A Hierarchical Phrase-Based Model for Statistical Machine Translation

David Chiang
Institute for Advanced Computer Studies (UMIACS)
University of Maryland, College Park, MD 20742, USA
dchiang@umiacs.umd.edu

Abstract

We present a statistical phrase-based translation model that uses *hierarchical phrases*—phrases that contain subphrases. The model is formally a synchronous context-free grammar but is learned from a bitext without any syntactic information. Thus it can be seen as a shift to the *formal* machinery of syntax-based translation systems without any *linguistic* commitment. In our experiments using BLEU as a metric, the hierarchical phrase-based model achieves a relative improvement of 7.5% over Pharaoh, a state-of-the-art phrase-based system.

1 Introduction

The alignment template translation model (Och and Ney, 2004) and related phrase-based models advanced the previous state of the art by moving from words to *phrases* as the basic unit of translation. Phrases, which can be any substring and not necessarily phrases in any syntactic theory, allow these models to learn local reorderings, translation of short idioms, or insertions and deletions that are sensitive to local context. They are thus a simple and powerful mechanism for machine translation.

The basic phrase-based model is an instance of the noisy-channel approach (Brown et al., 1993),[1] in which the translation of a French sentence f into an English sentence e is modeled as:

$$\arg\max_e P(e \mid f) = \arg\max_e P(e, f) \quad (1)$$

$$= \arg\max_e (P(e) \times P(f \mid e)) \quad (2)$$

The translation model $P(f \mid e)$ "encodes" e into f by the following steps:

1. segment e into phrases $\bar{e}_1 \cdots \bar{e}_I$, typically with a uniform distribution over segmentations;
2. reorder the $\bar{e}_i$ according to some distortion model;
3. translate each of the $\bar{e}_i$ into French phrases according to a model $P(\bar{f} \mid \bar{e})$ estimated from the training data.

Other phrase-based models model the joint distribution $P(e, f)$ (Marcu and Wong, 2002) or made $P(e)$ and $P(f \mid e)$ into features of a log-linear model (Och and Ney, 2002). But the basic architecture of phrase segmentation (or generation), phrase reordering, and phrase translation remains the same.

Phrase-based models can robustly perform translations that are localized to substrings that are common enough to have been observed in training. But Koehn et al. (2003) find that phrases longer than three words improve performance little, suggesting that data sparseness takes over for longer phrases. Above the phrase level, these models typically have a simple distortion model that reorders phrases independently of their content (Och and Ney, 2004; Koehn et al., 2003), or not at all (Zens and Ney, 2004; Kumar et al., 2005).

But it is often desirable to capture translations whose scope is larger than a few consecutive words.

[1] Throughout this paper, we follow the convention of Brown et al. of designating the source and target languages as "French" and "English," respectively. The variables f and e stand for source and target sentences; f_i^j stands for the substring of f from position i to position j inclusive, and similarly for e_i^j.

Proceedings of the 43rd Annual Meeting of the ACL, pages 263–270,
Ann Arbor, June 2005.

Consider the following Mandarin example and its English translation:

(3) 澳洲 是 与 北 韩 有 邦交
Aozhou shi yu Bei Han you bangjiao
Australia is with North Korea have dipl. rels.
的 少数 国家 之一
de shaoshu guojia zhiyi
that few countries one of

'Australia is one of the few countries that have diplomatic relations with North Korea'

If we count *zhiyi*, lit. 'of-one,' as a single token, then translating this sentence correctly into English requires reversing a sequence of five elements. When we run a phrase-based system, Pharaoh (Koehn et al., 2003; Koehn, 2004a), on this sentence (using the experimental setup described below), we get the following phrases with translations:

(4) [Aozhou] [shi] [yu] [Bei Han] [you] [bangjiao]$_1$ [de shaoshu guojia zhiyi]

[Australia] [is] [dipl. rels.]$_1$ [with] [North Korea] [is] [one of the few countries]

where we have used subscripts to indicate the reordering of phrases. The phrase-based model is able to order "diplomatic...Korea" correctly (using phrase reordering) and "one...countries" correctly (using a phrase translation), but does not accomplish the necessary inversion of those two groups. A lexicalized phrase-reordering model like that in use in ISI's system (Och et al., 2004) might be able to learn a better reordering, but simpler distortion models will probably not.

