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). Recently Xiong et al. (2011) used monolingual trigger pairs to improve handling of long-distance dependencies in machine translation output.

All of this previous work used heuristics or local statistical tests to extract patterns from corpora. In this paper, we present probabilistic models that generate text using gappy patterns of arbitrary length and with arbitrarily-many gaps. We exploit nonparametric priors and use Bayesian inference to discover the most salient gappy patterns in monolingual and parallel text. We first inspect these patterns manually and discuss the categories of phenomena that they capture. We also add them as features in a discriminatively-trained phrase-based MT system, using standard techniques to train their weights (Arun and Koehn, 2007; Watanabe et al., 2007) and incorporate them during decoding (Chiang, 2007). We present experiments for Spanish-English and Chinese-English translation, reporting encouraging preliminary results.

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

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 512–522,
Edinburgh, Scotland, UK, July 30–31, 2011.

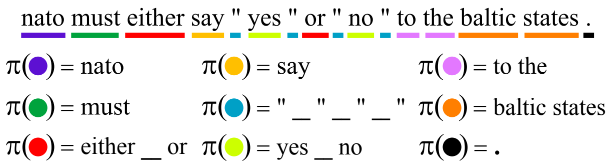

Figure 1: A sentence from the news commentary corpus, along with color assignments for the words and the π function for each color.

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.

3 Monolingual Pattern Models

We first present a model that generates a sentence as a set of lexical items that we will refer to as **gappy patterns**, or simply **patterns**. A pattern is defined as a sequence containing elements of two types: **words** and **gaps**. All patterns must obey the regular expression `w+ (_ w+)*`, where `w` is a word and _ is a gap. That is, patterns must begin and end with words and may not contain consecutive gaps.

We assume that we have an n-word sentence $\boldsymbol{w}_{1:n}$.[1] We represent patterns in a sentence by associating each word with a **color**. To do so, we introduce a vector of color assignment variables $\boldsymbol{c}_{1:n}$, with one for each word. We represent a color C_j as a set in terms of the c_i variables: $C_j = \{i : c_i = j\}$. Each color corresponds to a pattern that is obtained by concatenating its words from left to right in the sentence, inserting gaps when necessary. We denote the pattern for a color C_j by $\pi(C_j)$; Figure 1 shows examples of the correspondence between colors and patterns.

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: $\boldsymbol{w}_{C_j} \sim \text{Mult}(\mu)$. If the words $\boldsymbol{w}_{C_j}$ 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 $\boldsymbol{w}_{1:n}$, color assignments $\boldsymbol{c}_{1:n}$, and number of colors m is

$$p(\boldsymbol{w}_{1:n}, \boldsymbol{c}_{1:n}, m \mid \beta, \mu) = \frac{1}{Z}\left(\frac{\beta^n}{n!}e^{-\beta}\right)\left(\frac{1}{n}\right)\left(\frac{1}{m}\right)^n \prod_{j=1}^{m} p_\mu(\pi(C_j)) \quad (1)$$

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 α 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]

Inference We use collapsed Gibbs sampling for inference. Our goal is to obtain samples from the posterior distribution $p(\{\boldsymbol{c}^{(i)}, m^{(i)}\}_{i=1}^{S} \mid \{\boldsymbol{w}^{(i)}\}_{i=1}^{S}, \nu, \alpha)$, where S is the total number of sentences in the corpus and μ is marginalized out.[3]

[1] We use boldface lowercase letters to denote vectors (e.g., $\boldsymbol{f}$), denote entry i as f_i, and denote the range from i to j as $\boldsymbol{f}_{i:j}$.

[2] The number of ways of arranging y gaps among x words is "$(x-1)$ choose y".

[3] Since we assume the words are given, β 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

$$\frac{\#_{\pi(\{i\})} + \alpha P_0(\pi(\{i\}))}{\# + \alpha} \tag{2}$$

where $\#_\pi$ is the count of pattern π 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

$$\frac{\#_{\pi(C_j \cup \{i\})} + \alpha P_0(\pi(C_j \cup \{i\}))}{\#_{\pi(C_j)} + \alpha P_0(\pi(C_j))} \tag{3}$$

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 (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 $\boldsymbol{c}^{(i)}$ 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.

