Factored Markov Translation with Robust Modeling

Yang Feng[†] **Trevor Cohn**[‡] **Xinkai Du**

[†] Information Sciences Institue
Computer Science Department
University of Southern California

[‡] Computing and Information Systems
The University of Melbourne
VIC 3010 Australia

{yangfeng145, xinkaid}@gmail.com t.cohn@unimelb.edu.au

Abstract

Phrase-based translation models usually memorize local translation literally and make independent assumption between phrases which makes it neither generalize well on unseen data nor model sentence-level effects between phrases. In this paper we present a new method to model correlations between phrases as a Markov model and meanwhile employ a robust smoothing strategy to provide better generalization. This method defines a recursive estimation process and backs off in parallel paths to infer richer structures. Our evaluation shows an 1.1–3.2% BLEU improvement over competitive baselines for Chinese-English and Arabic-English translation.

1 Introduction

Phrase-based methods to machine translation (Koehn et al., 2003; Koehn et al., 2007) have drastically improved beyond word-based approaches, primarily by using phrase-pairs as translation units, which can memorize local lexical context and reordering patterns. However, this literal memorization mechanism makes it generalize poorly to unseen data. Moreover, phrase-based models make an independent assumption, stating that the application of phrases in a derivation is independent to each other which conflicts with the underlying truth that the translation decisions of phrases should be dependent on context.

There are some work aiming to solve the two problems. Feng and Cohn (2013) propose a word-based Markov model to integrate translation and reordering into one model and use the sophisticated hierarchical Pitman-Yor process which backs off from larger to smaller context to provide dynamic adaptive smoothing. This model shows good generalization to unseen data while it uses words as the translation unit which cannot handle multiple-to-multiple links in real word alignments. Durrani et al. (2011) and Durrani et al. (2013) propose an operation sequence model (OSM) which models correlations between minimal translation units (MTUs) and evaluates probabilities with modified Kneser-Ney smoothing. On one hand the use of MTUs can help retain the multiple-to-multiple alignments, on the other hand its definition of operations where source words and target words are bundled into one operation makes it subjected to sparsity. The common feature of the above two methods is they both back off in one fixed path by dropping least recent events first which precludes some useful structures. For the segment pairs <bǎ tā kǎolǜ jìnqù, take it into account> in Figure 1, the more common structure is <bǎ ... kǎolǜ jìnqù, take ... into account>. If we always drop the least recent events first, then we can only learn the pattern <... tā kǎolǜ jìnqù, ... it into account>.

On these grounds, we propose a method with new definition of correlations and more robust probability modeling. This method defines a Markov model over correlations between minimal phrases where each is decomposed into three factors (*source, target* and *jump*). In the meantime it employs a fancier smoothing strategy for the Markov model which backs off by dropping multiple conditioning factors in parallel in order to learn richer structures. Both the uses of factors and parallel backoff give rise to robust modeling against sparsity. In addition, modeling bilingual information and reorderings into one model instead of adding them to the linear model as separate features allows for using more sophisticated estimation methods rather than get a loose weight for each feature from tuning algorithms.

We compare the performance of our model with that of the phrase-based model and the hierarchical phrase-based model on the Chinese-English and Arabic-English NIST test sets, and get an im-

Proceedings of the Eighteenth Conference on Computational Language Learning, pages 151–159,
Baltimore, Maryland USA, June 26-27 2014. ©2014 Association for Computational Linguistics

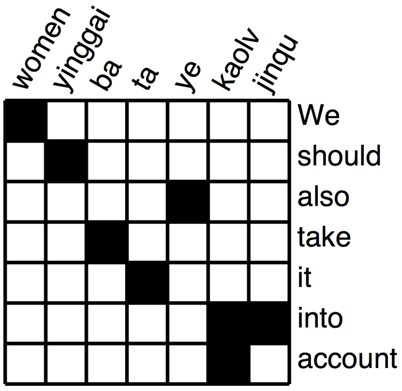


Figure 1: Example Chinese-English sentence pair with word alignments shown as filled grid squares.

provement up to 3.2 BLEU points absolute.[1]

2 Modelling

Our model is phrase-based and works like a phrase-based decoder by generating target translation left to right using phrase-pairs while jumping around the source sentence. For each derivation, we can easily get its minimal phrase (*MPs*) sequence where MPs are ordered according to the order of their target side. Then this sequence of events is modeled as a Markov model and the log probability under this Markov model is included as an additional feature into the linear SMT model (Och, 2003).

A MP denotes a phrase which cannot contain other phrases. For example, in the sentence pair in Figure 1, <*bǎ tā* , take it> is a phrase but not a minimal phrase, as it contains smaller phrases of <*bǎ* , take> and <*tā* , it>. MPs are a complex event representation for sequence modelling, and using these naively would be a poor choice because few bigrams and trigrams will be seen often enough for reliable estimation. In order to reason more effectively from sparse data, we consider more generalized representations by decomposing MPs into their component events: the source phrase (*source* $\bar{f}$), the target phrase (*target* $\bar{e}$) and the jump distance from the preceding MP (*jump* j), where the jump distance is counted in MPs, not in words. For sparsity reasons, we do not use the jump distance directly but instead group it into 12 buckets:

$\{\text{insert}, \leq -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, \geq 5\}$,

where the jump factor is denoted as insert when the source side is NULL. For the sentence pair in Figure 1, the MP sequence is shown in Figure 2.

