# Learning a Log-Linear Model with Bilingual Phrase-Pair Features for Statistical Machine Translation

**Bing Zhao** and **Alex Waibel**
School of Computer Science
Carnegie Mellon University
{bzhao, waibel}@cs.cmu.edu

## Abstract

We propose a set of informative feature functions together with a log-linear model framework for bilingual phrase-pair extraction to improve phrase-based statistical machine translation. The base feature functions investigated are phrase length model, phrase-level centers' distortion, lexicon translation equivalence, bracketing constraints and word alignment links. Two generative models show strong baselines with these base features, illustrating the effectiveness of the proposed feature functions. Strategies of extending the features and a log-linear model of learning the weighted combination of them are proposed to effectively extract phrase-pairs from parallel data. Experimental results of TIDES'03 Chinese-English small data track show improved translation qualities.

## 1 Introduction

Bilingual phrase-pair extraction from parallel data becomes a key component in today's state of the art phrase-based statistical machine translation systems. The significant advantages of using phrase-pairs over word level mixture models (Brown et al., 1993) are that both fertilities and distortions above phrase level are simpler to model and phrase-based approaches have flexibilities of modeling local word-reordering and are less sensitive to the preprocessing errors such as word segmentations (e.g., for Chinese and Japanese) and tokenization. These advantages are observed and supported by the positive evidences from many previous works such as (Wu, 1997; Och and Ney, 2004; Koehn et al., 2003; Zens and Ney, 2004; Vogel et al., 2003).

In practice, a phrase-pair extraction based on word alignment (Koehn, 2004; Tillmann, 2003; Och and Ney, 2004) works quite well given simple heuristics and millions of parallel sentences to cover enough instances needed for translation. However, it is more or less difficult to extend the heuristics to include more informative clues for better phrase-pair extractions. We propose a principled framework of combining a set of informative feature functions via a log-linear model for bilingual phrase-pair extraction. In particular, we introduce a few informative feature functions in keeping the philosophy of phrase translations. The base feature functions are designed to model three diverse aspects of a *block* (a phrase pair): a phrase-level length model to approximate phrasal length relevance, a center distortion model to model relative positions' differences for phrases in a given sentence pair and a generative phrase level lexical model to model translational equivalence. Furthermore, we extend the base feature functions by symmetrizing with noisy-channel models in two directions: source-to-target and target-to-source. The utilities of the proposed features are demonstrated using two generative models establishing strong baselines; together with constraints from word alignments, these feature functions are combined in a log-linear model to extract phrase pairs effectively; improved translation qualities are achieved over a state of the art system on TIDES'03 Chinese-English small data tack.

The remainder paper is structured as follows: in Section 2, statistical machine translation is briefly reviewed with notations for Blocks; in Section 3, base feature functions are explained; Section 4 contains two novel generative models as our baselines; our proposed log-linear model with extended feature functions are in Section 5; in Section 6, our experimental results; conclusions and discussions are given in Section 7.

## 2 Statistical Machine Translation

Generally speaking, the task of statistical machine translation (SMT) is to translate one sentence in a source language *F* into a target language *E*. For example[1], given a French sentence **f** with $J$ words denoted as $f_1^J = f_1 f_2 ... f_J$, an SMT system automatically translates it into an English sentence **e** with $I$ words denoted by $e_1^I = e_1 e_2 ... e_I$. The SMT system first proposes English hypotheses in its model space. Among all the hypotheses, the system selects the one with the highest conditional probability according to Bayes's decision rule as follows:

$$\hat{e}_1^I = \arg\max_{\{e_1^I\}} P(e_1^I|f_1^J) = \arg\max_{\{e_1^I\}} P(f_1^J|e_1^I)P(e_1^I), \quad (1)$$

where $P(f_1^J|e_1^I)$ is called *translation model* and $P(e_1^I)$ is the *language model*. For most of the phrase based machine translation systems, the translation model are essentially a collection of bilingual phrase-pairs extracted from parallel sentence pairs. This paper focus on how to extract high quality phrase-pairs from parallel data.