We propose a solution to these problems that does not interfere with the strengths of the phrase-based approach, but rather capitalizes on them: since phrases are good for learning reorderings of words, we can use them to learn reorderings of phrases as well. In order to do this we need *hierarchical phrases* that consist of both words and subphrases. For example, a hierarchical phrase pair that might help with the above example is:

(5) ⟨yu [1] you [2], have [2] with [1]⟩

where [1] and [2] are placeholders for subphrases. This would capture the fact that Chinese PPs almost always modify VP on the left, whereas English PPs usually modify VP on the right. Because it generalizes over possible prepositional objects and direct objects, it acts both as a discontinuous phrase pair and as a phrase-reordering rule. Thus it is considerably more powerful than a conventional phrase pair.

Similarly,

(6) ⟨[1] de [2], the [2] that [1]⟩

would capture the fact that Chinese relative clauses modify NPs on the left, whereas English relative clauses modify on the right; and

(7) ⟨[1] zhiyi, one of [1]⟩

would render the construction *zhiyi* in English word order. These three rules, along with some conventional phrase pairs, suffice to translate the sentence correctly:

(8) [Aozhou] [shi] [[[yu [Bei Han]$_1$ you [bangjiao]$_2$] de [shaoshu guojia]$_3$] zhiyi]

[Australia] [is] [one of [the [few countries]$_3$ that [have [dipl. rels.]$_2$ with [North Korea]$_1$]]]

The system we describe below uses rules like this, and in fact is able to learn them automatically from a bitext without syntactic annotation. It translates the above example almost exactly as we have shown, the only error being that it omits the word 'that' from (6) and therefore (8).

These hierarchical phrase pairs are formally productions of a synchronous context-free grammar (defined below). A move to synchronous CFG can be seen as a move towards syntax-based MT; however, we make a distinction here between *formally* syntax-based and *linguistically* syntax-based MT. A system like that of Yamada and Knight (2001) is both formally and linguistically syntax-based: formally because it uses synchronous CFG, linguistically because the structures it is defined over are (on the English side) informed by syntactic theory (via the Penn Treebank). Our system is formally syntax-based in that it uses synchronous CFG, but not necessarily linguistically syntax-based, because it induces a grammar from a parallel text without relying on any linguistic annotations or assumptions; the result sometimes resembles a syntactician's grammar but often does not. In this respect it resembles Wu's

bilingual bracketer (Wu, 1997), but ours uses a different extraction method that allows more than one lexical item in a rule, in keeping with the phrase-based philosophy. Our extraction method is basically the same as that of Block (2000), except we allow more than one nonterminal symbol in a rule, and use a more sophisticated probability model.

In this paper we describe the design and implementation of our hierarchical phrase-based model, and report on experiments that demonstrate that hierarchical phrases indeed improve translation.

2 The model

Our model is based on a weighted synchronous CFG (Aho and Ullman, 1969). In a synchronous CFG the elementary structures are rewrite rules with aligned pairs of right-hand sides:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \tag{9}$$

where X is a nonterminal, γ and α are both strings of terminals and nonterminals, and $\sim$ is a one-to-one correspondence between nonterminal occurrences in γ and nonterminal occurrences in α. Rewriting begins with a pair of linked start symbols. At each step, two coindexed nonterminals are rewritten using the two components of a single rule, such that none of the newly introduced symbols is linked to any symbols already present.

Thus the hierarchical phrase pairs from our above example could be formalized in a synchronous CFG as:

$$\text{X} \rightarrow \langle \text{yu X}_{\boxed{1}} \text{ you X}_{\boxed{2}}, \text{have X}_{\boxed{2}} \text{ with X}_{\boxed{1}} \rangle \tag{10}$$

$$\text{X} \rightarrow \langle \text{X}_{\boxed{1}} \text{ de X}_{\boxed{2}}, \text{the X}_{\boxed{2}} \text{ that X}_{\boxed{1}} \rangle \tag{11}$$

$$\text{X} \rightarrow \langle \text{X}_{\boxed{1}} \text{ zhiyi, one of X}_{\boxed{1}} \rangle \tag{12}$$

where we have used boxed indices to indicate which occurrences of X are linked by $\sim$.