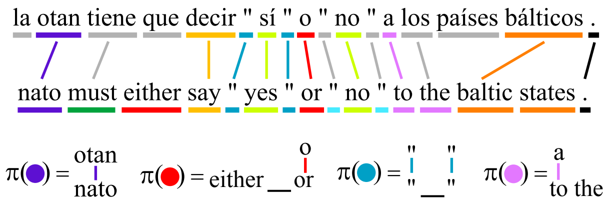
la otan tiene que decir " sí " o " no " a los países bálticos .
nato must either say " yes " or " no " to the baltic states .
π(●) = otan nato
π(●) = either __ o or
π(●) = " " " __ "
π(●) = a to the

Figure 2: A Spanish-English sentence pair with the intersection of automatic word alignments in each direction. Some source words accept the colors of target words aligned to them while others (light gray) do not. Bilingual patterns for a few colors are shown.

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 $\boldsymbol{w}_{1:n}$, we assume we have an n'-word source sentence $\boldsymbol{w}'_{1:n'}$ and word alignments $\boldsymbol{a}_{1:n'}$ where $a_i = j$ iff w'_i is aligned to w_j and $a_i = 0$ if w'_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 $\boldsymbol{g}$ 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 γ 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: $m' \sim \mathrm{Uniform}(1, n'), m \sim \mathrm{Uniform}(1, n)$
3. Sample the alignment vector from any distribution that ensures links are 1-to-1:[4] $\boldsymbol{a}_{1:n'} \sim p(\boldsymbol{a})$
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: $c_i \sim \mathrm{Uniform}(1, m)$. 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: $c'_i \sim \mathrm{Uniform}(1, m')$
6. If any source color has no words, repeat Step 5.
7. For each source color $j = 1 \ldots m'$:
    1. Sample from a multinomial over source patterns: $\boldsymbol{w}_{C'_j} \sim \mathrm{Mult}(\mu')$. While the words $\boldsymbol{w}_{C'_j}$ 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: $\boldsymbol{w}_{C_j} \sim \mathrm{Mult}(\mu)$. While the words $\boldsymbol{w}_{C_j}$ are not consistent with the color assignments, repeat this step.

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

$$p_\gamma(g_i = 1 \mid a_i \neq -1) = \gamma$$
$$p_\gamma(g_i = 1 \mid a_i = -1) = 0$$

where γ determines how frequently source tokens will be added to target patterns.

The probability of generating target words $\boldsymbol{w}_{1:n}$, source words $\boldsymbol{w}'_{1:n'}$, alignments $\boldsymbol{a}_{1:n'}$, target color assignments $\boldsymbol{c}_{1:n}$, source color assignments $\boldsymbol{c}'_{1:n'}$, color propagation variables $\boldsymbol{g}_{1:n'}$, number of target colors m, and number of source colors m' is

$$\begin{aligned}&\frac{1}{Z} p(n)p(n')p(m \mid n)p(m' \mid n')p(\boldsymbol{a}_{1:n'}) \\ &\quad\times \left(\prod_{i=1}^{n} p(c_i \mid m)\right) \\ &\quad\times \left(\prod_{i=1}^{n'} p_\gamma(g_i \mid a_i) p(c'_i \mid m')^{I[g_i == 0]}\right) \\ &\quad\times \left(\prod_{j=1}^{m'} p'_\mu(\pi(C'_j))\right) \left(\prod_{j=1}^{m} p_\mu(\pi(C_j))\right)\end{aligned}$$

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 μ and μ'. They both use DP priors with identical concentration parameters α 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 $\mathrm{Poisson}(1)$ 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 $p(\{\langle \boldsymbol{c}, \boldsymbol{c}', \boldsymbol{g}, m, m' \rangle^{(i)}\}_{i=1}^{S} \mid \{\langle \boldsymbol{w}, \boldsymbol{w}', \boldsymbol{a} \rangle^{(i)}\}_{i=1}^{S})$.

[4] Since we assume alignments are provided during inference, it does not matter what distribution is used, so long as only 1-to-1 links are permitted.

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).

" __ "	as __ as	" __ " __ " __ "
– __ –	the __ of __ in	why __ ?
(__)	the __ is	, __ the __ of
the __ of	not only __ but	from __ to
, __ , __ ,	it is __ that	the __ between __ and
the __ (__)	of __ " __ "	such as __ ,
both __ and	not __ , but	either __ or
the __ of __ and	in __ , __ in	but __ is
more __ than	the __ of __ ,	" __ " __ the
- __ -	what __ ?	has __ been
, __ " __ "	between __ and	in __ , __ ,
the __ " __ "	the __ of __ ’s	an __ of

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.

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 ν and α 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`

academy __ sciences	regulators __ supervisors
beijing __ shanghai	sine __ non
booms __ busts	stalin __ mao
council __ advisers	treasury secretary __ geithner
dominicans __ haitian	sooner __ later
flemish __ walloons	first __ foremost
gref __ program	played __ role
heat __ droughts	down __ road
humanitarian __ displaced	freedom __ expression
karnofsky __ hassenfeld	at __ disposal
kazakhstan __ kyrgyzstan	take __ granted
portugal __ greece	- __ -

Table 2: Gappy patterns with highest conditional probability $p(\pi|\boldsymbol{w}(\pi))$.