Each phrase pair is represented as a *Block*:$X$ in a given parallel sentence pair:

$$X \rightarrow (f_j^{j+l}, e_i^{i+k}), \quad (2)$$

where $f_j^{j+l}$ is the source phrase with $(l+1)$ French words; its projection is $e_i^{i+k}$ in the target sentence with left boundary at the position of $i$ and right boundary at $(i+k)$. We view the *phrase-pair extraction* as a local search algorithm: given a source phrase $f_j^{j+l}$, search for the projected boundaries of candidate target phrase $e_i^{i+k}$ according to a weighted combination of diverse feature functions in a log-linear model. The log linear model then servers as a performance measure to guide a local search (i.e., a stochastic hill-climbing) to extract bilingual phrase pairs from the parallel data.

## 3 Base Feature Functions

We introduce several informative base feature functions, some of which are shown to be helpful in our previous works (Zhao and Vogel, 2005). They'll be briefly summarized here to be more self-contained. Each of the feature functions corresponds to one special aspect of the block embedded in the context of a given sentence pair. These aspects are then quantified by the proposed sub-models or our designed feature functions.

[1]We follow the notations used in (Brown et al., 1993)

### 3.1 A Phrase-Level Length Model

Given the word fertility defined as in (Brown et al., 1993), we can compute a probability to predict *phrase length* relevance between a pair of phrases: given the candidate target phrase (English) $e_1^I$ and a source phrase (French) $f_1^J$ of length $J$, the model gives the probabilistic estimation of $P(J|e_1^I)$ via a dynamic programming algorithm using the English word fertilities models $P(\phi|e_i)$. Figure 1 shows an example fertility

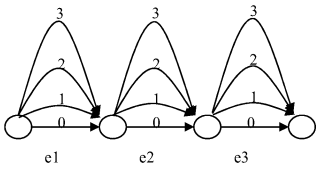


Figure 1: A trellis of an English trigram with maximum fertility of 4 for dynamic programming

trellis of an English trigram, where each edge between two nodes represents one English word $e_i$. The arc between two nodes represents one candidate non-zero fertility for word $e_i$. The fertility of zero (i.e., generating a Null word) corresponds to the direct edge between two nodes and thus, the Null word is incorporated into this model's representation. Each arc is associated with a English word fertility probability $P(\phi_i|e_i)$. A path $\phi_1^I$ through the trellis represents the number of French words $\phi_i$ generated by the English trigram. Thus, the probability of generating $J$ words from the English phrase along the Viterbi path is:

$$P(J|e_1^I) = \max_{\{\phi_1^I, J=\sum_{i=1}^I \phi_i\}} \prod_{i=1}^{I} P(\phi_i|e_i) \quad (3)$$

The Viterbi path is inferred via dynamic programming as follows:

$$\phi[j,i] = max \begin{cases} \phi[j,i-1] + \log P_{Null}(0|e_i) \\ \phi[j-1,i-1] + \log P_\phi(1|e_i) \\ \phi[j-2,i-1] + \log P_\phi(2|e_i) \\ \phi[j-3,i-1] + \log P_\phi(3|e_i) \end{cases}$$

where $P_{Null}(0|e_i)$ is the probability of generating a Null word from $e_i$; $P_\phi(k=1|e_i)$ is the usual word fertility model of generating one French word from the word $e_i$; $\phi[j,i]$ is the cost so far for generating $j$ words from $e_1^i : e_1, \cdots, e_i$. After computing the cost of $\phi[J,I]$, we can trace back the Viterbi path, along which the probability $P(J|e_1^I)$ of generating $J$ French words from the English phrase $e_1^I$ as shown in Eqn. 3.

Thus, for each block, one can compute a fertility based score Eqn. 3 to estimate to how relevant the phrase pairs' lengths are to each other.