Note that we have used only a single nonterminal symbol X instead of assigning syntactic categories to phrases. In the grammar we extract from a bitext (described below), all of our rules use only X, except for two special "glue" rules, which combine a sequence of Xs to form an S:

$$\text{S} \rightarrow \langle \text{S}_{\boxed{1}} \text{X}_{\boxed{2}}, \text{S}_{\boxed{1}} \text{X}_{\boxed{2}} \rangle \tag{13}$$

$$\text{S} \rightarrow \langle \text{X}_{\boxed{1}}, \text{X}_{\boxed{1}} \rangle \tag{14}$$

These give the model the option to build only partial translations using hierarchical phrases, and then combine them serially as in a standard phrase-based model. For a partial example of a synchronous CFG derivation, see Figure 1.

Following Och and Ney (2002), we depart from the traditional noisy-channel approach and use a more general log-linear model. The weight of each rule is:

$$w(X \rightarrow \langle \gamma, \alpha \rangle) = \prod_i \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i} \tag{15}$$

where the ϕ_i are features defined on rules. For our experiments we used the following features, analogous to Pharaoh's default feature set:

- $P(\gamma \mid \alpha)$ and $P(\alpha \mid \gamma)$, the latter of which is not found in the noisy-channel model, but has been previously found to be a helpful feature (Och and Ney, 2002);
- the lexical weights $P_w(\gamma \mid \alpha)$ and $P_w(\alpha \mid \gamma)$ (Koehn et al., 2003), which estimate how well the words in α translate the words in γ;[2]
- a phrase penalty exp(1), which allows the model to learn a preference for longer or shorter derivations, analogous to Koehn's phrase penalty (Koehn, 2003).

The exceptions to the above are the two glue rules, (13), which has weight one, and (14), which has weight

$$w(\text{S} \rightarrow \langle \text{S}_{\boxed{1}} \text{X}_{\boxed{2}}, \text{S}_{\boxed{1}} \text{X}_{\boxed{2}} \rangle) = \exp(-\lambda_g) \tag{16}$$

the idea being that λ_g controls the model's preference for hierarchical phrases over serial combination of phrases.

Let D be a derivation of the grammar, and let $f(D)$ and $e(D)$ be the French and English strings generated by D. Let us represent D as a set of triples $\langle r, i, j \rangle$, each of which stands for an application of a grammar rule r to rewrite a nonterminal that spans $f(D)_i^j$ on the French side.[3] Then the weight of D

[2] This feature uses word alignment information, which is discarded in the final grammar. If a rule occurs in training with more than one possible word alignment, Koehn et al. take the maximum lexical weight; we take a weighted average.

[3] This representation is not completely unambiguous, but is sufficient for defining the model.

$\langle \mathrm{S}_{\boxed{1}}, \mathrm{S}_{\boxed{1}} \rangle \Rightarrow \langle \mathrm{S}_{\boxed{2}} \mathrm{X}_{\boxed{3}}, \mathrm{S}_{\boxed{2}} \mathrm{X}_{\boxed{3}} \rangle$
$\Rightarrow \langle \mathrm{S}_{\boxed{4}} \mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}}, \mathrm{S}_{\boxed{4}} \mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}} \rangle$
$\Rightarrow \langle \mathrm{X}_{\boxed{6}} \mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}}, \mathrm{X}_{\boxed{6}} \mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}} \rangle$
$\Rightarrow \langle$ Aozhou $\mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}}$, Australia $\mathrm{X}_{\boxed{5}} \mathrm{X}_{\boxed{3}} \rangle$
$\Rightarrow \langle$ Aozhou shi $\mathrm{X}_{\boxed{3}}$, Australia is $\mathrm{X}_{\boxed{3}} \rangle$
$\Rightarrow \langle$ Aozhou shi $\mathrm{X}_{\boxed{7}}$ zhiyi, Australia is one of $\mathrm{X}_{\boxed{7}} \rangle$
$\Rightarrow \langle$ Aozhou shi $\mathrm{X}_{\boxed{8}}$ de $\mathrm{X}_{\boxed{9}}$ zhiyi, Australia is one of the $\mathrm{X}_{\boxed{9}}$ that $\mathrm{X}_{\boxed{8}} \rangle$
$\Rightarrow \langle$ Aozhou shi yu $\mathrm{X}_{\boxed{1}}$ you $\mathrm{X}_{\boxed{2}}$ de $\mathrm{X}_{\boxed{9}}$ zhiyi, Australia is one of the $\mathrm{X}_{\boxed{9}}$ that have $\mathrm{X}_{\boxed{2}}$ with $\mathrm{X}_{\boxed{1}} \rangle$