To evaluate the Markov model, we condition each MP on the previous $k-1$ MPs and model each of the three factors separately based on a chain rule decomposition. Given a source sentence $\boldsymbol{f}$ and a target translation $\boldsymbol{e}$, the joint probability is defined as

$$\begin{aligned} p(\bar{\boldsymbol{e}}_1^I, \boldsymbol{j}_1^I, \bar{\boldsymbol{f}}_1^I) = & \prod_{i=1}^{I} p(\bar{e}_i | \bar{f}_{i-k+1}^{i}, j_{i-k+1}^{i}, \bar{e}_{i-k+1}^{i-1}) \\ & \times \prod_{i=1}^{I} p(\bar{f}_i | \bar{f}_{i-k+1}^{i-1}, j_{i-k+1}^{i}, \bar{e}_{i-k+1}^{i-1}) \\ & \times \prod_{i=1}^{I} p(j_i | \bar{f}_{i-k+1}^{i-1}, j_{i-k+1}^{i-1}, \bar{e}_{i-k+1}^{i-1}) \end{aligned} \quad (1)$$

where $\bar{f}_i$, $\bar{e}_i$ and j_i are the factors of MP$_i$, $\bar{\boldsymbol{f}}_1^I = (\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_I)$ is the sequence of source MPs, $\bar{\boldsymbol{e}}_1^I = (\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_I)$ is the sequence of target MPs, and $\boldsymbol{j}_1^I = (j_1, j_2, \ldots, j_I)$ is the vector of jump distance between MP$_{i-1}$ and MP$_i$, or `insert` for MPs with null source sides.[2] To evaluate each of the k-gram models, we use modified Keneser-Ney smoothing to back off from larger context to smaller context recursively.

In summary, adding the Markov model into the decoder involves two passes: 1) training a model over the MP sequences extracted from a word aligned parallel corpus; and 2) calculating the probability of the Markov model for each translation hypothesis during decoding. This Markov model is combined with a standard phrase-based model[3] (Koehn et al., 2007) and used as an additional feature in the linear model.

In what follows, we will describe how to estatimate the k-gram Markov model, focusing on backoff (§2.1) and smoothing (§2.2).

2.1 Parallel Backoff

Backoff is a technique used in language model — when estimating a higher-order gram, instead of using the raw occurrence count, only a portion is used and the remainder is computed using a lower-order model in which one of the context factors

[1] We will contribute the code to Moses.

[2] Note that factors at indices $0, -1, \ldots, -(k-1)$ are set to a sentinel value to denote the start of sentence.

[3] The phrase-based model considers larger phrase-pairs than just MPs, while our Markov model consider only MPs. As each phrase-pair is composed of a sequence of MPs under fixed word alignment, by keeping the word alignment for each phrase, a decoder derivation unambiguously specifies the MP sequence for scoring under our Markov model.

index	*sentence pair*						*minimal phrase sequence*			
	wǒmén	yīnggāi	bǎ	tā	yě	kǎolv̀ jìnqù		*jump*	*source*	*target*
1	We						T_1	1	wǒmén	We
2		should					T_2	1	yīnggāi	should
3					also		T_3	3	yě	also
4			take				T_4	-2	bǎ	take
5				it			T_5	1	tā	it
6						into account	T_6	2	kǎolv̀ jìnqù	into account

Figure 2: The minimal phrase sequence $T_1, ..., T_6$ extracted from the sentence pair in Figure 1.

step	3-gram $\bar{e}_3 \mid \bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, j_2, \bar{e}_1, \bar{f}_1, j_1$
0	into account | kǎolv̀ jìnqù, 2, it, tā, 1, take, bǎ, -2
	↓ 1
1	into account | kǎolv̀ jìnqù, 2, it, tā, –, take, bǎ, -2
	↓ tā
2	into account | kǎolv̀ jìnqù, 2, it, –, –, take, bǎ, -2
	↓ it
3	into account | kǎolv̀ jìnqù, 2, –, –, –, take, bǎ, -2
	↓ -2
4	into account | kǎolv̀ jìnqù, 2, –, –, –, take, bǎ, –
	↓ bǎ
5	into account | kǎolv̀ jìnqù, 2, –, –, –, take, –, –
	↓ take
6	into account | kǎolv̀ jìnqù, 2, –, –, –, –, –, –
	↓ 2
7	into account | kǎolv̀ jìnqù, –, –, –, –, –, –, –
	↓ kǎolv̀ jìnqù
8	into account | –, –, –, –, –, –, –, –

Figure 3: One backoff path for the 3-gram in Equation 2. The symbols besides each arrow mean the current factor to drop; "–" is a placeholder for factors which can take any value.

