Phrase Table Training For Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair?

Yonggang Deng[*] , **Jia Xu**[+] *and* **Yuqing Gao**[*]
[*]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
{ydeng,yuqing}@us.ibm.com
[+]Chair of Computer Science VI, RWTH Aachen University, D-52056 Aachen, Germany
xujia@cs.rwth-aachen.de

Abstract

In this work, the problem of extracting phrase translation is formulated as an information retrieval process implemented with a log-linear model aiming for a balanced precision and recall. We present a generic phrase training algorithm which is parameterized with feature functions and can be optimized jointly with the translation engine to directly maximize the end-to-end system performance. Multiple data-driven feature functions are proposed to capture the quality and confidence of phrases and phrase pairs. Experimental results demonstrate consistent and significant improvement over the widely used method that is based on word alignment matrix only.

1 Introduction

Phrase has become the standard basic translation unit in Statistical Machine Translation (SMT) since it naturally captures context dependency and models internal word reordering. In a phrase-based SMT system, the phrase translation table is the defining component which specifies alternative translations and their probabilities for a given source phrase. In learning such a table from parallel corpus, two related issues need to be addressed (either separately or jointly): which pairs are considered valid translations and how to assign weights, such as probabilities, to them. The first problem is referred to as phrase pair extraction, which identifies phrase pairs that are supposed to be translations of each other. Methods have been proposed, based on syntax, that take advantage of linguistic constraints and alignment of grammatical structure, such as in Yamada and Knight (2001) and Wu (1995). The most widely used approach derives phrase pairs from word alignment matrix (Och and Ney, 2003; Koehn et al., 2003). Other methods do not depend on word alignments only, such as directly modeling phrase alignment in a joint generative way (Marcu and Wong, 2002), pursuing information extraction perspective (Venugopal et al., 2003), or augmenting with model-based phrase pair posterior (Deng and Byrne, 2005).

Using relative frequency as translation probability is a common practice to measure goodness of a phrase pair. Since most phrases appear only a few times in training data, a phrase pair translation is also evaluated by lexical weights (Koehn et al., 2003) or term weighting (Zhao et al., 2004) as additional features to avoid overestimation. The translation probability can also be discriminatively trained such as in Tillmann and Zhang (2006).

The focus of this paper is the phrase pair extraction problem. As in information retrieval, precision and recall issues need to be addressed with a right balance for building a phrase translation table. High precision requires that identified translation candidates are accurate, while high recall wants as much valid phrase pairs as possible to be extracted, which is important and necessary for online translation that requires coverage. In the word-alignment derived phrase extraction approach, precision can be improved by filtering out most of the entries by using a statistical significance test (Johnson et al., 2007). On the other hand, there are valid translation pairs in the training corpus that are not learned due to word alignment errors as shown in Deng and Byrne (2005).

Proceedings of ACL-08: HLT, pages 81–88,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

We would like to improve phrase translation accuracy and at the same time extract as many as possible valid phrase pairs that are missed due to incorrect word alignments. One approach is to leverage underlying word alignment quality such as in Ayan and Dorr (2006). In this work, we present a generic discriminative phrase pair extraction framework that can integrate multiple features aiming to identify correct phrase translation candidates. A significant deviation from most other approaches is that the framework is parameterized and can be optimized jointly with the decoder to maximize translation performance on a development set. Within the general framework, the main work is on investigating useful metrics. We employ features based on word alignment models and alignment matrix. We also propose information metrics that are derived from both bilingual and monolingual perspectives. All these features are data-driven and independent of languages. The proposed phrase extraction framework is general to apply linguistic features such as semantic, POS tags and syntactic dependency.

2 A Generic Phrase Training Procedure

Let $\mathbf{e} = e_1^I$ denote an English sentence and let $\mathbf{f} = f_1^J$ denote its translation in a foreign language, say Chinese. Phrase extraction begins with sentence-aligned parallel corpora $\{(\mathbf{e}_i, \mathbf{f}_i)\}$. We use $E = e_{i_b}^{i_e}$ and $F = f_{j_b}^{j_e}$ to denote an English and foreign phrases respectively, where $i_b(j_b)$ is the position in the sentence of the beginning word of the English(foreign) phrase and $i_e(j_e)$ is the position of the ending word of the phrase.

