7-2011

Generative Models of Monolingual and Bilingual Gappy Patterns

Kevin Gimpel
Carnegie Mellon University

Noah A. Smith
Carnegie Mellon University, nasmith@cs.cmu.edu

Follow this and additional works at: http://repository.cmu.edu/lti

Published In

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Language Technologies Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.
Generative Models of Monolingual and Bilingual Gappy Patterns

Kevin Gimpel  Noah A. Smith
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{kgimpel,nasmith}@cs.cmu.edu

Abstract

A growing body of machine translation research aims to exploit lexical patterns (e.g., n-grams and phrase pairs) with gaps (Simard et al., 2005; Chiang, 2005; Xiong et al., 2011). Typically, these “gappy patterns” are discovered using heuristics based on word alignments or local statistics such as mutual information. In this paper, we develop generative models of monolingual and parallel text that build sentences using gappy patterns of arbitrary length and with arbitrarily many gaps. We exploit Bayesian nonparametrics and collapsed Gibbs sampling to discover salient patterns in a corpus. We evaluate the patterns qualitatively and also add them as features to an MT system, reporting promising preliminary results.

1 Introduction

Beginning with the success of phrase-based translation models (Koehn et al., 2003), a trend arose of modeling larger and increasingly complex structural units in translation. One thread of work has focused on the use of lexical patterns with gaps. Simard et al. (2005) proposed using phrase pairs with gaps in a phrase-based translation model, providing a heuristic method to extract gappy phrase pairs from word-aligned parallel corpora. The widely-used hierarchical phrase-based translation framework was introduced by Chiang (2005) and also relies on a simple heuristic for phrase pair extraction. On the monolingual side, researchers have taken inspiration from trigger-based language modeling for speech recognition (Rosenfeld, 1996). Xiong et al. (2011) used Rosenfeld’s mutual information procedure to discover trigger pairs and added a single feature to a phrase-based MT system that scores new words based on all potential triggers from previous parts of the derivation. We are not aware of prior work that uses generative modeling and Bayesian nonparametrics to discover these same types of patterns automatically; doing so allows us to discover larger patterns with more words and gaps if they are warranted by the data.

In addition to the gappy phrase-based (Simard et al., 2005) and hierarchical phrase-based (Chiang, 2005) models mentioned earlier, other researchers have explored the use of bilingual gappy structures for machine translation. Crego and Yvon (2009) and

2 Related Work

There is a rich history of trigger-based language modeling in the speech recognition community, typically involving the use of statistical tests to discover useful trigger-word pairs (Rosenfeld, 1996; Jelinek, 1997). Xiong et al. (2011) used Rosenfeld’s mutual information procedure to discover trigger pairs and added a single feature to a phrase-based MT system that scores new words based on all potential triggers from previous parts of the derivation. We are not aware of prior work that uses generative modeling and Bayesian nonparametrics to discover these same types of patterns automatically; doing so allows us to discover larger patterns with more words and gaps if they are warranted by the data.

In addition to the gappy phrase-based (Simard et al., 2005) and hierarchical phrase-based (Chiang, 2005) models mentioned earlier, other researchers have explored the use of bilingual gappy structures for machine translation. Crego and Yvon (2009) and
Galley and Manning (2010) proposed ways of incorporating phrase pairs with gaps into standard left-to-right decoding algorithms familiar to phrase-based and N-gram-based MT; both used heuristics to extract phrase pairs. Bansal et al. (2011) presented a model and training procedure for word alignment that uses phrase pairs with gaps. They use a semi-Markov model with an enlarged dynamic programming state in order to represent alignment between gappy phrases. Their model permits up to one gap per phrase while our models permit an arbitrary number.

Figure 1: A sentence from the news commentary corpus, along with color assignments for the words and the \( \pi \) function for each color.

The generative story for a single sentence follows:

1. Sample the number of words: \( n \sim \text{Poisson}(\beta) \)
2. Sample the number of unique colors in the sentence given \( n \): \( m \sim \text{Uniform}(1, n) \)
3. For each word index \( i = 1 \ldots n \), sample the color of word \( i \): \( c_i \sim \text{Uniform}(1, m) \). If any of the \( m \) colors has no words, repeat this step.
4. For each color \( j = 1 \ldots m \), sample from a multinomial distribution over patterns: \( \mathbf{w}_{C_j} \sim \text{Mult}(\mu) \). If the words \( \mathbf{w}_{C_j} \) are not consistent with the color assignments, i.e., wrong number of words or gaps, gaps not in the correct locations, repeat this step.