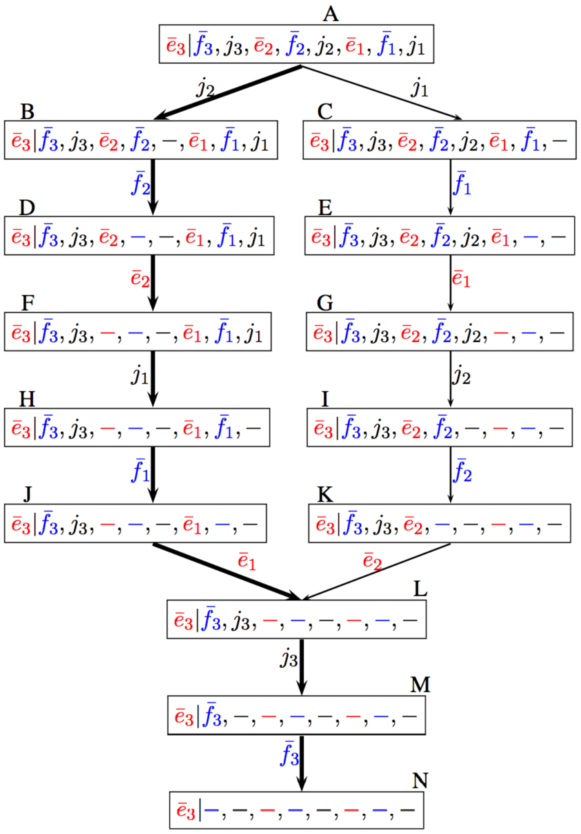

A
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, j_2, \bar{e}_1, \bar{f}_1, j_1$
j_2
j_1
B
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, -, \bar{e}_1, \bar{f}_1, j_1$
C
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, j_2, \bar{e}_1, \bar{f}_1, -$
$\bar{f}_2$
$\bar{f}_1$
D
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, -, -, \bar{e}_1, \bar{f}_1, j_1$
E
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, j_2, \bar{e}_1, -, -$
$\bar{e}_2$
$\bar{e}_1$
F
$\bar{e}_3|\bar{f}_3, j_3, -, -, -, \bar{e}_1, \bar{f}_1, j_1$
G
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, j_2, -, -, -$
j_1
j_2
H
$\bar{e}_3|\bar{f}_3, j_3, -, -, -, \bar{e}_1, \bar{f}_1, -$
I
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, \bar{f}_2, -, -, -, -$
$\bar{f}_1$
$\bar{f}_2$
J
$\bar{e}_3|\bar{f}_3, j_3, -, -, -, \bar{e}_1, -, -$
K
$\bar{e}_3|\bar{f}_3, j_3, \bar{e}_2, -, -, -, -, -$
$\bar{e}_1$
$\bar{e}_2$
L
$\bar{e}_3|\bar{f}_3, j_3, -, -, -, -, -, -$
j_3
M
$\bar{e}_3|\bar{f}_3, -, -, -, -, -, -, -$
$\bar{f}_3$
N
$\bar{e}_3|-, -, -, -, -, -, -, -$

Figure 4: The backoff graph for the 3-gram model of the target factor. The symbol beside each arrow is the factor to drop.

is dropped. Here the probabilities of the lower-order which is used to construct the higher-order is called the *backoff probability* of the higher-order gram. Different from standard language models which drop the least recent words first, we employ a different backoff strategy which considers all possible backoff paths. Taking as an example the 3-gram $T_4T_5T_6$ in Figure 2, when estimating the probability of the target factor

$$\text{p(into account} \mid \text{kǎolv̀ jìnqù, 2, it, tā, 1, take, bǎ, -2)} \;, \tag{2}$$

we consider two backoff paths: path$_1$ drops the factors in the order *-2*, *bǎ*, *take*, *1*, *tā*, *it*, *2*, *kǎolv̀ jìnqù*; path$_2$ uses order *1*, *tā*, *it*, *-2*, *bǎ*, *take*, *2*, *kǎolv̀ jìnqù*. Figure 3 shows the backoff process for path$_2$. In this example with two backoff paths, the backoff probability g is estimated as

$$g(\text{into acc.}|\boldsymbol{c}) = \frac{1}{2}p(\text{into acc.}|\boldsymbol{c}') + \frac{1}{2}p(\text{into acc.}|\boldsymbol{c}'') \,,$$

where $\boldsymbol{c} =<$ kǎolv̀ jìnqù, 2, it, tā, 1, take, bǎ, -2 $>$, $\boldsymbol{c}' =<$ kǎolv̀ jìnqù, 2, it, tā, 1, take, bǎ, – $>$ and $\boldsymbol{c}'' =<$ kǎolv̀ jìnqù, 2, it, tā, –, take, bǎ, -2 $>$.

Formally, we use the notion of backoff graph to define the recursive backoff process of a k-gram and denote as nodes the k-gram and the lower-order grams generated by the backoff. Once one node occurs in the training data fewer than τ times, then estimates are calculated by backing off to the nodes in the next lower level where one factor is dropped (denoted using the placeholder – in Figure 4). One node can have one or several candidate backoff nodes. In the latter case, the backoff probability is defined as the *average* of the probabilities of the backoff nodes in the next lower level.