We first train word alignment models and will use them to evaluate the goodness of a phrase and a phrase pair. Let $f_k(E,F), k = 1, 2, \cdots, K$ be K feature functions to be used to measure the quality of a given phrase pair (E,F). The generic phrase extraction procedure is an evaluation, ranking, filtering, estimation and tuning process, presented in Algorithm 1.

Step 1 (line 1) is the preparation stage. Beginning with a flat lexicon, we train IBM Model-1 word alignment model with 10 iterations for each translation direction. We then train HMM word alignment models (Vogel et al., 1996) in two directions simultaneously by merging statistics collected in the E-step from two directions motivated by Zens et al. (2004) with 5 iterations. We use these models to define the feature functions of candidate phrase pairs such as phrase pair posterior distribution. More details will be given in Section 3.

Algorithm 1 A Generic Phrase Training Procedure

```
 1: Train Model-1 and HMM word alignment models
 2: for all sentence pair $(\mathbf{e}, \mathbf{f})$ do
 3:    Identify candidate phrases on each side
 4:    for all candidate phrase pair $(E, F)$ do
 5:       Calculate its feature function values $f_k$
 6:       Obtain the score $q(E,F) = \sum_{k=1}^{K} \lambda_k f_k(E,F)$
 7:    end for
 8:    Sort candidate phrase pairs by their final scores $q$
 9:    Find the maximum score $qm = \max q(E,F)$
10:    for all candidate phrase pair $(E, F)$ do
11:       If $q(E,F) \geq qm - \tau$, dump the pair into the pool
12:    end for
13: end for
14: Built a phrase translation table from the phrase pair pool
15: Discriminatively train feature weights $\lambda_k$ and threshold $\tau$
```

Step 2 (line 2) consists of phrase pair evaluation, ranking and filtering. Usually all n-grams up to a pre-defined length limit are considered as candidate phrases. This is also the place where linguistic constraints can be applied, say to avoid non-compositional phrases (Lin, 1999). Each normalized feature score derived from word alignment models or language models will be log-linearly combined to generate the final score. Phrase pair filtering is simply thresholding on the final score by comparing to the maximum within the sentence pair. Note that under the log-linear model, applying threshold for filtering is equivalent to comparing the "likelihood" ratio.

Step 3 (line 14) pools all candidate phrase pairs that pass the threshold testing and estimates the final phrase translation table by maximum likelihood criterion. For each candidate phrase pair which is above the threshold, we assign HMM-based phrase pair posterior as its soft count when dumping them into the global phrase pair pool. Other possibilities for the weighting include assigning constant one or the exponential of the final score etc.

One of the advantages of the proposed phrase training algorithm is that it is a parameterized procedure that can be optimized jointly with the translation engine to minimize the final translation errors measured by automatic metrics such as BLEU (Papineni et al., 2002). In the final step 4 (line 15), parameters $\{\lambda_k, \tau\}$ are discriminatively trained on a development set using the downhill simplex method (Nelder and Mead, 1965).

This phrase training procedure is general in the sense that it is configurable and trainable with different feature functions and their parameters. The commonly used phrase extraction approach based on word alignment heuristics (referred as *ViterbiExtract* algorithm for comparison in this paper) as described in (Och, 2002; Koehn et al., 2003) is a special case of the algorithm, where candidate phrase pairs are restricted to those that respect word alignment boundaries.

We rely on multiple feature functions that aim to describe the quality of candidate phrase translations and the generic procedure to figure out the best way of combining these features. A good feature function pops up valid translation pairs and pushes down incorrect ones.

3 Features

Now we present several feature functions that we investigated to help extracting correct phrase translations. All these features are data-driven and defined based on models, such as statistical word alignment model or language model.

3.1 Model-based Phrase Pair Posterior

In a statistical generative word alignment model (Brown et al., 1993), it is assumed that (i) a random variable $\mathbf{a}$ specifies how each target word f_j is generated by (therefore aligned to) a source [1] word e_{a_j}; and (ii) the likelihood function $f(\mathbf{f}, \mathbf{a}|\mathbf{e})$ specifies a generative procedure from the source sentence to the target sentence. Given a phrase pair in a sentence pair, there will be many generative paths that align the source phrase to the target phrase. The likelihood of those generative procedures can be accumulated to get the likelihood of the phrase pair (Deng and Byrne, 2005). This is implemented as the summation of the likelihood function over all valid hidden word alignments.