Figure 1: Example partial derivation of a synchronous CFG.

is the product of the weights of the rules used in the translation, multiplied by the following extra factors:

$$w(D) = \prod_{\langle r,i,j \rangle \in D} w(r) \times p_{lm}(e)^{\lambda_{lm}} \times \exp(-\lambda_{wp}|e|) \quad (17)$$

where p_{lm} is the language model, and $\exp(-\lambda_{wp}|e|)$, the word penalty, gives some control over the length of the English output.

We have separated these factors out from the rule weights for notational convenience, but it is conceptually cleaner (and necessary for polynomial-time decoding) to integrate them into the rule weights, so that the whole model is a weighted synchronous CFG. The word penalty is easy; the language model is integrated by intersecting the English-side CFG with the language model, which is a weighted finite-state automaton.

3 Training

The training process begins with a word-aligned corpus: a set of triples $\langle f, e, \sim \rangle$, where f is a French sentence, e is an English sentence, and $\sim$ is a (many-to-many) binary relation between positions of f and positions of e. We obtain the word alignments using the method of Koehn et al. (2003), which is based on that of Och and Ney (2004). This involves running GIZA++ (Och and Ney, 2000) on the corpus in both directions, and applying refinement rules (the variant they designate "final-and") to obtain a single many-to-many word alignment for each sentence.

Then, following Och and others, we use heuristics to hypothesize a distribution of possible derivations of each training example, and then estimate the phrase translation parameters from the hypothesized distribution. To do this, we first identify *initial phrase* pairs using the same criterion as previous systems (Och and Ney, 2004; Koehn et al., 2003):

Definition 1. Given a word-aligned sentence pair $\langle f, e, \sim \rangle$, a rule $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair of $\langle f, e, \sim \rangle$ iff:

1. $f_k \sim e_{k'}$ for some $k \in [i, j]$ and $k' \in [i', j']$;
2. $f_k \nsim e_{k'}$ for all $k \in [i, j]$ and $k' \notin [i', j']$;
3. $f_k \nsim e_{k'}$ for all $k \notin [i, j]$ and $k' \in [i', j']$.

Next, we form all possible differences of phrase pairs:

Definition 2. The set of rules of $\langle f, e, \sim \rangle$ is the smallest set satisfying the following:

1. If $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair, then
$$\mathrm{X} \rightarrow \langle f_i^j, e_{i'}^{j'} \rangle$$
is a rule.
2. If $r = \mathrm{X} \rightarrow \langle \gamma, \alpha \rangle$ is a rule and $\langle f_i^j, e_{i'}^{j'} \rangle$ is an initial phrase pair such that $\gamma = \gamma_1 f_i^j \gamma_2$ and $\alpha = \alpha_1 e_{i'}^{j'} \alpha_2$, then
$$\mathrm{X} \rightarrow \langle \gamma_1 \mathrm{X}_{\boxed{k}} \gamma_2, \alpha_1 \mathrm{X}_{\boxed{k}} \alpha_2 \rangle$$
is a rule, where k is an index not used in r.

The above scheme generates a very large number of rules, which is undesirable not only because it makes training and decoding very slow, but also

because it creates *spurious ambiguity*—a situation where the decoder produces many derivations that are distinct yet have the same model feature vectors and give the same translation. This can result in *n*-best lists with very few different translations or feature vectors, which is problematic for the algorithm we use to tune the feature weights. Therefore we filter our grammar according to the following principles, chosen to balance grammar size and performance on our development set:

1. If there are multiple initial phrase pairs containing the same set of alignment points, we keep only the smallest.
2. Initial phrases are limited to a length of 10 on the French side, and rule to five (nonterminals plus terminals) on the French right-hand side.
3. In the subtraction step, f_i^j must have length greater than one. The rationale is that little would be gained by creating a new rule that is no shorter than the original.
4. Rules can have at most two nonterminals, which simplifies the decoder implementation. Moreover, we prohibit nonterminals that are adjacent on the French side, a major cause of spurious ambiguity.
5. A rule must have at least one pair of aligned words, making translation decisions always based on some lexical evidence.