We define the backoff process for the 3-gram model predicting the target factor, $\bar{e}_3$, as illustrated in Figure 4. The top level is the full 3-gram, from which we derive two backoff paths by dropping factors from contextual events, one at a time. Formally, the backoff strategy is to drop the previous two MPs one by one while for each MP the dropping routine is first the jump factor, then the source factor and final the target factor. Each step on the path corresponds to dropping an individual contextual factor from the context. The paths converge when only the third MP left, then the backoff proceeds by dropping the jump action, j_3, then finally the source phrase, $\bar{f}_3$. The paths B-D-F-H-J and C-E-G-I-K show all the possible orderings (corresponding to $\boldsymbol{c}''$ and $\boldsymbol{c}'$, respectively) for dropping the two previous MPs. The example backoff in Figure 3 corresponds the path A-B-D-F-H-J-L-M-N in Figure 4, shown as heavier lines. When generizing to the k-gram for target $p(\bar{e}_k|\bar{f}_1^k, j_1^k, \bar{e}_1^{k-1})$, the backoff strategy is to first drop the previous k-1 MPs one by one (for each MP, still drops in the order of jump, source and target), then the *kth* jump factor and finally the *kth* source factor. According to the strategy, the top node has k-1 nodes to back off to and for the node $\bar{e}_k|\bar{f}_2^k, j_2^k, \bar{e}_2^{k-1}$ where only the factors of MP$_1$ are dropped, there are k-2 nodes to back off to.

2.2 Probability Estimation

We adopt the technique used in factor language models (Bilmes and Kirchhoff, 2003; Kirchhoff et al., 2007) to estimate the probability of a k-gram $p(\bar{e}_i|\boldsymbol{c})$ where $\boldsymbol{c} = \bar{f}_{i-k+1}^i, j_{i-k+1}^i, \bar{e}_{i-k+1}^{-1}$. According to the definition of backoff, only when the count of the k-gram exceeds some given threshold, its maximum-likelihood estimate, $p_{ML}(\bar{e}_k|\boldsymbol{c}) = \frac{N(\bar{e}_k,\boldsymbol{c})}{N(\boldsymbol{c})}$ is used, where $N(\cdot)$ is the count of an event and/or context. Otherwise, only a portion of $p_{ML}(\bar{e}_k|\boldsymbol{c})$ is used and the remainder is constructed from a lower-level (by dropping a factor). In order to ensure valid probability estimates, i.e. sums to unity, probability mass needs to be "stolen" from the higher level and given to the lower level. Hence, the whole definition is

$$p(\bar{e}_i|\boldsymbol{c}) = \begin{cases} d_{N(\bar{e}_i,\boldsymbol{c})} p_{ml}(\bar{e}_i|\boldsymbol{c}) & \text{if } N(\bar{e}_i,\boldsymbol{c}) > \tau_k \\ \alpha(\boldsymbol{c}) g(\bar{e}_i,\boldsymbol{c}) & \text{otherwise} \end{cases} \tag{3}$$

where $d_{N(\bar{e}_i,\boldsymbol{c})}$ is a discount parameter which reserves probability from the maximum-likelihood estimate for backoff smoothing at the next lower-level, and we estimate $d_{N(\bar{e}_i,\boldsymbol{c})}$ using modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1996); τ_k is the threshold for the count of the k-gram, $\alpha(\boldsymbol{c})$ is the backoff weight used to make sure the entire distribution still sums to unity,

$$\alpha(\boldsymbol{c}) = \frac{1 - \sum_{\bar{e}:N(\bar{e},\boldsymbol{c})>\tau_k} d_{N(\bar{e},\boldsymbol{c})} p_{ML}(\bar{e}|\boldsymbol{c})}{\sum_{\bar{e}:N(\bar{e},\boldsymbol{c})\leq\tau_k} g(\bar{e},\boldsymbol{c})},$$

and $g(\bar{e}_i,\boldsymbol{c})$ is the backoff probability which we estimate by averaging over the nodes in the next lower level,

$$g(\bar{e}_i,\boldsymbol{c}) = \frac{1}{\phi} \sum_{\boldsymbol{c}'} p(\bar{e}_i|\boldsymbol{c}'),$$

where ϕ is the number of nodes to back off, $\boldsymbol{c}'$ is the lower-level context after dropping one factor from $\boldsymbol{c}$.