More specifically, let $A_{(i_1,i_2)}^{(j_1,j_2)}$ be the set of word alignment $\mathbf{a}$ that aligns the source phrase $e_{i_1}^{j_1}$ to the target phrase $f_{j_1}^{j_2}$ (links to NULL word are ignored for simplicity):

$$A_{(i_1,i_2)}^{(j_1,j_2)} = \{\mathbf{a} : a_j \in [i_1, i_2] \text{ iff } j \in [j_1, j_2]\}$$

The alignment set given a phrase pair ignores those pairs with word links across the phrase boundary. Consequently, the phrase-pair posterior distribution is defined as

$$P_\theta(e_{i_1}^{i_2} \rightarrow f_{j_1}^{j_2}|\mathbf{e}, \mathbf{f}) \quad = \quad \frac{\sum_{\mathbf{a} \in A_{(i_1,i_2)}^{(j_1,j_2)}} f(\mathbf{a}, \mathbf{f}|\mathbf{e}; \theta)}{\sum_{\mathbf{a}} f(\mathbf{a}, \mathbf{f}|\mathbf{e}; \theta)} \quad (1)$$

Switching the source and the target, we can obtain the posterior distribution in another translation direction. This distribution is applicable to all word alignment models that follow assumptions (i) and (ii). However, the complexity of the likelihood function could make it impractical to calculate the summations in Equation 1 unless an approximation is applied.

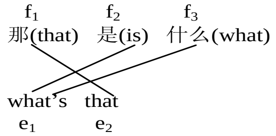
Several feature functions will be defined on top of the posterior distribution. One of them is based on HMM word alignment model. We use the geometric mean of posteriors in two translation directions as a symmetric metric for phrase pair quality evaluation function under HMM alignment models. Table 1 shows the phrase pair posterior matrix of the example.

Replacing the word alignment model with IBM Model-1 is another feature function that we added. IBM Model-1 is simple yet has been shown to be effective in many applications (Och et al., 2004). There is a close form solution to calculate the phrase pair posterior under Model-1. Moreover, word to word translation table under HMM is more concentrated than that under Model-1. Therefore, the posterior distribution evaluated by Model-1 is smoother and potentially it can alleviate the overestimation problem in HMM especially when training data size is small.

3.2 Bilingual Information Metric

Trying to find phrase translations for any possible n-gram is not a good idea for two reasons. First, due to data sparsity and/or alignment model's capability, there would exist n-grams that cannot be aligned

[1] The word *source* and *target* are in the sense of word alignment direction, not as in the source-channel formulation.



	e_1^1	e_1^2	e_2^2	$H_{BL}(f_{j_1}^{j_2})$
f_1^1	0.0006	0.012	0.89	0.08
f_1^2	0.0017	0.035	0.343	0.34
f_1^3	0.07	0.999	0.0004	0.24
f_2^2	0.03	0.0001	0.029	0.7
f_2^3	0.89	0.006	0.006	0.05
f_3^3	0.343	0.002	0.002	0.06
$H_{BL}(e_{i_1}^{i_2})$	0.869	0.26	0.70	

Table 1: Phrase pair posterior distribution for the example

well, for instance, n-grams that are part of a paraphrase translation or metaphorical expression. To give an example, the unigram 'tomorrow' in 'the day after tomorrow' whose Chinese translation is a single word '后天'. Extracting candidate translations for such kind of n-grams for the sake of improving coverage (recall) might hurt translation quality (precision). We will define a confidence metric to estimate how reliably the model can align an n-gram in one side to a phrase on the other side given a parallel sentence. Second, some n-grams themselves carry no linguistic meaning; their phrase translations can be misleading, for example non-compositional phrases (Lin, 1999). We will address this in section 3.3.