– __ –	whether __ or	france __ germany
(__)	around __ world	he __ his
- __ -	has __ been	allow __ to
both __ and	how __ ?	for __ first time
not only __ but	the __ (__)	china __ india
" __ "	on __ basis	what __ do
more __ than	less __ than	we __ our
either __ or	on __ other hand	over __ past
why __ ?	at __ level	prevent __ from
neither __ nor	it is __ that	in __ way
what __ ?	not __ , but	one __ another
rule __ law	play __ role	political __ economic

Table 3: Top-ranked gappy patterns according to $p(\pi)p(\pi|\boldsymbol{w}(\pi))$.

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:

$$p(\pi|\boldsymbol{w}(\pi)) = \frac{\#_\pi}{\#_{\boldsymbol{w}(\pi)}}$$

where $\boldsymbol{w}(\pi)$ returns the sequence of words in the pattern π and $\#_{\boldsymbol{w}(\pi)}$ is the number of occurrences of this sequence of words in the corpus that are compatible with pattern π. 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.

	Rank	Gappy Lexical Patterns	Rank	Gappy Brown Cluster Patterns
Punctuation	1	-- __ --	2	{what, why, whom, whatever} __ {?, !}
	2	(__)	6	{--, -, –} __ {--, -, –}
	6	" __ "	28	{according, compared, subscribe, thanks, referring} to __ ,
	9	why __ ?	178	{–, -, –} {even, especially, particularly, mostly, mainly} __ {–, -, –}
	63	according to __ ,	239	{obama, bush, clinton, mccain, brown} __ " __ "
Agreement	26	he __ his	8	{people, things, americans, journalists, europeans} __ their
	31	we __ our	12	we __ {our, my}
	46	his __ his	21	{children, women, others, men, students} __ their
	86	china __ its	23	{china, europe, america, russia, iran} ’s __ its
	90	his __ he	43	{obama, bush, clinton, mccain, brown} __ his
	99	you __ your	46	{our, my} __ {our, my}
	136	leaders __ their	149	{people, things, americans, journalists, europeans} __ they
	140	we __ ourselves	172	{president, bill, sen., king, senator} {obama, bush, clinton, mccain, brown} __ his
	165	these __ are	180	{all, both, either} __ {countries, companies, banks, groups, issues}
Connectives	4	both __ and	5	{more, less} __ {more, less}
	5	not only __ but	9	if __ , __ {will, would, could, should, might}
	8	either __ or	19	{deal, plan, vote, decision, talks} {against, between, involving} __ and
	10	neither __ nor	40	a __ {against, between, involving} __ and
	13	whether __ or	45	{better, different, further, higher, lower} __ than
	19	less __ than	50	{much, far, slightly, significantly, substantially} __ than
	23	not __ , but	56	{yet, instead, perhaps, thus, neither} __ but
	54	if __ then	68	not {only, necessarily} __ {also, hardly}
	109	between __ and	98	as {much, far, slightly, significantly, substantially} __ as
	192	relationship between __ and	131	is __ {more, less} __ than
Topicality	25	france __ germany	1	⟨UNK⟩ __ ⟨UNK⟩
	29	china __ india	15	{china, europe, . . . } ’s __ {system, crisis, program, recession, situation}
	36	political __ economic	30	{health, security, defense, safety, intelligence} __ {health, . . . }
	43	rich __ poor	47	{china, europe, . . . } __ {china, europe, . . . } __ {china, europe, . . . }
	50	oil __ gas	62	{power, growth, interest, development} __ {10, 1, 20, 30, 2} {percent, %, p.m., a.m.}
	62	billions __ dollars	72	in {iraq, washington, london, 2008, 2009} __ {iraq, washington, london, 2008, 2009}
	96	economic __ social	73	the {end, cost, head, rules, average} of __ {prices, markets, services, problems, costs}
	106	the us __ europe	113	{china, europe, . . . } ’s __ {economy, election, elections, population, investigation}
	181	public __ private	119	{prices, markets, . . . } __ {oil, energy, tax, food, investment} __ {oil, energy, . . . }
Prepositions	14	around __ world	14	for __ {first, second, third, final, whole} {time, period, term, class, avenue}
	18	on __ basis	17	in __ {last, next, 20th} {year, week, month, season, summer}
	38	at __ time	51	at __ {end, cost, head, rules, average} of
	42	in __ region	71	at __ {group, rate, leader, level, manager}
	80	in __ manner	112	for __ {times, points, games, goals, reasons}
	85	at __ expense	126	{over, around, across, behind, above} __ {country, company, region, nation, virus}
	112	during __ period	190	{one, none} of __ {best, top, largest, main, biggest}
Constructions	33	prevent __ from		
	84	enable __ to		
	114	provide __ for		
	123	impose __ on		
	177	turn __ into		