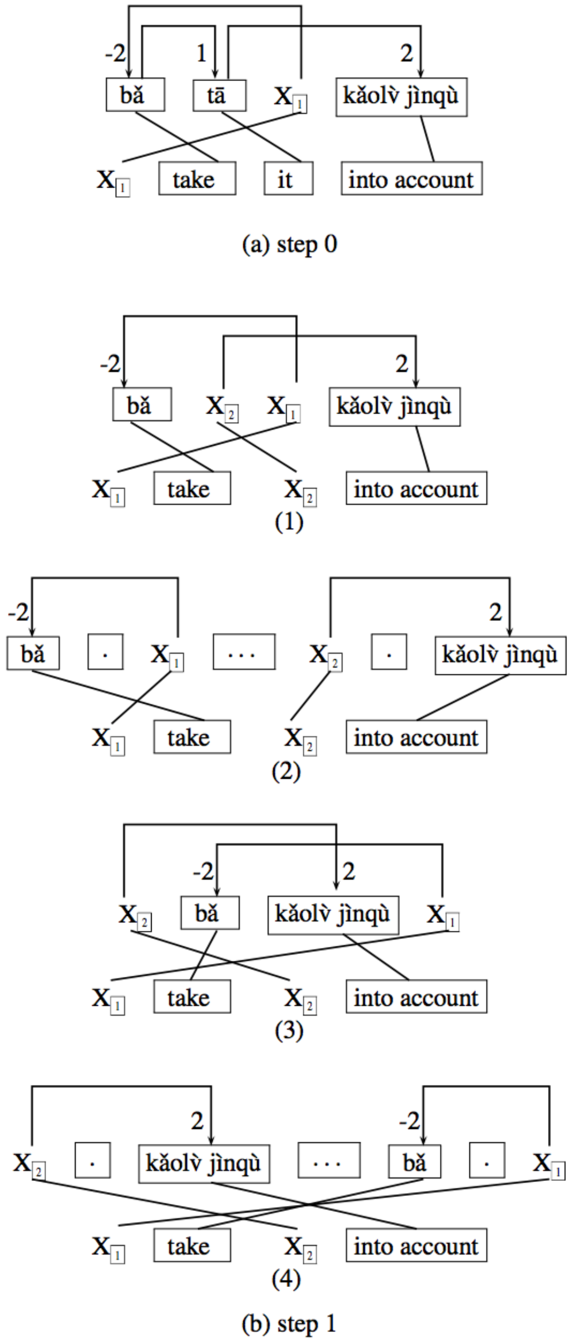
The k-gram for the source and jump factors are estimated in the same way, using the same backoff semantics.[4] Note (3) is applied independently to each of the three models, so the use of backoff may differ in each case.

3 Discussion

As a part of the backoff process our method can introduce gaps in estimating rule probabilities; these backoff patterns often bear close resemblance to SCFG productions in the hierarchical phrase-based model (Chiang, 2007). For example, in step 0 in Figure 3, as all the jump factors are present, this encodes the full ordering of the MPs and gives rise to the aligned MP pairs shown in Figure 5 (a). Note that an $\mathrm{X}_{\boxed{1}}$ placeholder is included to ensure the jump distance from the previous MP to the MP <*bǎ, take*> is *-2*. The approximate SCFG production for the MP pairs is

<bǎ tā $\mathrm{X}_{\boxed{1}}$ kǎolǜ jìnqù, $\mathrm{X}_{\boxed{1}}$ take it into account>.

[4] Although there are fewer final steps, L-M-N in Fig. 4, as we assume the MP is generated in the order jump, source phrase then target phrase in a chain rule decomposition.



(a) step 0

(1)

(2)

(3)

(4)

(b) step 1

Figure 5: Approximate SCFG patterns for step 0, 3 of Figure 3. X is a non-terminal which can only be rewritten by one MP. $\boxed{\cdot}$ and $\boxed{\cdots}$ denote gaps introduced by the left-to-right decoding algorithm and $\boxed{\cdot}$ can only cover one MP while $\boxed{\cdots}$ can cover zero or more MPs.

In step 1, as the jump factor *1* is dropped, we do not know the orientation between bă and tā. However several jump distances are known: from $X_{\boxed{1}}$ to *bă* is distance *-2* and *tā* to *kǎolv̀ jìnqù* is 2. In this case, the source side can be

bă tā $X_{\boxed{1}}$ *kǎolv̀ jìnqù*,

bă $\boxed{\cdot}$ $X_{\boxed{1}}$ $\boxed{\cdots}$ *tā* $\boxed{\cdot}$ *kǎolv̀ jìnqù*,

tā bă kǎolv̀ jìnqù $X_{\boxed{1}}$,

tā $\boxed{\cdot}$ *kǎolv̀ jìnqù* $\boxed{\cdots}$ bă $\boxed{\cdot}$ $X_{\boxed{1}}$,

where X and $\boxed{\cdot}$ can only hold one MP while $\boxed{\cdots}$ can cover zero or more MPs. In step 3 after dropping *tā* and *it*, we introduce a gap $X_{\boxed{2}}$ as shown in Figure 5 (b).

From above, we can see that our model has two kinds of gaps: 1) in the source due to the left-to-right target ordering (such as the $\boxed{\cdot}$ in step 3); and 2) in the target, arising from backoff (such as the $X_{\boxed{2}}$ in step 3). Accordingly our model supports rules than cannot be represented by a 2-SCFG (e.g., step 3 in Figure 5 requires a 4-SCFG). In contrast, the hierarchical phrase-based model allows only 2-SCFG as each production can rewrite as a maximum of two nonterminals. On the other hand, our approach does not enforce a valid hierarchically nested derivation which is the case for Chiang's approach.