Given a sentence pair, the basic assumption is that if the HMM word alignment model can align an English phrase well to a foreign phrase, the posterior distribution of the English phrase generating all foreign phrases on the other side is significantly biased. For instance, the posterior of one foreign phrase is far larger than that of the others. We use the entropy of the posterior distribution as the confidence metric:

$$H_{BL}(e_{i_1}^{i_2}|\mathbf{e}, \mathbf{f}) = H(\hat{P}_{\theta_{HMM}}(e_{i_1}^{i_2} \rightarrow *)) \qquad (2)$$

where $H(P) = -\sum_x P(x) \log P(x)$ is the entropy of a distribution $P(x)$, $\hat{P}_{\theta_{HMM}}(e_{i_1}^{i_2} \rightarrow *)$ is the normalized probability (sum up to 1) of the posterior $P_{\theta_{HMM}}(e_{i_1}^{i_2} \rightarrow *)$ as defined in Equation 1. Low entropy signals a high confidence that the English phrase can be aligned correctly. On the other hand, high entropy implies ambiguity presented in discriminating the correct foreign phrase from the others from the viewpoint of the model.

Similarly we calculate the confidence metric of aligning a foreign phrase correctly with the word alignment model in foreign to English direction. Table 1 shows the entropy of phrases. The unigram of foreign side f_2^2 is unlikely to survive with such high ambiguity. Adding the entropy in two directions defines the bilingual information metric as another feature function, which describes the reliability of aligning each phrase correctly by the model. Note that we used HMM word alignment model to find the posterior distribution. Other models such as Model-1 can be applied in the same way. This feature function quantitatively captures the goodness of phrases. During phrase pair ranking, it can help to move upward phrases that can be aligned well and push downward phrases that are difficult for the model to find correct translations.

3.3 Monolingual Information Metric

Now we turn to monolingual resources to evaluate the quality of an n-gram being a good phrase. A phrase in a sentence is specified by its boundaries. We assume that the boundaries of a good phrase should be the "right" place to break. More generally, we want to quantify how effective a word boundary is as a phrase boundary. One would perform say NP-chunking or parsing to avoid splitting a linguistic constituent. We apply a language model (LM) to describe the predictive uncertainty (PU) between words in two directions.

Given a history w_1^{n-1}, a language model specifies a conditional distribution of the future word being predicted to follow the history. We can find the entropy of such pdf: $H_{LM}(w_1^{n-1}) = H(P(\cdot|w_1^{n-1}))$. So given a sentence w_1^N, the PU of the boundary between word w_i and w_{i+1} is established by two-way entropy sum using a forward and backward language model: $PU(w_1^N, i) = H_{LMF}(w_1^i) + H_{LMB}(w_N^{i+1})$

We assume that the higher the predictive uncertainty is, the more likely the left or right part of the word boundary can be "cut-and-pasted" to form another reasonable sentence. So a good phrase is characterized with high PU values on the boundaries. For example, in 'we want to have a table near the window', the PU value of the point after 'table' is 0.61, higher than that between 'near' and 'the' 0.3, using trigram LMs.

With this, the feature function derived from monolingual clue for a phrase pair can be defined as the product of PUs of the four word boundaries.

3.4 Word Alignments Induced Metric

The widely used ViterbiExtract algorithm relies on word alignment matrix and no-crossing-link assumption to extract phrase translation candidates. Practically it has been proved to work well. However, discarding correct phrase pairs due to incorrect word links leaves room for improving recall. This is especially true for not significantly large training corpora. Provided with a word alignment matrix, we define within phrase pair consistency ratio (WPPCR) as another feature function. WPPCR was used as one of the scores in (Venugopal et al., 2003) for phrase extraction. It is defined as the number of consistent word links associated with any words within the phrase pair divided by the number of all word links associated with any words within the phrase pair. An inconsistent link connects a word within the phrase pair to a word outside the phrase pair. For example, the WPPCR for (e_1^2, f_1^2) in Table 1 is 2/3. As a special case, the ViterbiExtract algorithm extracts only phrase pairs with WPPCR is 1.

To further discriminate the pairs with higher WPPCR from those with lower ratio, we apply a Bi-Linear Transform (BLT) (Oppenheim and Schafer, 1989) mapping. BLT is commonly used in signal processing to attenuate the low frequency parts. When used to map WPPCR, it exaggerates the difference between phrase pairs with high WPPCR and those with low WPPCR, making the pairs with low ratio more unlikely to be selected as translation candidates. One of the nice properties of BLT is that there is a parameter that can be changed to adjust the degree of attenuation, which provides another dimension for system optimization.