### 3.2 A Center-Distortion Model

As introduced in Section 1, the distortion model above phrase level is usually easier to model. Empirical observations show that most high quality blocks are located close to the diagonal or the inverse diagonal in the alignment matrix of a sentence pair. A simple distortion model is designed to estimate how far away the phrase pairs are from each other.

The *center* $\odot_{f_j^{j+l}}$ of the phrase $f_j^{j+l}$ is a normalized *relative position* in the source sentence defined as follows:

$$\odot_{f_j^{j+l}} = \frac{1}{J} \sum_{j'=j}^{j'=j+l} \frac{j'}{l+1} \tag{4}$$

The center of the English phrase is computed accordingly. Figure 2 shows histograms of the differences between the centers: $(\odot_{f_j^{j+l}} - \odot_{e_i^{i+k}})$ of $30.8K$ oracle phrase pairs extracted from 627 human word-aligned sentence pairs: for each source phrase, find the left-most and right-most projected positions in the target sentence according to both word alignment and the coherence constraint (Fox, 2002).

For phrase-pair extraction, the expected center of the phrase $e_i^{i+k}$ is estimated for a given French phrase's center and then a local search starting around it is carried out to get candidate target phrases. The expected relative center for every French word $f_{j'}$ is first computed as follows:

$$\odot_{e_i^{i+k}}(f_{j'}) = \frac{1}{I} \cdot \frac{\sum_{i'=i}^{(i+k)} i' \cdot P(f_{j'}|e_{i'})}{\sum_{i'=i}^{(i+k)} P(f_{j'}|e_{i'})}, \tag{5}$$

where $P(f_{j'}|e_i)$ is the word translation lexicon estimated in IBM Models. $i$ is the position index, which is weighted by the word level translation probabilities; the term of $\sum_{i=1}^{I} P(f_{j'}|e_i)$ provides a normalization so that the expected center is within the range of target sentence length. After this, the expected center of $e_i^{i+k}$ is simply a average of $\odot_{e_i^{i+k}}(f_{j'})$:

$$\odot_{e_i^{i+k}} = \frac{1}{l+1} \sum_{j'=j}^{j+l} \odot_{e_i^{i+k}}(f_{j'}) \tag{6}$$

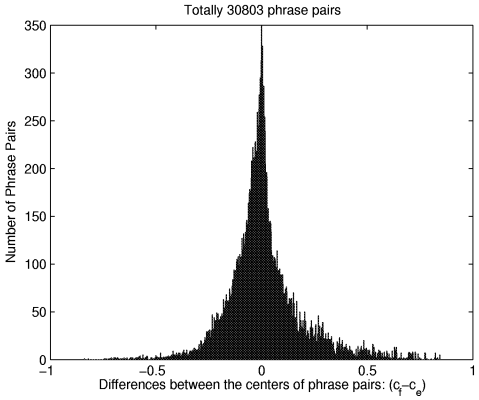


Figure 2: Histogram of relative centers' differences between oracle phrase pairs (blocks) extracted from 627 human word-aligned parallel sentence pairs.

Given the estimated centers of $\odot_{f_j^{j+l}}$ and $\odot_{e_i^{i+k}}$, we can compute how close they are by the probability of $P(\odot_{f_j^{j+l}}|\odot_{e_i^{i+k}})$. To estimate $P(\odot_{f_j^{j+l}}|\odot_{e_i^{i+k}})$, one can start with a flat gaussian model to enforce the point of $(\odot_{e_i^{i+k}}, \odot_{f_j^{j+l}})$ not too far off the diagonal and build an initial list of phrase pairs and then compute the histogram to approximate $P(\odot_{f_j^{j+l}}|\odot_{e_i^{i+k}}) \simeq P(\odot_{f_j^{j+l}} - \odot_{e_i^{i+k}})$. As shown in Figure 2, this probability can be approximated as a gaussian distribution.