Non-Gappy Lexical Patterns		Non-Gappy Brown Cluster Patterns	
as well	their own	as {well, soon, quickly, seriously, slowly} as	{rather, please} than
the united states	prime minister	the united {states, nations, airlines}	{don, didn, doesn, isn, wasn} ’t
have been	climate change	{president, bill, sen., king, senator} {mr., mr, john, david, michael} {obama, bush, clinton, . . . }	
rather than	the bush administration	{order, plans, needs, efforts, failed} to {make, take, give, keep, provide}	
based on	developing countries	{will, would, could, should, might} not be	{can, ’ll} be

Table 4: Gappy patterns manually divided into categories of long-distance dependencies. Patterns were ranked according to $p(\pi)p(\pi|\boldsymbol{w}(\pi))$ 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 commonly 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 ($\boldsymbol{\lambda}$) from weights for our pattern features ($\boldsymbol{\theta}$). We use $\boldsymbol{h}(e)$ to denote the feature vector for translation e. The function $B_i(t)$ returns the sentence BLEU score for translation t given reference e_i (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 $\boldsymbol{\lambda}_M$ (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

[7] `www.statmt.org/wmt10/baseline.html`.

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

```
Input: input sentences $F = \{f_i\}_{i=1}^N$, references
       $E = \{e_i\}_{i=1}^N$, initial weights $\boldsymbol{\lambda}_0$, size of
       $k$-best list $k$, MIRA max step size $C$, num.
       iterations $T$
Output: learned weights: $\boldsymbol{\lambda}_M$, $\langle \boldsymbol{\lambda}^*, \boldsymbol{\theta}^* \rangle$
$\boldsymbol{\lambda}_M \leftarrow$ MERT ($F$, $E$, $\boldsymbol{\lambda}_0$);
$\{\ell_i\}_{i=1}^N \leftarrow$ generateLattices ($F$, $\boldsymbol{\lambda}_M$);
$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}_M$; $\boldsymbol{\theta} \leftarrow \mathbf{0}$;
$\langle \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}} \rangle \leftarrow \langle \boldsymbol{\lambda}, \boldsymbol{\theta} \rangle$;
for $iter \leftarrow 1$ to $T$ do
    for $i \leftarrow 1$ to $N$ do
        $\{t_j\}_{j=1}^k \leftarrow$ Decode($\ell_i$, $\langle \boldsymbol{\lambda}, \boldsymbol{\theta} \rangle$);
        $e^+ \leftarrow \mathrm{argmax}_{1 \leq j \leq k} B_i(t_j)$;
        $e^- \leftarrow \mathrm{argmin}_{1 \leq j \leq k} B_i(t_j)$;
        $\Delta \leftarrow \max(0, \langle \boldsymbol{\lambda}, \boldsymbol{\theta} \rangle^\top [\boldsymbol{h}(e^-) - \boldsymbol{h}(e^+)]$
                $+ B_i(e^+) - B_i(e^-))$;
        $\eta \leftarrow \min(C, \frac{\Delta}{\|\boldsymbol{h}(e^+) - \boldsymbol{h}(e^-)\|^2})$;
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta [\boldsymbol{h}(e^+) - \boldsymbol{h}(e^-)]$;
        $\langle \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}} \rangle \leftarrow \langle \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}} \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{\theta} \rangle$;
    end
end
$\langle \boldsymbol{\lambda}^*, \boldsymbol{\theta}^* \rangle \leftarrow \langle \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\theta}} \rangle \times \frac{1}{T \times N + 1}$;
return $\boldsymbol{\lambda}_M$, $\langle \boldsymbol{\lambda}^*, \boldsymbol{\theta}^* \rangle$;
```

Algorithm 1: Train

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 current iteration.

