subsampling, taking substantially longer. The resulting BLEU scores (not shown) are well within the projected standard deviations for optimizer instability (0.1 BLEU from Table 5). This indicates that subsampling is not the problem. To confirm our choice of hyperparameters, we conduct a grid search over the Maxent HRM's regularization parameter C (see Equation 4), covering the set $\{1, 0.1, 0.01, 0.001\}$, where $C = 0.1$ is the value used throughout this paper. Again, the resulting BLEU scores (not shown) are all within 0.1 of the means reported in Table 5.

Another possibility is that the Maxent HRM has an inferior feature set. We selected features for our Maxent and Sparse HRMs to be similar, but also to play to the strengths of each method. To level the playing field, we train and test both systems with the feature set shown in Table 6, which is the intersection of the features from Tables 1 and 2. The resulting average BLEU scores are shown in Table 7. With

		Chinese-English						Arabic-English					
Method	n	tune	std	mt06	std	mt08	std	tune	std	mt08	std	mt09	std
No RM	1	24.3	–	32.0	–	26.4	–	41.7	–	41.4	–	44.1	–
RM	1	25.2	–	33.3	–	27.4	–	42.4	–	42.6	–	45.2	–
HRM (baseline)	5	25.6	0.0	34.2	0.1	28.0	0.1	42.9	0.0	42.9	0.1	45.5	0.0
HRM + Maxent HRM	5	25.6	0.0	34.3	0.1	28.1	0.1	43.0	0.0	42.9	0.0	45.6	0.1
HRM + Sparse HRM	5	28.0	0.1	**35.4**	0.3	**29.0**	0.3	47.0	0.1	**44.6**	0.1	**47.3**	0.1

Table 5: Comparing reordering methods according to BLEU score. n indicates the number of tuning replications, while standard deviations (std) indicate optimizer stability. Test scores that are significantly higher ($p < 0.01$) than the HRM baseline are highlighted in bold.

Method		−HRM	+HRM
HRM (baseline)		–	44.2
Original	Maxent HRM	44.2	44.2
	Sparse HRM	45.4	46.0
Intersection	Maxent HRM	43.8	44.2
	Sparse HRM	45.2	46.0

Table 7: Arabic-English BLEU scores with each system's original feature set versus the intersection of the two feature sets, with and without the relative frequency HRM. BLEU is averaged across mt08 and mt09.

the baseline HRM included, performance does not change for either system with the intersected feature set. Sparse features continue to help, while the maximum entropy model does not. Without the HRM, both systems degrade under the intersection, though the Sparse HRM still improves over the baseline.

Finally, we examine Maxent HRM performance as a function of the amount of word-aligned training data. To do so, we hold all aspects of our system constant, except for the amount of bitext used to train either the baseline HRM or the Maxent HRM. Importantly, the phrase table always uses the complete bitext. For our reordering training set, we hold out the final two thousand sentences of bitext to calculate perplexity. This measures the model's surprise at reordering events drawn from previously unseen alignments; lower values are better. We proceed to subsample sentence pairs from the remaining bitext, in order to produce a series of training bitexts of increasing size. We subsample with the probability of accepting a sentence pair, P_a, set to $\{0.001, 0.01, 0.1, 1\}$. It is important to not confuse this subsampling of sentence pairs with the subsampling of low-frequency phrase pairs (see §3.2), which is still carried out by the Maxent HRM for each training scenario.

Figure 1 shows how BLEU (averaged across both test sets) and perplexity vary as training data increases from 1.5K sentences to the full 1.5M. At $P_a < 0.1$, corresponding to less than 150K sentences, the maximum entropy model actually makes a substantial difference in terms of BLEU. However, these deltas narrow to nothing as we reach millions of training sentences. This is consistent with the results of Nguyen et al. (2009), who report that maximum entropy reordering outperforms a similar baseline, but using only 50K sentence pairs.

A related observation is that held-out perplexity does not seem to predict BLEU in any useful way. In particular, perplexity does not predict that the two systems will become similar as data grows, nor does it predict that maxent's performance will level off. Predicting the orientations of unseen alignments is not the same task as predicting the orientation for a phrase during translation. We suspect that perplexity places too much emphasis on rare or previously unseen phrase pairs, due to phrase extraction's heavy tails. Preliminary attempts to correct for this using absolute discounting on the test counts did not resolve these issues. Unfortunately, in maximizing (regularized or smoothed) likelihood, both maxent and relative frequency HRMs are chasing the perplexity objective, not the BLEU objective.