Thus, the probability of generating number of words \( n \), words \( \mathbf{w}_{1:n} \), color assignments \( \mathbf{c}_{1:n} \), and number of colors \( m \) is

\[
\begin{align*}
 p(\mathbf{w}_{1:n}, \mathbf{c}_{1:n}, m | \beta, \mu) &= \frac{1}{Z} \left( \frac{\beta^n}{n!} e^{-\beta} \right) \left( \frac{1}{n} \right) \left( \frac{1}{m} \right)^m \prod_{j=1}^{m} p_\mu(\pi(C_j)) \\
\end{align*}
\]

where \( Z \) is a normalization constant required by the potential repetition of sampling in the final two steps of the generative story. Without \( Z \), the model would be deficient as we would waste probability mass on internally inconsistent color assignments.

The core of the model is a single multinomial distribution \( p_\mu(\cdot) \) over patterns. We use a Dirichlet process (DP) prior for this multinomial so that we can model an unbounded set of patterns: \( \mu \sim \text{DP}(\alpha, P_0) \), where \( \alpha \) is the concentration parameter and \( P_0 \) is the base distribution. The base distribution includes a \( \text{Poisson}(\nu) \) over the number of words in the pattern, a uniform distribution (over word types in the vocabulary) for each word, a uniform distribution over the number of gaps given the number of words, and a uniform distribution over the arrangement of gaps given the numbers of gaps and words.\(^2\)

\(^{2}\)The number of ways of arranging \( y \) gaps among \( x \) words is \( \binom{x-1}{y} \).

\(^{3}\)Since we assume the words are given, \( \beta \) is irrelevant.
During each iteration of Gibbs sampling, we proceed through the corpus and sample a new value for each \(c_i\) variable conditioned on the values of all others in the corpus. The \(m\) variables are determined by the \(c_i\) variables and therefore do not need to be sampled directly. When sampling \(c_i\), we first remove \(c_i\) from the corpus (and its color if the color only contained \(i\)). Where the remaining colors in the sentence are numbered from 1 to \(m\), there are \(m + 1\) possibilities for \(c_i\): \(m\) for each of the existing colors and one for choosing a new color.

Since choosing a new color corresponds to creating a new instance of the pattern \(\pi(\{i\})\), the probability of choosing a new color \(m + 1\) is proportional to

\[
\frac{\#\pi(\{i\}) + \alpha P_0(\pi(\{i\}))}{\# + \alpha}
\]

where \(\#\pi\) is the count of pattern \(\pi\) in the rest of the sentence and all other sentences in the corpus, and \(\#\) is the total count of all patterns in this same set. The probability of choosing the existing color \(j\) (for \(1 \leq j \leq m\)) is proportional to

\[
\frac{\#\pi(C_j \cup \{i\}) + \alpha P_0(\pi(C_j \cup \{i\}))}{\#\pi(C_j) + \alpha P_0(\pi(C_j))}
\]

where the denominator encodes the fact that the move will cause an instance of the pattern for the color \(C_j\) to be removed from the corpus as the new pattern for \(C_j \cup \{i\}\) is added.

We note that, even though these two types of moves will result in different numbers of colors \((m)\) in the sentence, we do not have to include a term for this in the sampler because we use a uniform distribution for \(m\) and therefore all \((\text{valid})\) numbers of colors have the same probability. The normalization constant \(Z\) in Equation 1 does not affect inference because our sampler is designed to only consider valid (i.e., internally consistent) settings for the \(e^{(i)}\) and \(m^{(i)}\) variables.

This model makes few assumptions, using uniform distributions whenever possible. This simplifies inference and causes the resulting lexicon to be influenced primarily by the “rich-get-richer” effect of the DP prior. Despite its simplicity, we will show later that this model discovers patterns that capture a variety of linguistic phenomena.

4 Bilingual Pattern Models

We now present a generative model for a sentence pair that will enable us to discover bilingual patterns. In this section we present one example of extending the previous model to be bilingual, but we note that many other extensions are possible; indeed, flexibility is one of the key advantages of working within the framework of probabilistic modeling.