Now we must hypothesize weights for all the derivations. Och's method gives equal weight to all the extracted phrase occurences. However, our method may extract many rules from a single initial phrase pair; therefore we distribute weight equally among initial phrase pairs, but distribute that weight equally among the rules extracted from each. Treating this distribution as our observed data, we use relative-frequency estimation to obtain $P(\gamma \mid \alpha)$ and $P(\alpha \mid \gamma)$.

4 Decoding

Our decoder is a CKY parser with beam search together with a postprocessor for mapping French derivations to English derivations. Given a French sentence f, it finds the best derivation (or n best derivations, with little overhead) that generates $\langle f, e \rangle$ for some e. Note that we find the English yield of the highest-probability single derivation

$$e\left(\arg\max_{D \text{ s.t. } f(D) = f} w(D)\right) \tag{18}$$

and not necessarily the highest-probability e, which would require a more expensive summation over derivations.

We prune the search space in several ways. First, an item that has a score worse than β times the best score in the same cell is discarded; second, an item that is worse than the bth best item in the same cell is discarded. Each cell contains all the items standing for X spanning f_i^j. We choose b and β to balance speed and performance on our development set. For our experiments, we set $b = 40, \beta = 10^{-1}$ for X cells, and $b = 15, \beta = 10^{-1}$ for S cells. We also prune rules that have the same French side ($b = 100$).

The parser only operates on the French-side grammar; the English-side grammar affects parsing only by increasing the effective grammar size, because there may be multiple rules with the same French side but different English sides, and also because intersecting the language model with the English-side grammar introduces many states into the nonterminal alphabet, which are projected over to the French side. Thus, our decoder's search space is many times larger than a monolingual parser's would be. To reduce this effect, we apply the following heuristic when filling a cell: if an item falls outside the beam, then any item that would be generated using a lower-scoring rule or a lower-scoring antecedent item is also assumed to fall outside the beam. This heuristic greatly increases decoding speed, at the cost of some search errors.

Finally, the decoder has a constraint that prohibits any X from spanning a substring longer than 10 on the French side, corresponding to the maximum length constraint on initial rules during training. This makes the decoding algorithm asymptotically linear-time.

The decoder is implemented in Python, an interpreted language, with C++ code from the SRI Language Modeling Toolkit (Stolcke, 2002). Using the settings described above, on a 2.4 GHz Pentium IV, it takes about 20 seconds to translate each sentence (average length about 30). This is faster than our

Python implementation of a standard phrase-based decoder, so we expect that a future optimized implementation of the hierarchical decoder will run at a speed competitive with other phrase-based systems.

5 Experiments

Our experiments were on Mandarin-to-English translation. We compared a baseline system, the state-of-the-art phrase-based system Pharaoh (Koehn et al., 2003; Koehn, 2004a), against our system. For all three systems we trained the translation model on the FBIS corpus (7.2M+9.2M words); for the language model, we used the SRI Language Modeling Toolkit to train a trigram model with modified Kneser-Ney smoothing (Chen and Goodman, 1998) on 155M words of English newswire text, mostly from the Xinhua portion of the Gigaword corpus. We used the 2002 NIST MT evaluation test set as our development set, and the 2003 test set as our test set. Our evaluation metric was BLEU (Papineni et al., 2002), as calculated by the NIST script (version 11a) with its default settings, which is to perform case-insensitive matching of n-grams up to $n = 4$, and to use the shortest (as opposed to nearest) reference sentence for the brevity penalty. The results of the experiments are summarized in Table 1.