### 3.3 A Phrase-Level Lexicon Model

Similar to IBM Model-1 (Brown et al., 1993), we use a bag-of-word generative model within the block. According to Bayes rule:

$$P(f_j^{j+l}|e_i^{i+k}) = \prod_{j' \in [j,j+l]} \sum_{i' \in [i,i+k]} P(f_{j'}|e_{i'}) P(e_{i'}|e_i^{i+k}), \tag{7}$$

where $P(e_{i'}|e_i^{i+k}) \simeq 1/(k+1)$ is approximated by a unigram bag-of-word language model. Because phrase-pairs are usually very short, this assumption works very well in practice (Koehn et al., 2003; Brown et al., 1993).

The parameters used by the three feature functions are estimated using IBM Model-4.

## 4 Two Generative Models

We here present two generative models of $Pr(f_j^{j+l}|e_i^{i+k})$ for phrase-pair extraction. The first one models three aspects of a phrase pair, i.e., the phrase level lexical translation equivalence, positions' distortion and length relevance. The second one takes into consideration of bracketing a sentence pair at a block level and generate sub-blocks synchronously.

### 4.1 Generative Model-1

To model $Pr(f_j^{j+l}|e_i^{i+k})$, *Model-1* first proposes how many words of $f_j$ to generate according to $P(l+1|e_i^{i+k})$; then it proposes the location of the source phrases $f_j^{j+l}$ by predicting the center $\odot_{f_j^{j+l}}$ of the phrase; and then the model generates the words $f_j$ according to a lexicon model $P(f|e)$. The model is summarized in Eqn. 8.

$$Pr(f_j^{j+l}|e_i^{i+k}) = \max_{\{f_j^{j+l}\}} P(l+1|e_i^{i+k}) \cdot P(\odot_{f_j^{j+l}}|\odot_{e_i^{i+k}}) P(f_j^{j+l}|e_i^{i+k}), \quad (8)$$

where the three components $P(l+1|e_i^{i+k})$, $P(\odot_{f_j^{j+l}}|\odot_{e_i^{i+k}})$ and $P(f_j^{j+l}|e_i^{i+k})$ control three different aspects of a bilingual phrase pair: phrase level fertility, center distortion and translation equivalence as explained in the previous section. A variation of this model is applied in a ACL05 shared task for phrase-based statistical machine translation (Zhao and Vogel, 2005).

### 4.2 Generative Model-2: Bracketing

Instead of modeling the three aspects individually as in Model-1, we propose to model the brackets induced by the segmentation of the parallel sentence pair given a block. Shown in Figure 3, a phrase pair (block $A$) split the sentence pairs into five shaded parts $A, B, C, D, E$, which are the valid parts to be generated in our Model-2.

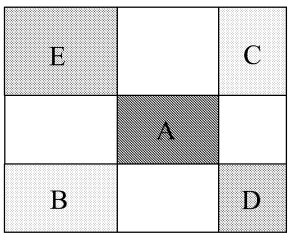


Figure 3: Ngram Bracketing Model

We enforce the following bracketing constraints so that one can only bracket the sentence pair in the following two ways:

$$\begin{aligned} \delta^{[]}(\mathbf{f},\mathbf{e}) &\rightarrow [B, A, C] \\ \delta^{<>}(\mathbf{f},\mathbf{e}) &\rightarrow <D, A, E>, \end{aligned} \quad (9)$$

where $\delta^{[]}$ indicates the bracketing along the diagonal and $\delta^{<>}$ indicates the inverse bracketing. Each bracketing direction is associated with a probability under the same assumption of "bag-of-words" generation as in Eqn. 7. In a way, this model relates to the bilingual bracketing (Wu, 1997) as it requires the other two brackets (either $(B, C)$ or $(D, E)$) to be generated synchronously. However, the model is a flat one because it requires only one level bracketing for any given block $A$. The model is summarized as follows:

$$Pr(f_j^{j+l}, e_i^{i+k}|\mathbf{e},\mathbf{f}) = \max_{\{\delta^{[]},\delta^{<>}\}} P(\delta|\mathbf{e},\mathbf{f}), \quad (10)$$

where $P(\delta^{[]}|\mathbf{e},\mathbf{f}) \simeq P(A)P(B)P(C)$; and $P(A)$, $P(B)$ and $P(C)$ are defined similarly as in Eqn. 7 using the lexicon of $P(f|e)$.