We assume that we are given sentence pairs and one-to-one word alignments. That is, in addition to an \(n\)-word target sentence \(w_{1:n}\), we assume we have an \(n'\)-word source sentence \(w'_{1:n'}\) and word alignments \(a_{1:n'}\), where \(a_i = j\) iff \(w'_{a_i}\) is aligned to \(w_j\) and \(a_i = 0\) if \(w'_{a_i}\) is aligned to null.

To model bilingual patterns, we distinguish source colors from target colors. A target-language word can only be colored with a target color, but a source word can be colored with either a source color or with the target color of the target word it is aligned to (if any). We have \(m\) target colors as before and now add \(m'\) source colors. We introduce additional random variables in the form of a binary vector \(g\) of length \(n'\) that indicates, for each source word, whether or not it accepts the color of its aligned target word. We introduce an additional parameter \(\gamma\) for the probability that a source word will accept the color of its aligned word. We fix its value to 0.5 and do not learn it during inference. Figure 2 shows an example Spanish-English sentence pair with automatic word alignments and color assignments. The bilingual patterns for a few target colors are shown.

The generative story for a sentence pair follows:

1. Sample the numbers of words in the source and target sentences: \(n', n \sim \text{Poisson}(\beta)\)
2. Sample the numbers of source and target colors given \( n', n \): \( m' \sim \text{Uniform}(1, n') \), \( m \sim \text{Uniform}(1, n) \)

3. Sample the alignment vector from any distribution that ensures links are 1-to-1:\(^4 a_{1:n'} \sim p(a)\)

4. For each target word index \( i = 1 \ldots n \), sample the color of target word \( i \) from a uniform distribution over all target colors: \( c_i \sim \text{Uniform}(1, m) \). While any of the \( m \) colors has no words, repeat this step.

5. For each source word index \( i = 1 \ldots n' \):
   1. Decide whether to use a source color or to use the target color of the aligned target word: \( g_i \sim p_\gamma(g_i \mid a_i) \)
   2. If \( g_i = 1 \), set \( c_i' = c_{a_i} \); otherwise, sample a source color: \( c_i' \sim \text{Uniform}(1, m') \)
   3. If any source color has no words, repeat Step 5.

6. For each source color \( j = 1 \ldots m' \):
   1. Sample from a multinomial over source patterns: \( w_{C_j'} \sim \text{Mult}(\mu') \). While the words \( w_{C_j'} \) are not consistent with the color assignments, repeat this step.

8. For each target color \( j = 1 \ldots m \):
   1. Sample from a multinomial over bilingual patterns: \( w_{C_j} \sim \text{Mult}(\mu) \). While the words \( w_{C_j} \) are not consistent with the color assignments, repeat this step.

The distribution \( p_\gamma(g_i \mid a_i) \) is defined below:

\[
\begin{align*}
p_\gamma(g_i = 1 \mid a_i \neq -1) &= \gamma \\
p_\gamma(g_i = 1 \mid a_i = -1) &= 0
\end{align*}
\]

where \( \gamma \) determines how frequently source tokens will be added to target patterns.

The probability of generating target words \( w_{1:n} \), source words \( w'_{1:n'} \), alignments \( a_{1:n'} \), target color assignments \( c_{1:n'} \), source color assignments \( c_{1:n'}' \), color propagation variables \( g_{1:n'} \), number of source colors \( m \), and number of source colors \( m' \) is

\[
\frac{1}{Z} p(n)p(n')p(m \mid n)p(m' \mid n')p(a_{1:n'})
\]

\[
\times \left( \prod_{i=1}^{n} p(c_i \mid m) \right)
\]

\[
\times \left( \prod_{i=1}^{n'} p_\gamma(g_i \mid a_i)p(c_i' \mid m')^{I[g_i=-0]} \right)
\]

\[
\times \left( \prod_{j=1}^{m'} p_\mu'(\pi(C_j')) \right) \left( \prod_{j=1}^{m} p_\mu(\pi(C_j)) \right)
\]

where \( Z \) again serves as a normalization constant to prevent the model from leaking probability mass on internally inconsistent configurations.

There are now two multinomial distributions over patterns with parameter vectors \( \mu \) and \( \mu' \). They both use DP priors with identical concentration parameters \( \alpha \) and differing base distributions \( P_0 \) and \( P_0' \). The base distribution for source patterns, \( P_0' \), takes the same form as the base distribution for the model described in §3.