5.1 Baseline

The baseline system we used for comparison was Pharaoh (Koehn et al., 2003; Koehn, 2004a), as publicly distributed. We used the default feature set: language model (same as above), $p(\bar{f} \mid \bar{e})$, $p(\bar{e} \mid \bar{f})$, lexical weighting (both directions), distortion model, word penalty, and phrase penalty. We ran the trainer with its default settings (maximum phrase length 7), and then used Koehn's implementation of minimum-error-rate training (Och, 2003) to tune the feature weights to maximize the system's BLEU score on our development set, yielding the values shown in Table 2. Finally, we ran the decoder on the test set, pruning the phrase table with $b = 100$, pruning the chart with $b = 100, \beta = 10^{-5}$, and limiting distortions to 4. These are the default settings, except for the phrase table's b, which was raised from 20, and the distortion limit. Both of these changes, made by Koehn's minimum-error-rate trainer by default, improve performance on the development set.

Rank	Chinese	English
1	。	.
3	的	the
14	在	in
23	的	's
577	$X_{[1]}$ 的 $X_{[2]}$	the $X_{[2]}$ of $X_{[1]}$
735	$X_{[1]}$ 的 $X_{[2]}$	the $X_{[2]}$ $X_{[1]}$
763	$X_{[1]}$ 之一	one of $X_{[1]}$
1201	$X_{[1]}$ 总统	president $X_{[1]}$
1240	$X_{[1]}$ 美元	\$ $X_{[1]}$
2091	今年 $X_{[1]}$	$X_{[1]}$ this year
3253	百分之 $X_{[1]}$	$X_{[1]}$ percent
10508	在 $X_{[1]}$ 下	under $X_{[1]}$
28426	在 $X_{[1]}$ 前	before $X_{[1]}$
47015	$X_{[1]}$ 的 $X_{[2]}$	the $X_{[2]}$ that $X_{[1]}$
1752457	与 $X_{[1]}$ 有 $X_{[2]}$	have $X_{[2]}$ with $X_{[1]}$

Figure 2: A selection of extracted rules, with ranks after filtering for the development set. All have X for their left-hand sides.

5.2 Hierarchical model

We ran the training process of Section 3 on the same data, obtaining a grammar of 24M rules. When filtered for the development set, the grammar has 2.2M rules (see Figure 2 for examples). We then ran the minimum-error rate trainer with our decoder to tune the feature weights, yielding the values shown in Table 2. Note that λ_g penalizes the glue rule much less than λ_{pp} does ordinary rules. This suggests that the model will prefer serial combination of phrases, unless some other factor supports the use of hierarchical phrases (e.g., a better language model score).

We then tested our system, using the settings described above.[4] Our system achieves an absolute improvement of 0.02 over the baseline (7.5% relative), without using any additional training data. This difference is statistically significant ($p < 0.01$).[5] See Table 1, which also shows that the relative gain is higher for higher n-grams.

[4] Note that we gave Pharaoh wider beam settings than we used on our own decoder; on the other hand, since our decoder's chart has more cells, its b limits do not need to be as high.

[5] We used Zhang's significance tester (Zhang et al., 2004), which uses bootstrap resampling (Koehn, 2004b); it was modified to conform to NIST's current definition of the BLEU brevity penalty.

System	BLEU-n 4	n-gram precisions 1	2	3	4	5	6	7	8
Pharaoh	0.2676	0.72	0.37	0.19	0.10	0.052	0.027	0.014	0.0075
hierarchical	0.2877	0.74	0.39	0.21	0.11	0.060	0.032	0.017	0.0084
+constituent	0.2881	0.73	0.39	0.21	0.11	0.062	0.032	0.017	0.0088

Table 1: Results on baseline system and hierarchical system, with and without constituent feature.

System	Features $P_{lm}(e)$	$P(\gamma\|\alpha)$	$P(\alpha\|\gamma)$	$P_w(\gamma\|\alpha)$	$P_w(\alpha\|\gamma)$	Word	Phr	λ_d	λ_g	λ_c
Pharaoh	0.19	0.095	0.030	0.14	0.029	−0.20	0.22	0.11	—	—
hierarchical	0.15	0.036	0.074	0.037	0.076	−0.32	0.22	—	0.09	—
+constituent	0.11	0.026	0.062	0.025	0.029	−0.23	0.21	—	0.11	0.20

Table 2: Feature weights obtained by minimum-error-rate training (normalized so that absolute values sum to one). Word = word penalty; Phr = phrase penalty. Note that we have inverted the sense of Pharaoh's phrase penalty so that a positive weight indicates a penalty.