The parameters $P(f|e)$ are from IBM Model-4. Using the constraints from the bracketing in Eqn. 9, this model gives quite good performance shown in our experiments.

### 4.3 A Local Search for Blocks

Both Eqn. 8 and Eqn. 10 involve some local search within a sentence pair: given a source phrase $f_j^{j+l}$, search for the candidate phrase translation $e_i^{i+k}$ in the target sentence according to the score computed in Eqn. 8 or Eqn. 10. In practice, only the phrase pairs to be used in decoding are searched, i.e., we only search phrase pairs in which the source phrase $f_j^{j+l}$ is seen in a pre-defined list to save computations and disk

space. With Eqn. 8 or Eqn. 10 as the performance measures, we employ a twisted stochastic hill-climbing, in which several down-hill moves are accepted to allow one or two left or right functional words to be attached to $e_i^{i+k}$.

To make the local search more effective, we normalize the lexicon model $P(f|e)$ within the parallel sentence pair in Eqn. 11

$$\hat{P}(f|e) = \frac{P(f|e)}{\sum_{j'=1}^{J} P(f_{j'}|e)}. \quad (11)$$

In this way, the distribution of $\hat{P}(f|e)$ is sharper and more focused in the context of a sentence pair.

We'll use the aforementioned two generative models as our baselines. The sub models are generalized and extended as feature functions in our proposed log-linear model for phrase-pair extraction in Section 5.

## 5 A Log-Linear Model

The phrase level fertility model (3.1), distortion model(3.2), lexicon model(3.3) and the bracketing model in 4.2 are all real-valued and bounded ($\in [0, 1]$). We define a log-linear model to combine these sub-models in Eqn. 12:

$$Pr(X|\mathbf{e}, \mathbf{f}) = \frac{exp(\sum_{m=1}^{M} \lambda_m \phi_m(X, \mathbf{e}, \mathbf{f}))}{\sum_{\{X'\}} exp(\sum_{m=1}^{M} \lambda_m \phi_m(X', \mathbf{e}, \mathbf{f}))} \quad (12)$$

where $\phi_m(X, \mathbf{e}, \mathbf{f})$ is a feature function corresponding to the *log probabilities* (i.e. raw scores) from the models listed above. The parameters are the feature functions' weights $\{\lambda_m\}$.

### 5.1 Feature Extensions with Brackets

We define three base feature functions *E2FFScoreIn*: $P(l+1|e_i^{i+k})$; *E2FIBMScoreIN*: $P(f_j^{j+l}|e_i^{i+k})$; and *E2FIBMBracket*: $Pr(X|\mathbf{e}, \mathbf{f})$ as refereed in Section 4.

We then extend the base feature functions by considering the remaining part of the sentence pair excluding the block. This means, the region exclude block $A$ in Figure 3. The motivation is if the block is of high quality, the remaining part should also be explained well by the model. Therefore, we add the following three extended feature functions:

- *E2FFScoreOut*: $P(J-l-1|e_{i' \notin [i,i+k]})$ which estimates how well the remaining English words $e_{i' \notin [i,i+k]}$ can generate the remaining sentence length of $(J-l-1)$. This model can be computed similarly via dynamic programming as in 3.

- *E2FIBMScoreOut*: Generating the remaining French words in the sentence pair:

  $$P(f_{j' \notin [j,j+l]}|e_{i' \notin [i,i+k]}) = \prod_{j' \notin [j,j+l]} \sum_{i' \notin [i,i+k]} P(f_{j'}|e_{i'}) P(e_{i'}|e_{i' \notin [i,i+k]}).$$

  This estimates how well the translational equivalence are kept in accordance with the philosophy of the phrase extraction from a parallel sentence pair.