For target patterns with aligned source words, \( P_0 \) generates the target part of the pattern like the base distribution in §3 and then generates the number of aligned source words to each target word with a Poisson(1) distribution; the number of aligned source words can only be 0 or 1 when all word links are 1-to-1. If it is 1, the base distribution generates the aligned source word by sampling uniformly from among all source types.

While there are connections between this model and work on performing translation using phrase pairs with gaps, the patterns we discover are not guaranteed to be bilingual translation units. Rather, they typically contain additional target-side words that have no explicit correlate on the source side. They can be used to assist an existing translation model by helping to choose the best phrase translation for each source phrase. To define a generative model for phrase pairs with gaps, changes would have to be made to the bilingual model we presented.

**Inference** As before, we use collapsed Gibbs sampling for inference. Our goal is to obtain samples from the posterior \( p(\langle \{ c', g, m, m' \rangle i = 1 \rangle \mid \{ w, w', a_1(i) \rangle i = 1 \rangle) \).
We go through each sentence pair and sample new color assignment variables for each word. For an aligned word pair \((w'_i, w_j)\), we sample a new value for the tuple \((g_i, c'_i, c_j)\). The possible values for \(c_j\) include all target colors, including a new target color. The possible values for \(g_i\) are 0, in which case \(c'_i\) can be any of the source colors, including a new source color, and 1, for which \(c'_i\) must be \(c_j\). For an unaligned target word \(w_j\), \(c_j\) can be any target color, including a new one, and for an unaligned source word \(w'_i\), \(c'_i\) can be any source color, including a new one. The full equations for sampling can be easily derived using the equations from §3.

5 Evaluation

We conducted evaluation to determine (1) what types of phenomena are captured by the most probable patterns discovered by our models, and (2) whether including the patterns as features can improve translation quality.

5.1 Qualitative Evaluation

5.1.1 Monolingual Model

Since inference is computationally expensive, we used the 126K-sentence English news commentary corpus provided for the WMT shared tasks (Callison-Burch et al., 2010). We ran Gibbs sampling for 600 iterations through the data, discarding the first 300 samples for burn-in and computing statistics of the patterns using the remaining 300 samples. Each iteration took approximately 3 minutes on a single 2.2GHz CPU. When looking primarily at the most frequent patterns, we found that this list did not vary much when only using half of the data instead. We set \(\nu = 3\) and \(\alpha = 100\); we found these hyperparameters to have only minor effects on the results.

Since many frequent patterns include the period (.), we found it useful to constrain the model to treat this token differently: we modify the base distribution so that it assigns zero probability to patterns that contain a period along with other words and we force each occurrence of a period to be alone in its own pattern during initialization. We do not need to change the inference procedure at all; with the modified base distribution and with no patterns including a period with other words, the probability of creating a new illegal pattern during inference is always zero (Eq. 3).

We also perform inference on a transformed version of the corpus in which every word is replaced with its hard word class obtained from Brown clustering (Brown et al., 1992). One property of Brown clusters is that each function word effectively receives its own class, as each ends up in a cluster in which it occupies \(\geq 95\%\) of the token counts of all types in the cluster. We call clusters that satisfy this property singleton clusters.

To obtain Brown clusters for the source and target languages, we used code from Liang (2005).\(^5\) We used the data from the news commentary corpus along with the first 500K sentences of the additional monolingual newswire data also provided for the WMT shared tasks. We used 300 clusters, ignoring words that appeared only once in this corpus. We did not use the hierarchical information from the clusters but merely converted each cluster name into a unique integer, using one additional integer for unknown words.

We used the same values for \(\nu\) and \(\alpha\) as above but ran Gibbs sampling for 1,300 iterations, again using the last 300 for collecting statistics on patterns. Judging by the number of color assignments changed on each iteration, the sampler takes longer to converge when run on word clusters than on words. As above, we constrain the singleton word cluster corresponding to the period to be alone during both initialization and inference.

\(^5\)http://www.cs.berkeley.edu/~pliang/software

Table 1: Top-ranked gappy patterns from samples according to \(p(\pi)\); patterns without gaps are omitted. The special string "_" represents a gap that can be filled by any nonempty sequence of words.
### Pattern Ranking Statistics

Several choices exist for ranking patterns. The simplest is to take the pattern count from the posterior samples, averaged over all sampling iterations after burn-in. We refer to this criterion as the **marginal probability**:

\[
p(\pi) = \frac{\#_\pi}{\#}
\]

where \(#_\pi\) is the average count of the pattern across the posterior samples and \(#\) is the count of all patterns. The top-ranked gappy patterns under this criterion are shown in Table 1. While many of these patterns match our intuitions, there are also several that are highly-ranked simply because their constituent words are frequent.