5.2 Sparse Feature Analysis

The results in Table 7 from §5.1 already provide us with a number of insights regarding the Sparse HRM. First, note that the intersected feature set uses only information found within a single phrase. The fact that the Sparse HRM performs so well with

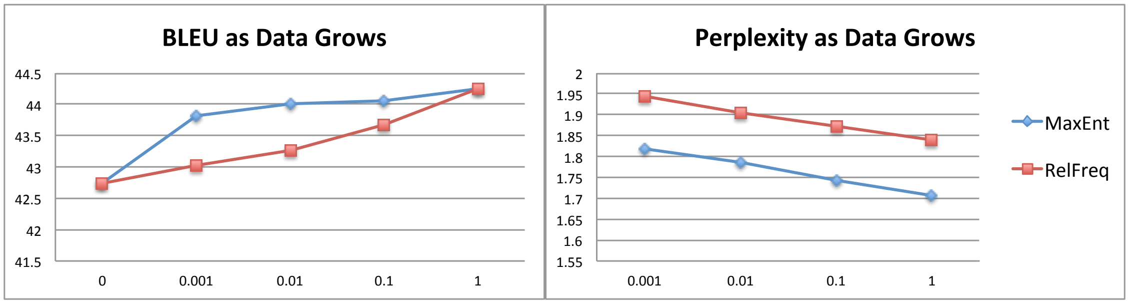

Figure 1: Learning curves for Relative Frequency and Maximum Entropy reordering models on Arabic-English.

Feature Group	Count	BLEU
No Sparse HRM	0	44.2
Between	312	44.4
Stack	1404	45.2
Phrase	1872	45.9
20 Clusters	506	45.4
50 Clusters	1196	45.8
80 Words	1886	45.8
Full Sparse HRM	3588	46.0

Table 8: Versions of the Sparse HRM built using organized subsets of the complete feature set for Arabic-English. Count is the number of distinct features, while BLEU is averaged over mt08 and mt09.

intersected features indicates that modeling context outside a phrase is not essential for strong performance. Furthermore, the −**HRM** portion of the table indicates that the sparse HRM does not require the baseline HRM to be present in order to outperform it. This is remarkable when one considers that the Sparse HRM uses less than 4k features to model phrase orientations, compared to the 530M probabilities[7] maintained by the baseline HRM's relative frequency model.

To determine which feature groups are most important, we tested the Sparse HRM on Arabic-English with a number of feature subsets. We report BLEU scores averaged over both test sets in Table 8. First, we break our features into three groups according to what part of the hypothesis is used to assess orientation. For each of these location groups, all forms of word representation (clusters or frequent words) are employed. The groups consist of **Between**: the words between the top of the stack and the phrase to be added; **Stack**: words describing the current top of the stack; and **Phrase**: words describing the phrase pair being added to the hypothesis. Each group was tested alone to measure its usefulness. This results in a clear hierarchy, with the phrase features being the most useful (nearly as useful as the complete system), and the between features being the least. Second, we break our features into three groups according to how words are represented. For each of these representation groups, all location groups (Between, Stack and Phrase) are employed. The groups are quite intuitive: **20 Clusters**, **50 Clusters** or **80 Words**. The differences between representations are much less dramatic than the location groups. All representations perform well on their own, with the finer-grained ones performing better. Including multiple representations provides a slight boost, but these experiments suggest that a leaner model could certainly drop one or two representations with little impact.

In its current implementation, the Sparse HRM is roughly 4 times slower than the baseline decoder. Our sparse feature infrastructure is designed for flexibility, not speed. To affect reordering, each sparse feature template is re-applied with each hypothesis extension. However, the intersected feature set from §5.1 is only 2 times slower, and could be made faster still. That feature set uses only within-phrase features to asses orientations; therefore, the total weight for each orientation for each phrase-pair could be pre-calculated, making its cost comparable to the baseline.

[7] 88.4M phrase pairs × 3 orientations (M, S and D) × 2 translation directions (left-to-right and right-to-left).

Chinese-English	tune	mt06	mt08
Base	27.7	39.9	33.7
+Sparse HRM	29.2	41.0	34.1

Arabic-English	tune	mt08	mt09
Base	49.6	49.1	51.6
+Sparse HRM	51.7	49.9	52.2

Table 9: The effect of Sparse HRMs on complex systems.

5.3 Impact on Competition-Grade SMT

Thus far, we have employed a baseline that has been designed for both translation quality and replicability. We now investigate the impact of our Sparse HRM on a far more complex baseline: our internal system used for MT competitions such as NIST.