- *AlignmentLinks*: Averaged word alignment links per source word. We count how many alignment links within the block and normalize this number by the length of the source phrase.

### 5.2 Feature Extensions via Symmetrization

As introduced in Eqn. 1, so far all our models' parameters are using one direction noisy-channel model. In practice, we train both directions of IBM Model-4 — source-to-target and target-to-source to further extend our base feature functions. In this way, we obtain the lexicon models $P(f|e)$ and $P(e|f)$, the fertility models $P(\phi|e)$ and $P(\phi|f)$ to compute the feature functions defined in Section 5.1. Therefore, we have additional five more feature functions of *F2EFScoreIn*, *F2EFScoreOut*, *F2EIBMScoreIN*, *F2EIBMScoreOut* and *F2EIBMBracket* defined accordingly as in Section 5.1. Therefore, we have in total 11 real-valued feature functions for bilingual phrase-pair extraction. Except the feature function of AlignmentLinks, the other 10 feature functions are all bounded in the range of $[0, 1]$.

### 5.3 Learning and Inference

Using direct maximum entropy model for statistical machine translation was explored in (Papineni et al., 1998). To learn the log-linear model in Eqn. 12, a maximum bleu score optimizer implemented in (Koehn, 2004) is modified with a

sampling of N-Best list phrase pairs generated by an initial assignment of weights. To optimize the weights, we view each extracted phrase-pair as a hypothesis block and the reference blocks are extracted from the human word-aligned sentence pairs as described in Section 3.2. We compute word-level F-measure for each extracted block according to all the reference blocks, which contain the same extracted source phrase. Therefore, the data point for optimization is $M$ raw scores [2] of feature functions together with a performance indicator of word-level F-measure. Finally, a modified optimizer similar to (Och and Ney, 2002) is utilized to obtain the optimized weights for the proposed feature functions.

The inference is similar to the one in Section 4.3: a hill-climbing with a performance measure to score the phrase pairs $(f_j^{j+l}, e_i^{i+k})$ according to the log-linear model as in Eqn 13:

$$\hat{X} = \arg\max_{\{X\}} \sum_{m=1}^{M} \lambda_m \phi_m(X, \mathbf{e}, \mathbf{f}), \qquad (13)$$

where $\phi_m(X, \mathbf{e}, \mathbf{f})$ are log probabilities computed using the models from Section 3 and their extensions in Section 5.1 and 5.2.

## 6 Experimental Results

We evaluate our proposed phrase extraction models under the conditions of TIDES'03 Chinese-English statistical machine translation Small-Data track. A small $10K$ LDC bilingual dictionary is allowed to be used and the entries in the dictionary are used as additional phrase pairs. Table 1 summarizes the data statistics.

| | | English | Chinese |
|---|---|---|---|
| Train | Sent. Pairs | 4172 | |
| | Words | 133598 | 105331 |
| | Voc Size | 8359 | 7984 |
| Test | Sent. Pairs | 919 | |
| | Words | - | 26232 |
| | Voc Size | - | 4947 |

Table 1: Training and Test data statistics

Using GIZA++(Och and Ney, 2003), we trained the IBM-Model-4 in both directions for the fertility tables, lexicons and word alignments. We then refine the word alignment by growing the intersections from two directions with new unaligned word pairs which occur in the union (Koehn et al., 2003). We report NIST and *Bleu* (Papineni et al., 2002) scores as the translation performance measures using a decoder in (Vogel et al., 2003). The trigram language model is trained using Gigaword Xinhua news part.

[2]The log probabilities from sub-models

### 6.1 Baseline Generative Models

The two generative models Eqn. 8 and Eqn. 10 give strong baselines. In the local search of phrase pairs in Eqn. 8 and 10, the *Top-N* scored target phrase candidates for each source phrase within the sentence pair are collected for decoding. There is no significant difference observed between the two generative models. Figure 4 shows the performance of Model-2 over Top-N configurations, in which the Top-7 gives the best performance. However, Top-4 configuration already gives performance close to the optimal and we can avoid extracting too many noisy phrase-pairs which significantly slow down the decoding process. In the experiments so forth, we'll use up to Top-4 candidates for phrase-pair extraction.