At decoding time, we follow a procedure similar to training: we generate lattices for each source sentence using Moses with its standard set of features and using weights $\boldsymbol{\lambda}_M$. We rescore the lattices using $\boldsymbol{\lambda}^*$ and use cube pruning (Chiang, 2007; Huang and Chiang, 2007) to incorporate the gappy pattern features with weights $\boldsymbol{\theta}^*$. 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 initial weights $\boldsymbol{\lambda}_0$, we used the default Moses feature weights. We used $k = 100$, $C = 0.0001$, and $T = 15$. For the n-best list size used during cube pruning during both training and decoding, we used $n = 100$. There are several Moses parameters that affect the scope of the search during decoding and therefore the size of the phrase lattices. We used default values for these except for the stack size parameter, for which we used 100. The resulting lattices encode up to 10^{50} derivations for ES→EN and 10^{65} derivations for ZH→EN.

	ES→EN	ZH→EN
MERT	25.64	32.47
Alg. 1	25.85	32.33

Table 5: Comparing MERT to our training procedure. All numbers are %BLEU.

Table 5 shows test set %BLEU for each language pair and training algorithm. Our procedure performs comparably to MERT. Therefore we use it as our baseline for subsequent experiments since it can handle a large number of feature weights; this allows us to observe the contribution of the additional gappy pattern features more clearly.

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 patterns 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 features (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,

	ES→EN	ZH→EN
Baseline	25.85	32.33
MonoPats	25.84	32.81
BiPats	25.92	32.68
MonoPats + BiPats	25.59	32.80

Table 6: Adding gappy pattern features. All numbers are %BLEU.

	Ranking	%BLEU
Baseline	N/A	32.33
MonoPats	$p(\pi)$	32.65
MonoPats	$p(\pi \vert \boldsymbol{w}(\pi))$	32.53
MonoPats	$p(\pi)p(\pi \vert \boldsymbol{w}(\pi))$	32.81
BiPats	$p(\pi)$	32.68
MonoPats + BiPats	$p(\pi)$	32.78
MonoPats + BiPats	$p(\pi)p(\pi \vert \boldsymbol{w}(\pi))$	32.80

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

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 $p(\pi)$. The results show that ranking monolingual patterns using the product-of-experts method results in the highest BLEU scores, validating our intuitions from observing Tables 1-3. Table 8 shows the most highly-weighted pattern features for the best ZH→EN model.

said that __ the	however , __ the	agence france __ presse
's __ , __ 's	us __ iraq	reported __ the
of __ million	, __ likely	said that __ and
added __ "	- __ -	rate __ percent
the __ {media, school, university, election, bank} __ {made, established, given, taken, reached}		
{said, stressed, stated, indicated, noted} that __ in		
{meeting, report, conference, reports} __ {1, july, june, march, april}		
{news, press, spokesman, reporter} {meeting, . . . } __ {1, july, . . . }		
{news, press, spokesman, reporter} __ {1, july, june, march, april}		
the __ {enterprises, companies, students, customers, others} __ {enterprises, companies, students, customers, others}		
{japan, russia, europe, 2003, 2004} __ {us, japanese, russian, u.s.}		

Table 8: Selected features from the 15 most highly-weighted lexical and cluster pattern features in the best ZH→EN model.

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

A. Arun and P. Koehn. 2007. Online learning methods for discriminative training of phrase based statistical machine translation. In *Proc. of MT Summit XI*.

M. Bansal, C. Quirk, and R. Moore. 2011. Gappy phrasal alignment by agreement. In *Proc. of ACL*.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. 1992. Class-based N-gram models of natural language. *Computational Linguistics*, 18.

C. Callison-Burch, P. Koehn, C. Monz, K. Peterson, M. Przybocki, and O. Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proc. of the 5th Workshop on Statistical Machine Translation*.

P. Chang, M. Galley, and C. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proc. of the Third Workshop on Statistical Machine Translation*.

S. Chen and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report 10-98, Harvard University.

D. Chiang, Y. Marton, and P. Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proc. of EMNLP*.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL*.

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

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

J. M. Crego and F. Yvon. 2009. Gappy translation units under left-to-right SMT decoding. In *Proc. of EAMT*.

M. Galley and C. D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Proc. of NAACL*.

L. Huang and D. Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. of ACL*.

F. Jelinek. 1997. *Statistical methods for speech recognition*. MIT Press.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL (demo session)*.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. of COLING-ACL*.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

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

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of ACL*.

K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10(3).

M. Simard, N. Cancedda, B. Cavestro, M. Dymetman, É. Gaussier, C. Goutte, K. Yamada, P. Langlais, and A. Mauser. 2005. Translating with non-contiguous phrases. In *Proc. of HLT-EMNLP*.

A. Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Proc. of ICSLP*.

N. Ueffing, F. J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. of EMNLP*.

T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proc. of EMNLP-CoNLL*.

D. Xiong, M. Zhang, and H. Li. 2011. Enhancing language models in statistical machine translation with backward N-grams and mutual information triggers. In *Proc. of ACL*.