Alternatively, we can rank patterns by the **conditional probability** of the pattern given the words that comprise it:

\[
p(\pi | w(\pi)) = \frac{\#_\pi}{\#_w(\pi)}
\]

where \(w(\pi)\) returns the sequence of words in the pattern \(\pi\) and \(#_w(\pi)\) is the number of occurrences of this sequence of words in the corpus that are compatible with pattern \(\pi\). The ranking of patterns under this criterion is shown in Table 2. This method favors precision but also causes very rare patterns to be highly ranked.

To address this, we also consider a product-of-experts model by simply multiplying together the two probabilities, resulting in the ranking shown in Table 3. This ranking is similar to that in Table 1 but penalizes patterns that are only ranked highly because they consist of common words. Table 4 shows a manual grouping of these highly-ranked patterns into several categories. We show both lexical and Brown cluster patterns.\(^6\)

It is common in both types of patterns to find long-distance dependencies involving punctuation near the top of the ranking. Among agreement patterns, the lexical model finds relationships between pronouns and their associated possessive adjectives while the cluster model finds more general patterns involving classes of nouns. Cluster patterns are more likely to capture topicality within a sentence, while the finer granularity of the lexical model is required to identify constructions like those shown (verbs triggering particular prepositions).

There are also many probable patterns without gaps, shown at the bottom of Table 4. From these patterns we can see that our models can also be used to find collocations, but we note that these are discovered in the context of the gappy patterns. That is, due to the use of latent variables in our models (the color assignments), there is a natural trading-off effect whereby the gappy patterns encourage particular non-gappy patterns to be used, and vice versa.

#### 5.1.2 Bilingual Model

We use the news commentary corpus for each language and take the intersection of GIZA++ (Och and Ney, 2003) word alignments in each direction, thereby ensuring that they are 1-to-1 alignments. We ran Gibbs sampling for 300 iterations, averaging pattern counts from the last 200. We set \(\alpha = 100\), \(\lambda = 3\), and \(\gamma = 0.5\). We ran the model in 3 conditions: source words, target words; source clusters, target clusters; and source clusters, target words. We

---

\(^6\)We filter Brown cluster patterns in which every cluster is a singleton, since these patterns are typically already accounted for in the lexical patterns.
<table>
<thead>
<tr>
<th>Rank</th>
<th>Gappy Lexical Patterns</th>
<th>Rank</th>
<th>Gappy Brown Cluster Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot; &quot;</td>
<td>2</td>
<td>(what, why, whom, whatever)</td>
</tr>
<tr>
<td>2</td>
<td>( &quot; &quot; )</td>
<td>6</td>
<td>&quot; &quot; (&quot; &quot;, &quot; &quot;)</td>
</tr>
<tr>
<td>5</td>
<td>according to</td>
<td>28</td>
<td>(accorded, compared, subscribe, thanks, referring)</td>
</tr>
<tr>
<td>9</td>
<td>why _ ?</td>
<td>178</td>
<td>(++, --) {even, especially, particularly, mostly, mainly}</td>
</tr>
<tr>
<td>63</td>
<td>according to</td>
<td>239</td>
<td>{obama, bush, clinton, mccain, brown}</td>
</tr>
</tbody>
</table>

Table 4: Gappy patterns manually divided into categories of long-distance dependencies. Patterns were ranked according to $p(\pi)p(\pi|w(\pi))$ and manually selected from the top 300 to exemplify categories. Lower pane shows top ranked non-gappy patterns. Clusters are shown as enough words to cover 95% of the token counts of the cluster, up to a maximum of 5.

Again ensured that the period and its word class remained isolated in their own patterns for each condition. We note that no source-side word order information is contained within these bilingual patterns; aligned source words can be in any order in the source sentence and the pattern will still match. The most probable patterns included many monolingual source-only and target-only patterns that are similar to those shown in Table 4. There were also many phrase pairs with gaps like those that are com-
monly extracted by heuristics (Galley and Manning, 2010). Additionally we noted examples of source words triggering more target-side information than merely one word. There were several examples of patterns that encouraged inclusion of the subject in English when translating from Spanish, as Spanish often drops the subject when it is clear from context, e.g., “we are(estamos)”. Also, one probable pattern for German-English was “the _ of the (des)” (des is aligned to the final the). The German determiner des is in the genitive case, so this pattern helps to encourage its object to also be in the genitive case when translated.