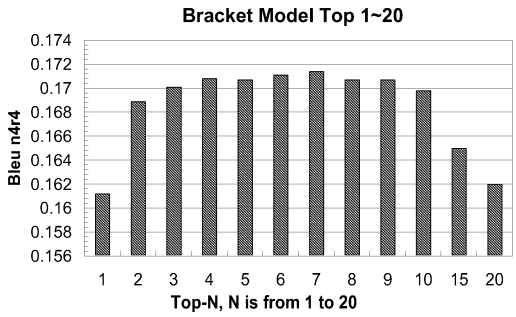


Figure 4: Baseline of Generative Model-2: Bracketing Models

Table 2 summarizes the baselines from the two generative models. The best baseline model is Model-1 using the Top-2 configuration for phrase-pair extraction.

### 6.2 Pair-wise Correlations among Features

The pair-wise correlations among the 11 ($M$=11) real-valued feature functions are investigated. The $M \times M$ correlation matrix is obtained by

| | Mode-1 | | Model-2 | |
|---|---|---|---|---|
| Top-N | Nist | Bleu | Nist | Bleu |
| Top-1 | 6.4976 | 0.1597 | 6.3618 | 0.1611 |
| Top-2 | 6.6749 | **0.1726** | 6.5385 | 0.1688 |
| Top-3 | 6.5934 | 0.1687 | 6.5364 | 0.1700 |
| Top-4 | 6.5752 | 0.1701 | 6.5243 | 0.1707 |

Table 2: Baselines of two generative models

computing the pairwise linear correlation coefficient between the feature functions using the phrase-pairs extracted from Model-2. The feature functions which are highly correlated are regrouped close to each other via standard $K$-means. The result is shown in Figure 5 and the clusters are shown in Table 3.

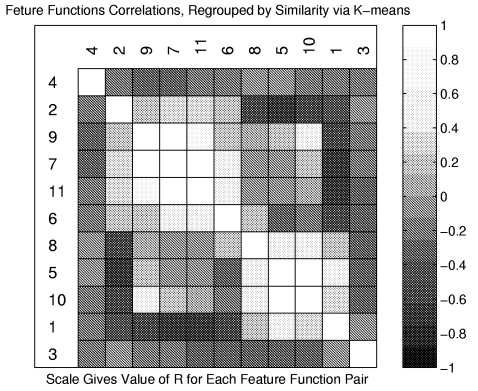


Figure 5: Pair-wise correlations among 11 Feature Functions.

| Feature Func. | FID | Feature Func. | FID |
|---|---|---|---|
| E2FFScoreIn | 2 | E2FIBMBracket | 9 |
| E2FIBMScoreIN | 7 | AlignmentLinks | 11 |
| *F2EIBMScoreOut* | 6 | | |
| F2EFScoreIn | 1 | *E2FIBMScoreOut* | 8 |
| F2EIBMScoreIN | 5 | F2EIBMBracket | 10 |
| E2FFScoreOut | 4 | | |
| F2EFScoreOut | 3 | | |

Table 3: Clustered Feature Functions with *FID*

The more correlated the two feature functions are, the more overlapping information the two share with. In Figure 5, intuitively, features of FID $2, 9, 7, 11, 6$ are grouped together with aspects using the noisy channel model in English-to-Chinese direction and FID of $8, 5, 10, 1$ are clustered together related to the Chinese-to-English direction. The feature functions for the outer brackets like F2EIBMScoreOut (FID 6) and E2FIBMScoreOut (FID 8) are somewhat misplaced and the the fertility models for outside parts F2EFScoreOut and E2FFScoreOut are singled out in the clustering process. These evidences illustrate that feature functions from the outer part of the block have little overlap with the feature functions from the inner part of the block. Two directions of the noisy channel model are weakly correlated. All these observations confirm our intuitions.