4 Related Work

The method introduced in this paper uses factors defined in the same manner as in Feng and Cohn (2013), but the two methods are quite different. That method (Feng and Cohn, 2013) is word-based and under the frame of Bayesian model while this method is MP-based and uses a simpler Kneser-Ney smoothing method. Durrani et al. (2013) also present a Markov model based on MPs (they call minimal translation units) and further define operation sequence over MPs which are taken as the events in the Markov model. For the probability estimation, they use Kneser-Ney smoothing with a single backoff path. Different from operation sequence, our method gives a neat definition of factors which uses jump distance directly and avoids the bundle of source words and target words like in their method, and hence mitigates sparsity. Moreover, the use of parallel backoff infers richer structures and provides robust modeling.

There are several other work focusing on modeling bilingual information into a Markov model. Crego et al. (2011) develop a bilingual language model which incorporates words in the source and target languages to predict the next unit, and use it as a feature in a translation system. This line of work was extended by Le et al. (2012) who develop a novel estimation algorithm based around discriminative projection into continuous spaces. Neither work includes the jump distance, and nor do they consider dynamic strategies for estimating k-gram probabilities.

Galley and Manning (2010) propose a method to introduce discontinuous phrases into the phrase-based model. It makes use of the decoding mechanism of the phrase-based model which jumps over the source words and hence can hold discontinuous phrases naturally. However, their method doesn't touch the correlations between phrases and probability modeling which are the key points we focus on.

5 Experiments

We design experiments to first compare our method with the phrase-based model (PB), the operation sequence model (OSM) and the hierarchical phrase-based model (HPB), then we present several experiments to test:

1. how each of the factors in our model and parallel backoff affect overall performance;
2. how the language model order affects the relative gains, in order to test if we are just learning a high order LM, or something more useful;
3. how the Markov model interplay with the distortion and lexical reordering models of Moses, and are they complemenatary;
4. whether using MPs as translation units is better in our approach than the simpler tactic of using only word pairs.

5.1 Data Setup

We consider two language pairs: Chinese-English and Arabic-English. The Chinese-English parallel training data is made up of the non-UN portions and non-HK Hansards portions of the NIST training corpora, distributed by the LDC, having 1,658k sentence pairs with 40m and 44m Chinese and English words. We used the NIST 02 test set as the development set and evaluated performance on the test sets from NIST 03 and 05.

For the Arabic-English task, the training data comprises several LDC corpora,[5] including 276k sentence pairs and 8.21m and 8.97m words in Arabic and English, respectively. We evaluated on the NIST test sets from 2003 and 2005, and the NIST 02 test set was used for parameter tuning.

On both cases, we used the factor language model module (Kirchhoff et al., 2007) of the SRILM toolkit (Stolcke, 2002) to train a Markov model with the order $= 3$ over the MP sequences.[6] The threshold count of backoff for all nodes was $\tau = 2$.

We aligned the training data sets by first using GIZA++ toolkit (Och and Ney, 2003) to produce word alignments on both directions and then combining them with the *diag-final-and* heuristic. All experiments used a 5-gram language model which was trained on the Xinhua portion of the GIGAWORD corpus using the SRILM toolkit. Translation performance was evaluated using BLEU (Papineni et al., 2002) with case-insensitive $n \leq 4$-grams. We used minimum error rate training (Och, 2003) to tune the feature weights to maximize the BLEU score on the development set.

We used *Moses* for PB and *Moses-chart* for HPB with the configuration as follows. For both, max-phrase-length=*7*, ttable-limit[7]=*20*, stack-size=*50* and max-pop-limit=*500*; For Moses, search-algorithm=*1* and distortion-limit=*6*; For Moses-chart, search-algorithm=*3* and max-char-span[8]=*20* for Moses-chart. We used both the distortion model and the lexical reordering model for Moses (denoted as *Moses-l*) except in §5.5 we only used the distortion model (denoted as *Moses-d*). We implemented the OSM according to Durrani et al. (2013) and used the same configuration with *Moses-l*. For *our method* we used the same configuration as *Moses-l* but adding an additional feature of the Markov model over MPs.

5.2 Performance Comparison

We first give the results of performance comparison. Here we add another system (denoted as *Moses-l+trgLM*): *Moses-l* together with the target language model trained on the training data set, using the same configuration with *Moses-l*. This system is used to test whether our model gains improvement just for using additional information on the training set. We use the open tool of Clark et al. (2011) to control for optimizer stability and test statistical significance.

The results are shown in Tables 1 and 2. The two language pairs we used are quite different: Chinese has a much bigger word order difference c.f. English than does Arabic. The results show that our system can outperform the baseline

[5] LDC2004E72, LDC2004T17, LDC2004T18, LDC2006T02

[6] We only employed MPs with the length ≤ 3. If a MP had more than 3 words on either side, we omitted the alignment links to the first target word of this MP and extracted MPs according to the new alignment.

[7] The maximum number of lexical rules for each source span.

[8] The maximum span on the source a rule can cover.

System	NIST 02 (dev)	NIST 03	NIST 05
Moses-l	36.0	32.8	32.0
Moses-chart	36.9	33.6	32.6
Moses-l+trgLM	36.4	33.9	32.9
OSM	36.6	34.0	33.1
our model	37.9	36.0	35.1

Table 1: BLEU % scores on the Chinese-English data set.