4 Experimental Results

We evaluate the effect of the proposed phrase extraction algorithm with translation performance. We do experiments on IWSLT (Paul, 2006) 2006 Chinese-English corpus. The task is to translate Chinese utterances in travel domain into English. We report only text (speech transcription) translation results.

The training corpus consists of 40K Chinese-English parallel sentences in travel domain with total 306K English words and 295K Chinese words. In the data processing step, Chinese characters are segmented into words. English text are normalized and lowercased. All punctuation is removed.

Eval Set	04dev	04test	05test	06dev	06test
# of sentences	506	500	506	489	500
# of words	2808	2906	3209	5214	5550
# of refs	16	16	16	7	7

Table 2: Dev/test set statistics

There are five sets of evaluation sentences in tourism domain for development and test. Their statistics are shown in Table 2. We will tune training and decoding parameters on 06dev and report results on other sets.

4.1 Training and Translation Setup

Our decoder is a phrase-based multi-stack implementation of the log-linear model similar to Pharaoh (Koehn et al., 2003). Like other log-linear model based decoders, active features in our translation engine include translation models in two directions, lexicon weights in two directions, language model, lexicalized distortion models, sentence length penalty and other heuristics. These feature weights are tuned on the dev set to achieve optimal translation performance using downhill simplex method. The language model is a statistical trigram model estimated with Modified Kneser-Ney smoothing (Chen and Goodman, 1996) using only English sentences in the parallel training data.

Starting from the collection of parallel training sentences, we build word alignment models in two translation directions, from English to Chinese and from Chinese to English, and derive two sets of Viterbi alignments. By combining word alignments in two directions using heuristics (Och and Ney, 2003), a single set of static word alignments is then formed. Based on alignment models and word alignment matrices, we compare different approaches of building a phrase translation table and show the final translation results. We measure translation performance by the BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) scores with multiple translation references.

BLEU Scores

Table	04dev	04test	05test	06dev	06test
HMM	0.367	0.407	0.473	0.200	0.190
Model-4	0.380	0.403	0.485	0.210	0.204
New	0.411	0.427	0.500	0.216	0.208

METEOR Scores

Table	04dev	04test	05test	06dev	06test
HMM	0.532	0.586	0.675	0.482	0.471
Model-4	0.540	0.593	0.682	0.492	0.480
New	0.568	0.614	0.691	0.505	0.487

Table 3: Translation Results

4.2 Translation Results

Our baseline phrase table training method is the ViterbiExtract algorithm. All phrase pairs with respect to the word alignment boundary constraint are identified and pooled to build phrase translation tables with the Maximum Likelihood criterion. We prune phrase translation entries by their probabilities. The maximum number of words in Chinese and English phrases is set to 8 and 25 respectively for all conditions[2]. We perform online style phrase training, i.e., phrase extraction is not particular for any evaluation set.

Two different word alignment models are trained as the baseline, one is symmetric HMM word alignment model, the other is IBM Model-4 as implemented in the GIZA++ toolkit (Och and Ney, 2003). The translation results as measured by BLEU and METEOR scores are presented in Table 3. We notice that Model-4 based phrase table performs roughly 1% better in terms of both BLEU and METEOR scores than that based on HMM.

We follow the generic phrase training procedure as described in section 2. The most time consuming part is calculating posteriors, which is carried out in parallel with 30 jobs in less than 1.5 hours.

We use the Viterbi word alignments from HMM to define within phrase pair consistency ratio as discussed in section 3.4. Although Table 3 implies that Model-4 word alignment quality is better than that of HMM, we did not get benefits by switching to Model-4 to compute word alignments based feature values.

In estimating phrase translation probability, we use accumulated HMM-based phrase pair posteriors as their 'soft' frequencies and then the final translation probability is the relative frequency. HMM-based posterior was shown to be better than treating each occurrence as count one.

Once we have computed all feature values for all phrase pairs in the training corpus, we discriminatively train feature weights λ_ks and the threshold τ using the downhill simplex method to maximize the BLEU score on 06dev set. Since the translation engine implements a log-linear model, the discriminative training of feature weights in the decoder should be embedded in the whole end-to-end system jointly with the discriminative phrase table training process. This is globally optimal but computationally demanding. As a compromise, we fix the decoder feature weights and put all efforts on optimizing phrase training parameters to find out the best phrase table.