### 6.3 Log-Linear Model

We hold 627 sentence pairs with one reference each from the 2001 Dry-Run test as development dataset, which were word-aligned by bilingual speakers. From the word alignment, we can extract the gold-standard blocks (phrase pairs). On the same held-out 627 sentence pairs, we run our baseline models with up to Top-20 target phrase candidates for each source phrase to collect the raw scores from all feature functions and we compute word-level F-measure for each extracted block.

| Datasets | Size |
|---|---|
| Word alignment Links | 14769 |
| Gold Blocks | 30803 |
| Hypothesis Blocks | 297,829 |

Table 4: Development Dataset Statistics

The final optimized weights learnt are (0.0480, 0.0660, 0.0048, 0.0032, 0.1376, 0.1332, 0.1662, 0.1113, 0.2495, 0.4621, 0.0753), in the order of the FIDs assigned in Table 3. Table 5 summarizes the log-linear model's performances at different configurations. The best Bleu score for the log-linear model is 0.1834, an improvement over the best generative models' performance of 0.1726.

The log-linear model based on maximum entropy principle has several advantages over the generative models. It introduces less data fragmentation, requiring fewer independence assumptions and exploiting a principled technique for automatic feature weighting. However, a drawback of our approach is we have to simu-

| Model Settings | | Nist | Bleu |
|---|---|---|---|
| Log-Linear Model | Top1 | 6.8069 | 0.1790 |
| | Top2 | 6.9517 | 0.1811 |
| | Top3 | **6.9620** | **0.1834** |
| | Top4 | 6.8632 | 0.1790 |

Table 5: Log-Linear Model with $M$=11 Feature Functions for Phrase-Pair Extraction.

late the phrase-pair extraction performance measure from the hand-aligned data set to compute the word-level F-measure. This potentially introduces some errors before the optimizations.

## 7 Conclusions and Discussions

We presented a set of informative feature functions for bilingual phrase-pair extractions. A log-linear model is proposed to combine these feature functions and improvements are demonstrated in Chinese-English TIDES'03 small data track evaluations. The log-linear model is a promising framework which has the advantage of leveraging the overlapping features and it has flexibilities of exploiting more informative feature functions. However, the optimization took long to converge and run random restarts were needed to avoid local optima. Different optimization criteria, efficient algorithms and better feature functions can potentially bring more improvements.

## Acknowledgements

We thank Jaime Carbonell and Chiori Hori for their valuable comments and Philipp Koehn for his help with the Pharaoh software package; in particular, the optimization toolkit.

## References

P.F. Brown, Stephen A. Della Pietra, Vincent. J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 304–311, Philadelphia, PA, July 6-7.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based machine translation. In *Proc. of HLT-NAACL 2003*, pages 48–54, Edmonton, Canada, May-June.

Philip Koehn. 2004. Pharaoh: a beam search decoder for phrase-based smt. In *Proceedings of the Conference of the Association for Machine Translation in the Americans (AMTA)*.

Franz J. Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of ACL*, pages 440–447.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29, pages 19–51.

Franz J. Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. In *Computational Linguistics*, volume 30, pages 417–449.

Kishore Papineni, Salim Roukos, and Todd Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech & Signal Processing*, volume 1, pages 189–192, Seattle, May.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Conf. of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, July.

Christoph Tillmann. 2003. A projection extension algorithm for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The cmu statistical machine translation system. In *Proc. of MT SUMMIT IX*, pages 257–264, New Orleans, LA, September.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. In *Computational Linguistics*, volume 23(3), pages 377–403.

R. Zens and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of the Human Language Technology Conference (HLT-NAACL)s*, pages 257–264, Boston, MA, May.

Bing Zhao and Stephan Vogel. 2005. A generalized alignment-free phrase extraction. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 141–144, Ann Arbor, Michigan, June. Association for Computational Linguistics.