System	NIST 02 (dev)	NIST 03	NIST 05
Moses-l	60.4	52.0	52.8
Moses-chart	60.7	51.8	52.4
Moses-l+trgLM	60.8	52.6	53.3
OSM	61.1	52.9	53.4
our model	62.2	53.6	53.9

Table 2: BLEU % scores on the Arabic-English data set.

System	Chinese-English		Arabic-English	
	NIST 02	NIST 03	NIST 02	NIST 03
Moses-l	36.0	32.8	60.4	52.0
+t	36.3	33.8	60.9	52.4
+t+j	37.1	34.7	62.1	53.4
+t+j+s	37.6	34.8	62.5	53.9
+t+j+s+p	37.9	36.0	62.2	53.6

Table 3: The impact of factors and parallel backoff. Key: *t–target*, *j–jump*, *s–source*, *p–parallel backoff.*

System	2gram	3gram	4gram	5gram	6gram
Moses-l	27.2	32.4	33.0	32.8	33.2
our method	31.6	34.0	35.8	36.0	36.2

Table 4: The impact of the order of the standard language models.

systems significantly (with $p < 0.005$) on both language pairs, nevertheless, the improvement on Chinese-English is bigger. The big improvement over *Moses-l+trgLM* proves that the better performance of our model does not solely comes from the use of the training data. And the gain over OSM means our definition of factors gives a better handling to sparsity. We also notice that HPB does not give a higher BLEU score on Arabic-English than PB. The main difference between HPB and PB is that HPB employs gapped rules, so this result suggests that gaps are detrimental for Arabic-English translation. In §5.3, we experimentally validate this claim with our Markov model.

5.3 Impact of Factors and Parallel Backoff

We now seek to test the contribution of target, jump, source factors, as well as the parallel backoff technique in terms of BLEU score. We performed experiments on both Chinese-English and Arabic-English to test whether the contribution was related to language pairs. We designed the experiments as follows. We first trained a 3-gram Markov model only over target factors, $p(\bar{\boldsymbol{e}}_1^I|\bar{\boldsymbol{f}}_1^I) = \prod_{i=1}^I p(\bar{e}_i|\bar{e}_{i-2}^{i-1})$, denoted *+t*. Then we added the jump factor (*+t+j*), such that we now considering both target and jump events, $p(\bar{\boldsymbol{e}}_1^I, \bar{\boldsymbol{j}}_1^I|\bar{\boldsymbol{f}}_1^I) = \prod_{i=1}^I p(\bar{e}_i|\bar{j}_{i-2}^i, \bar{e}_{i-2}^{i-1}) p(\bar{j}_i|\bar{j}_{i-2}^{i-1}, \bar{e}_{i-2}^{i-1})$. Next we added the source factor (*+t+j+s*) such that now all three factors are included from Equation 1. For the above three Markov models we used simple least-recent backoff (akin to a standard language model), and consequently these methods cannot represent gaps in the target. Finally, we trained another Markov model by introducing parallel backoff to the third one as described in §2.1. Each of the four Markov model approaches are implemented as adding an additional feature, respectively, into the *Moses-l* baseline.

The results are shown in Table 3. Observe that adding each factor results in near uniform performance improvements on both language pairs. The jump factor gives big improvements of about 1% BLEU in both language pairs. However when using parallel backoff, the performance improves greatly for Chinese-English but degrades slightly on Arabic-English. The reason may be parallel backoff is used to encode common structures to capture the different word ordering between Chinese and English while for Arabic-English there are fewer consistent reordering patterns. This is also consistent with the results in Table 1 and 2 where HPB gets a little bit lower BLEU scores.

5.4 Impact of LM order

Our system resembles a language model in common use in SMT systems, in that it uses a Markov model over target words, among other factors. This raises the question of whether its improvements are due to it functioning as a target language model. Our experiments use order $k = 3$ over MP sequences and each MP can have at most 3 words. Therefore the model could in principle memorize 9-grams, although usually MPs are much smaller. To test whether our improvements are from using a higher-order language model or other reasons, we evaluate our system and the baseline system with a range of LMs of different order. If we can get consistent improvements over the baseline for

System	NIST 02 (dev)	NIST 03
Moses-d	35.1	31.3
Moses-l	36.0	32.8
Moses-d+M	36.4	34.8
Moses-l+M	37.9	36.0

Table 5: Comparison between our Markov model (denoted as *M*) and the lexical reordering model of Moses.

both small and large n, this suggests it's not the long context that plays the key role but is other information we have learned (e.g., jumps or rich structures).

Table 4 shows the results of using standard language models with orders $2-6$ in *Moses-l* and our method. We can see that language model order is very important. When we increase the order from 2 to 4, the BLEU scores for both systems increases drastically, but levels off for 4-gram and larger. Note that our system outperforms *Moses-l* by 4.4, 1.6, 2.8, 3.2 and 3.0 BLEU points, respectively. The large gain for 2-grams is likely due to the model behaving like a LM, however the fact that consistent gains are still realized for higher k suggests that the approach brings considerable complementary information, i.e., it is doing much more than simply language modelling.