The Arabic system uses roughly the same bilingual data as our original baseline, but also includes a 5-gram language model learned from the English Gigaword. The Chinese system adds the UN bitext as well as the English Gigaword. Both systems make heavy use of linear mixtures to create refined translation and language models, mixing across sources of corpora, genre and translation direction (Foster and Kuhn, 2007; Goutte et al., 2009). They also mix many different sources of word alignments, with the system adapting across alignment sources using either binary indicators or linear mixtures. Importantly, these systems already incorporate thousands of sparse features as described by Hopkins and May (2011). These provide additional information for each phrase pair through frequency bins, phrase-length bins, and indicators for frequent alignment pairs. Both systems include a standard HRM.

The result of adding the Sparse HRM to these systems is shown in Table 9. Improvements range from 0.4 to 1.1 BLEU, but importantly, all four test sets improve. The impact of these reordering features is reduced slightly in the presence of more carefully tuned translation and language models, but they remain a strong contributor to translation quality.

6 Conclusion

We have shown that sparse reordering features can improve the quality of phrase-based translations, even in the presence of lexicalized reordering models that track the same orientations. We have compared this solution to a maximum entropy model, which does not improve our HRM baseline. Our analysis of the maximum entropy solution indicates that smoothing the orientation estimates is not a major concern in the presence of millions of sentences of bitext. This implies that our sparse features are achieving their improvement because they optimize BLEU with the decoder in the loop, side-stepping the objective mismatch that can occur when training on word-aligned data. The fact that this is possible with such small tuning corpora is both surprising and encouraging.

In the future, we would like to investigate how to incorporate useful future cost estimates for our sparse reordering features. Previous work has shown future distortion penalty estimates to be important for both translation speed and quality (Moore and Quirk, 2007; Green et al., 2010), but we have ignored future costs in this work. We would also like to investigate features inspired by transition-based parsing, such as features that look further down the reordering stack. Finally, as there is evidence that ideas from lexicalized reordering can help hierarchical phrase-based SMT (Huck et al., 2012), it would be interesting to explore the use of sparse RMs in that setting.

Acknowledgments

Thanks to George Foster, Roland Kuhn and the anonymous reviewers for their valuable comments on an earlier draft.

References

Phil Blunsom and Trevor Cohn. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *ACL*, pages 865–874, Portland, Oregon, USA, June.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *HLT-NAACL*, pages 427–436, Montréal, Canada, June.

Colin Cherry, Robert C. Moore, and Chris Quirk. 2012. On hierarchical re-ordering and permutation parsing for phrase-based decoding. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 200–209, Montréal, Canada, June.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *EMNLP*, pages 224–233.

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

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL*, pages 176–181.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 128–135.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*, pages 848–856, Honolulu, Hawaii.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *HLT-NAACL*, Montreal, Canada, June.

Cyril Goutte, David Kurokawa, and Pierre Isabelle. 2009. Improving SMT by learning the translation direction. In *EAMT Workshop on Statistical Multilingual Analysis for Retrieval and Translation*.

Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *HLT-NAACL*, pages 867–875, Los Angeles, California, June.

Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 292–301, Jeju Island, Korea, July.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *EMNLP*, pages 1352–1362.

Matthias Huck, Stephan Peitz, Markus Freitag, and Hermann Ney. 2012. Discriminative reordering extensions for hierarchical phrase-based machine translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 313–320, Trento, Italy, May.

Philipp Koehn, Franz Joesef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*, pages 127–133.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings to the International Workshop on Spoken Language Translation (IWSLT)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180, Prague, Czech Republic, June.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *ACL*, pages 595–603, Columbus, Ohio, June.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the ACL and the AFNLP*, pages 1030–1038, Singapore, August.

Robert C. Moore and Chris Quirk. 2007. Faster beam-search decoding for phrasal statistical machine translation. In *MT Summit XI*, September.

Vinh Van Nguyen, Akira Shimazu, Minh Le Nguyen, and Thai Phuong Nguyen. 2009. Improving a lexicalized hierarchical reordering model using maximum entropy. In *MT Summit XII*, Ottawa, Canada, August.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), March.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4), December.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *EACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *ACL*, pages 11–21, Jeju Island, Korea, July.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL*, pages 101–104, Boston, USA, May.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *COLING-ACL*, pages 521–528, Sydney, Australia, July.

Sirvan Yahyaei and Christof Monz. 2010. Dynamic distortion in a discriminative reordering model for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 353–360.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *HLT-NAACL*, pages 257–264, Boston, USA, May.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 55–63, New York City, June.