The translation results with the discriminatively trained phrase table are shown as the row of "New" in Table 3. We observe that the new approach is consistently better than the baseline ViterbiExtract algorithm with either Model-4 or HMM word alignments on all sets. Roughly, it has 0.5% higher BLEU score on 2006 sets and 1.5% to 3% higher on other sets than Model-4 based ViterbiExtract method. Similar superior results are observed when measured with METEOR score.

5 Discussions

The generic phrase training algorithm follows an information retrieval perspective as in (Venugopal et al., 2003) but aims to improve both precision and recall with the trainable log-linear model. A clear advantage of the proposed approach over the widely used ViterbiExtract method is trainability. Under the general framework, one can put as many features as possible together under the log-linear model to evaluate the quality of a phrase and a phase pair. The phrase table extracting procedure is trainable and can be optimized jointly with the translation engine.

Another advantage is flexibility, which is provided partially by the threshold τ. As the figure 1 shows, when we increase the threshold by allowing more candidate phrase pair hypothesized as valid translation, we observe the phrase table size increases monotonically. On the other hand, we notice

[2] We chose large numbers for phrase length limit to build a strong baseline and to avoid impact of longer phase length.

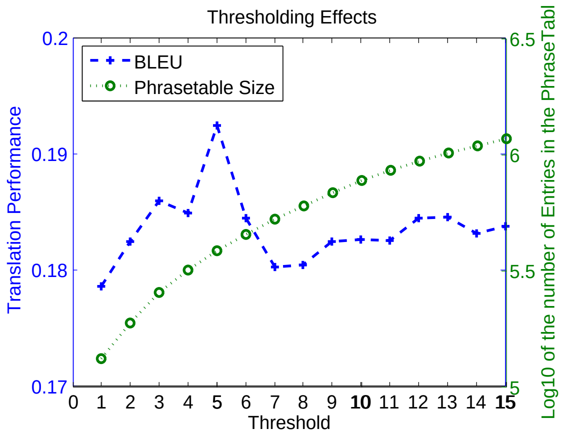


Figure 1: Thresholding effects on translation performance and phrase table size

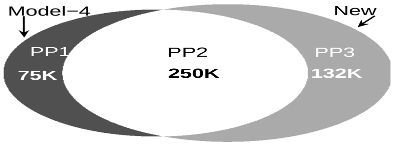
that the translation performance improves gradually. After reaching its peak, the BLEU score drops as the threshold τ increases. When τ is large enough, the translation performance is not changing much but still worse than the peak value. It implies a balancing process between precision and recall. The final optimal threshold τ is around 5.

The flexibility is also enabled by multiple configurable features used to evaluate the quality of a phrase and a phrase pair. Ideally, a perfect combination of feature functions divides the correct and incorrect candidate phrase pairs within a parallel sentence into two ordered separate sets. We use feature functions to decide the order and the threshold τ to locate the boundary guided with a development set.

So the main issue to investigate now is which features are important and valuable in ranking candidate phrase pairs. We propose several information metrics derived from posterior distribution, language model and word alignments as feature functions. The ViterbiExtract is a special case where a single binary feature function defined from word alignments is used. Its good performance (as shown in Table 3) suggests that word alignments are very indicative of phrase pair quality. So we design comparative experiments to capture word alignment impact only. We start with basic features that include model-based posterior, bilingual and monolingual information metrics. Its results on different test sets are presented in the "basic" row of Table 4. We add word alignment feature ("+align" row), and

Features	04dev	04test	05test	06dev	06test
basic	0.393	0.406	0.496	0.205	0.199
+align	0.401	0.429	0.502	0.208	0.196
+align_BLT	0.411	0.427	0.500	0.216	0.208

Table 4: Translation Results (BLEU) of discriminative phrase training approach using different features



Features	04dev	04test	05test	06dev	06test
PP2	0.380	0.395	0.480	0.207	0.202
PP1+PP2	0.380	0.403	0.485	0.210	0.204
PP2+PP3	0.411	0.427	0.500	0.216	0.208
PP1+PP2+PP3	0.412	0.432	0.500	0.217	0.214

Table 5: Translation Results (BLEU) of Different Phrase Pair Combination

then apply bilinear transform to the consistency ratio WPPCR as described in section 3.4 ("+align_BLT" row). The parameter controlling the degree of attenuation in BLT is also optimized together with other feature weights.