5.5 Comparison with Lexical Reordering

Our Markov model learns a joint model of jump, source and target factors and this is similar to the lexical reordering model of Moses (Koehn et al., 2007), which learns general orientations of pairs of adjacent phrases (classed as monotone, swap or other). Our method is more complex, by learning explicit jump distances, while also using broader context. Here we compare the two methods, and test whether our approach is complementary by realizing gains over the lexicalized reordering baseline. We test this hypothesis by comparing the results of Moses with its simple distortion model (*Moses-d*), then with both simple distortion and lexicalized reordering (*Moses-l*), and then with our Markov model (denoted as *Moses-d+M* or *Moses-l+M*, for both baselines respectively).

The results are shown in Table 5. Comparing the results of *Moses-l* and *Moses-d*, we can see that the lexical reordering model outperforms the distortion model by a margin of 1.5% BLEU. Comparing *Moses-d+M* with *Moses-l*, our Markov model provides further improvements of 2.0% BLEU. Our approach does much more than model reordering, so it is unlikely that this improvement is solely due to being better a model of distortion. This is underscored by the final result in Table 5, for combining lexicalized distortion with our model (*Moses-l+M*) which gives the highest BLEU score, yielding another 1.2% increase.

System	NIST 02 (dev)	NIST 03
Moses-l	36.0	32.8
Moses-l+word	36.9	34.0
Moses-l+MP	37.6	34.8

Table 6: Comparison between the MP-based Markov model and the word-based Markov model.

5.6 Comparison with Word-based Markov

Our approach uses minimal phrases as its basic unit of translation, in order to preserve the many-to-many links found from the word alignments. However we now seek to assess the impact of the choice of these basic units, considering instead a simpler word-based setting which retains only 1-to-1 links in a Markov model. To do this, we processed target words left-to-right and for target words with multiple links, we only retained the link which had the highest lexical translation probability. Then we trained a 3-gram word-based Markov model which backs off by dropping the factors of the least recent word pairs in the order of first jump then source then target. This model was included as a feature in the *Moses-l* baseline (denoted as *Moses-l+word*), which we compared to a system using a MP-based Markov model backing off in the same way (denoted as *Moses-l+MP*).

According to the results in Table 6, using MPs leads to better performance. Surprisingly even the word based method outperforms the baseline. This points to inadequate phrase-pair features in the baseline, which can be more robustly estimated using a Markov decomposition. In addition to allowing for advanced smoothing, the Markov model can be considered to tile phrases over one another (each k-gram overlaps $k-1$ others) rather than enforcing a single segmentation as is done in the PB and HPB approaches. Fox (2002) states that phrases tend to move as a whole during reordering, i.e., breaking MPs into words opens the possibility of making more reordering errors. We could easily use larger phrase pairs as the basic unit, such as the phrases used during decoding. However, doing this involves a hard segmentation and would exacerbate issues of data sparsity.

6 Conclusions

In this paper we try to give a solution to the problems in phrase-based models, including weak generalization to unseen data and negligence of correlations between phrases. Our solution is to define a Markov model over minimal phrases so as to model translation conditioned on context and meanwhile use a fancy smoothing technique to learn richer structures such that can be applied to unseen data. Our method further decomposes each minimal phrase into three factors and operates in the unit of factors in the backoff process to provide a more robust modeling.

In our experiments, we prove that our definition of factored Markov model provides complementary information to lexicalized reordering and high order language models and the use of parallel backoff infers richer structures even those out of the reach of 2-SCFG and hence brings big performance improvements. Overall our approach gives significant improvements over strong baselines, giving consistent improvements of between 1.1 and 3.2 BLEU points on large scale Chinese-English and Arabic-English evaluations.

7 Acknowledges

The first author is supported by DARPA BOLT, contract HR0011-12-C-0014. The second author is the recipient of an Australian Research Council Future Fellowship (project number FT130101105). Thank the anonymous reviews for their insightful comments.

References

Jeff Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proc. of HLT-NAACL*.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL*, pages 310–318.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33:201–228.

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

Josep Maria Crego, François Yvon, and José B. Mariño. 2011. Ncode: an open source bilingual n-gram smt toolkit. *Prague Bull. Math. Linguistics*, 96:49–58.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proc. of ACL-HLT*, pages 1045–1054, June.

Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model with minimal translation units, but decode with phrases. In *Proc. of NAACL*, pages 1–11.

Yang Feng and Trevor Cohn. 2013. A markov model of machine translation using non-parametric bayesian inference. In *Proc. of ACL*, pages 333–342.

Heidi Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proc. of EMNLP*, pages 304–311, July.

Michel Galley and Christopher D. Manning. 2010. Accurate non-hierarchical phrase-based translation. In *Proc. of NAACL*, pages 966–974.

Katrin Kirchhoff, Jeff Bilmes, and Kevin Duh. 2007. Factored language models tutorial.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher 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 *Proc. of ACL, Demonstration Session*.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *Proc. of NAACL*, pages 39–48.

Frans J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Frans J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167.

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

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proc. of ICSLP*.