5.3 Adding a constituent feature

The use of hierarchical structures opens the possibility of making the model sensitive to syntactic structure. Koehn et al. (2003) mention German ⟨es gibt, there is⟩ as an example of a good phrase pair which is not a syntactic phrase pair, and report that favoring syntactic phrases does not improve accuracy. But in our model, the rule

(19) $$\mathrm{X} \rightarrow \langle \text{es gibt } \mathrm{X}_{\boxed{1}}, \text{there is } \mathrm{X}_{\boxed{1}} \rangle$$

would indeed respect syntactic phrases, because it builds a pair of Ss out of a pair of NPs. Thus, favoring subtrees in our model that are syntactic phrases might provide a fairer way of testing the hypothesis that syntactic phrases are better phrases.

This feature adds a factor to (17),

(20) $$c(i,j) = \begin{cases} 1 & \text{if } f_i^j \text{ is a constituent} \\ 0 & \text{otherwise} \end{cases}$$

as determined by a statistical tree-substitution-grammar parser (Bikel and Chiang, 2000), trained on the Penn Chinese Treebank, version 3 (250k words). Note that the parser was run only on the test data and not the (much larger) training data. Rerunning the minimum-error-rate trainer with the new feature yielded the feature weights shown in Table 2. Although the feature improved accuracy on the development set (from 0.314 to 0.322), it gave no statistically significant improvement on the test set.

6 Conclusion

Hierarchical phrase pairs, which can be learned without any syntactically-annotated training data, improve translation accuracy significantly compared with a state-of-the-art phrase-based system. They also facilitate the incorporation of syntactic information, which, however, did not provide a statistically significant gain.

Our primary goal for the future is to move towards a more syntactically-motivated grammar, whether by automatic methods to induce syntactic categories, or by better integration of parsers trained on annotated data. This would potentially improve both accuracy and efficiency. Moreover, reducing the grammar size would allow more ambitious training settings. The maximum initial phrase length is currently 10; preliminary experiments show that increasing this limit to as high as 15 does improve accuracy, but requires more memory. On the other hand, we have successfully trained on almost 30M+30M words by tightening the initial phrase length limit for part of the data. Streamlining the grammar would allow further experimentation in these directions.

In any case, future improvements to this system will maintain the design philosophy proven here, that ideas from syntax should be incorporated into statistical translation, but not in exchange for the strengths of the phrase-based approach.

Acknowledgements

I would like to thank Philipp Koehn for the use of the Pharaoh software; and Adam Lopez, Michael Subotin, Nitin Madnani, Christof Monz, Liang Huang, and Philip Resnik. This work was partially supported by ONR MURI contract FCPO.810548265 and Department of Defense contract RD-02-5700. *S. D. G.*

References

A. V. Aho and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3:37–56.

Daniel M. Bikel and David Chiang. 2000. Two statistical parsing models applied to the Chinese Treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 1–6.

Hans Ulrich Block. 2000. Example-based incremental synchronous interpretation. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 411–417. Springer-Verlag, Berlin.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University Center for Research in Computing Technology.

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

Philipp Koehn. 2003. *Noun Phrase Translation*. Ph.D. thesis, University of Southern California.

Philipp Koehn. 2004a. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

Philipp Koehn. 2004b. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Shankar Kumar, Yonggang Deng, and William Byrne. 2005. A weighted finite state transducer translation template model for statistical machine translation. *Natural Language Engineering*. To appear.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 133–139.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the ACL*, pages 440–447.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 295–302.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.

Franz Josef Och, Ignacio Thayer, Daniel Marcu, Kevin Knight, Dragos Stefan Munteanu, Quamrul Tipu, Michel Galley, and Mark Hopkins. 2004. Arabic and Chinese MT at USC/ISI. Presentation given at NIST Machine Translation Evaluation Workshop.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the 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 the 40th Annual Meeting of the ACL*, pages 311–318.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–404.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the ACL*, pages 523–530.

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

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051–2054.