5.2 Quantitative Evaluation

We consider the Spanish-to-English (ES→EN) translation task from the ACL-2010 Workshop on Statistical Machine Translation (Callison-Burch et al., 2010). We trained a Moses system (Koehn et al., 2007) following the baseline training instructions for the shared task.7 In particular, we performed word alignment in each direction using GIZA++ (Och and Ney, 2003), used the “grow-diag-final-and” heuristic for symmetrization, and extracted phrase pairs up to a maximum length of seven. After filtering sentence pairs with one sentence longer than 50 words, we ended up with 1.45M sentence pairs of Europarl data and 91K sentence pairs of news commentary data. Language models (N = 5) were estimated using the SRI language modeling toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1998). Language models were trained on the target side of the parallel corpus as well as the first 5 million additional sentences from the extra English monolingual newswire data provided for the shared tasks. We used news-test2008 for tuning and news-test2009 for testing.

We also consider Chinese-English (ZH→EN) and followed a similar training procedure as above. We used 303K sentence pairs from the FBIS corpus (LDC2003E14) and segmented the Chinese data using the Stanford Chinese segmenter in “CTB” mode (Chang et al., 2008), giving us 7.9M Chinese words and 9.4M English words. A trigram language model was estimated using modified Kneser-Ney smoothing from the English side of the parallel corpus concatenated with 200M words of randomly-selected sentences from the Gigaword v4 corpus (excluding the NY Times and LA Times). We used NIST MT03 for tuning and NIST MT05 for testing. For evaluation, we used case-insensitive IBM BLEU (Papineni et al., 2001).

5.2.1 Training and Decoding

Unlike n-gram language models, our models have latent structure (the color assignments), making it difficult to compute the probability of a translation during decoding. We leave this problem for future work and instead simply add a feature for each of the most probable patterns discovered by our models. Each feature counts the number of occurrences of its pattern in the translation.

We wish to add thousands of features to our model, but the standard training algorithm – minimum error rate training (MERT; Och, 2003) – cannot handle large numbers of features. So, we leverage recent work on feature-rich training for MT using online discriminative learning algorithms. Our training procedure is shown as Algorithm 1. We find it convenient to notationally distinguish feature weights for the standard Moses features (λ) from weights for our pattern features (θ). We use h(θ) to denote the feature vector for translation e. The function Bi(t) returns the sentence BLEU score for translation t given reference ei (i.e., treating the sentence pair as a corpus).8

MERT is run to convergence on the tuning set to obtain weights for the standard Moses features (line 1). Phrase lattices (Ueffing et al., 2002) are generated for all source sentences in the tuning set using the trained weights λM (line 2). The lattices are used within a modified version of the margin-infused relaxed algorithm (MIRA; Crammer et al., 2006) for structured max-margin learning (lines 5-15). A k-best list is extracted from the current lattice (line 7), then the translations on the k-best list with the highest and lowest sentence-level BLEU scores are found (lines 8 and 9). The step size is then computed using the standard MIRA formula (lines 10-11) and the update is made (line 12). The returned weights are averaged over all updates.

This training procedure is inspired by several

---

7www.statmt.org/wmt10/baseline.html.

8When computing sentence BLEU, we smooth by replacing precisions of 0.0 with 0.01.
features with weights

and Chiang, 2007) to incorporate the gappy pattern

tence using Moses with its standard set of features

to training: we generate lattices for each source sen-

BLEU score than any on the

k

e

previous iterations and use it as

λ

n

BLEU-best translation on each

others that have been shown to be effective for

MT (Liang et al., 2006; Arun and Koehn, 2007;

Watanabe et al., 2007; Chiang et al., 2008). Though

not shown in the algorithm, in practice we store the

BLEU-best translation on each k-best list from all

previous iterations and use it as e+ if it has a higher

BLEU score than any on the k-best list on the cur-

rent iteration.