With the basic features, the new phrase extraction approach performs better than the baseline method with HMM word alignment models but similar to the baseline method with Model-4. With the word alignment based feature WPPCR, we obtain a 2% improvement on 04test set but not much on other sets except slight degradation on 06test. Finally, applying BLT transform to WPPCR leads to additional 0.8 BLEU point on 06dev set and 1.2 point on 06test set. This confirms the effectiveness of word alignment based features.

Now we compare the phrase table using the proposed method to that extracted using the baseline ViterbiExtract method with Model-4 word alignments. The Venn diagram in Table 5 shows how the two phrase tables overlap with each other and size of each part. As expected, they have a large number of common phrase pairs (PP2). The new method is able to extract more phrase pairs than the baseline with Model-4. PP1 is the set of phrase pairs found by Model-4 alignments. Removing PP1 from the baseline phrase table (comparing the first group of scores) or adding PP1 to the new phrase table (the second group of scores) overall results in no or marginal performance change. On the other hand, adding phrase pairs extracted by the new method only (PP3) can lead to significant BLEU score increases (comparing row 1 vs. 3, and row 2 vs. 4).

6 Conclusions

In this paper, the problem of extracting phrase translation is formulated as an information retrieval process implemented with a log-linear model aiming for a balanced precision and recall. We have presented a generic phrase translation extraction procedure which is parameterized with feature functions. It can be optimized jointly with the translation engine to directly maximize the end-to-end translation performance. Multiple feature functions were investigated. Our experimental results on IWSLT Chinese-English corpus have demonstrated consistent and significant improvement over the widely used word alignment matrix based extraction method. [3]

Acknowledgement We would like to thank Xiaodong Cui, Radu Florian and other IBM colleagues for useful discussions and the anonymous reviewers for their constructive suggestions.

References

N. Ayan and B. Dorr. 2006. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proc. of ACL*, pages 9–16.

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19:263–312.

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

Y. Deng and W. Byrne. 2005. HMM word and phrase alignment for statistical machine translation. In *Proc. of HLT-EMNLP*, pages 169–176.

H. Johnson, J. Martin, G. Foster, and R. Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proc. of EMNLP-CoNLL*, pages 967–975.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 48–54.

D. Lin. 1999. Automatic identification of non-compositional phrases. In *Proc. of ACL*, pages 317–324.

D. Marcu and D. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP*, pages 133–139.

J. A. Nelder and R. Mead. 1965. A simplex method for function minimization. *Computer Journal*, 7:308–313.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och, D. Gildea, and et al. 2004. A smorgasbord of features for statistical machine translation. In *Proc. of HLT-NAACL*, pages 161–168.

F. Och. 2002. *Statistical Machine Translation: From Single Word Models to Alignment Templates*. Ph.D. thesis, RWTH Aachen, Germany.

A. V. Oppenheim and R. W. Schafer. 1989. *Discrete-Time Signal Processing*. Prentice-Hall.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

M. Paul. 2006. Overview of the IWSLT 2006 evaluation campaign. In *Proc. of IWSLT*, pages 1–15.

C. Tillmann and T. Zhang. 2006. A discriminative global training algorithm for statistical MT. In *Proc. of ACL*, pages 721–728.

A. Venugopal, S. Vogel, and A. Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proc. of ACL*, pages 319–326.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM based word alignment in statistical translation. In *Proc. of the COLING*.

D. Wu. 1995. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proc. of ACL*, pages 244–251.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL*, pages 523–530.

R. Zens, E. Matusov, and H. Ney. 2004. Improved word alignment using a symmetric lexicon model. In *Proc. of COLING*, pages 36–42.

B. Zhao, S. Vogel, M. Eck, and A. Waibel. 2004. Phrase pair rescoring with term weighting for statistical machine translation. In *Proc. of EMNLP*, pages 206–213.

[3] By parallelism, we have shown the feasibility and effectiveness (results not presented here) of the proposed method in handling millions of sentence pairs.