At decoding time, we follow a procedure similar
to training: we generate lattices for each source sen-
tence using Moses with its standard set of features and

using weights λM. We rescore the lattices using λ∗ and use cube pruning (Chiang, 2007; Huang

and Chiang, 2007) to incorporate the gappy pattern features with weights θ∗. Cube pruning is necessary

because the pattern features may match anywhere in

the translation; thus they are non-local in the phrase

lattice and require approximate inference.

5.3 Training Algorithm Comparison

Before adding pattern features, we evaluate our

training algorithm by comparing it to MERT using

the same standard Moses features. As the ini-

Table 5: Comparing MERT to our training procedure. All

numbers are %BLEU.

<table>
<thead>
<tr>
<th></th>
<th>ES→EN</th>
<th>ZH→EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERT</td>
<td>25.64</td>
<td>32.47</td>
</tr>
<tr>
<td>Alg. 1</td>
<td>25.85</td>
<td>32.33</td>
</tr>
</tbody>
</table>

5.4 Feature Preparation

We chose monolingual and bilingual pattern features using the posterior samples obtained via the inference

procedures described above. We ranked patterns using the product-of-experts formula, removed patterns

consisting of only a single token, and added the top 10K patterns from the lexical model and the top

15K patterns from the Brown cluster model. For simplicity of implementation, we skipped over pat-

terns with 3 or more gaps and patterns with 2 gaps and more than 3 total words; this procedure skipped

fewer than 1% of the top patterns. For results with bilingual pattern features, we added 15K pattern fea-

tures (5K word-word, 5K cluster-cluster, and 5K cluster-word).

5.5 Results

The first set of results is shown in Table 6. The

first row is the same as in Table 5, the second

row adds monolingual pattern features, the third

adds bilingual pattern features, and the final row

includes both sets. While gains are modest overall,
the pattern features show an encouraging improvement of 0.48 BLEU for ZH→EN. This is similar to the improvement reported by Xiong et al. (2011) (+0.4 BLEU when adding their trigger pair language model). While bilingual patterns give an improvement of 0.35 BLEU, using both monolingual and bilingual features in the same model does not provide additional improvement over monolingual features alone.

For ES→EN, the pattern features have only small effects on BLEU; we suspect that the decreased BLEU score for the full feature set is due to overfitting. It is unclear why the results differ for the two language pairs. One possibility is the use of only a single reference translation when tuning and testing with ES→EN while four references were used for ZH→EN. Another possibility is that our pattern features are correcting some of the mid- to long-range reorderings that are known to be problematic for phrase-based modeling of ZH→EN translation. ES→EN exhibits less long-range reordering and therefore may not benefit as much from our patterns.

Table 7 shows additional ZH→EN results when varying the method of ranking patterns. When using both sets of features, the “Ranking” column contains the criterion for ranking monolingual patterns; bilingual patterns are always ranked using

<table>
<thead>
<tr>
<th>Ranking</th>
<th>%BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>N/A</td>
</tr>
<tr>
<td>MONOPATS</td>
<td>32.65</td>
</tr>
<tr>
<td>MONOPATS</td>
<td>32.53</td>
</tr>
<tr>
<td>MONOPATS</td>
<td>32.81</td>
</tr>
<tr>
<td>BIPATS</td>
<td>32.68</td>
</tr>
<tr>
<td>MONOPATS + BIPATS</td>
<td>32.78</td>
</tr>
<tr>
<td>MONOPATS + BIPATS</td>
<td>32.80</td>
</tr>
</tbody>
</table>

Table 7: Comparing ways of ranking patterns from posterior samples. Scores are on MT05 for ZH→EN translation.

6 Conclusion

We have presented generative models for monolingual and bilingual gappy patterns. A qualitative analysis shows that the models discover patterns that match our intuitions in capturing linguistic phenomena. Our experimental results show promise for the ability of these patterns to improve translation for certain language pairs. A key advantage of generative models is the ability to rapidly develop and experiment with variations, especially when using Gibbs sampling for inference. In order to encourage modifications and extensions to these models we have made our source code available at www.ark.cs.cmu.edu/MT.

Acknowledgments

The authors thank Chris Dyer, Qin Gao, Alon Lavie, Nathan Schneider, Stephan Vogel, and the anonymous reviewers for helpful comments. This research was supported in part by the NSF through grant IIS-0844507, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533, and Sandia National Laboratories (fellowship to K. Gimpel).
References


