Training Phrase Translation Models with Leaving-One-Out

Joern Wuebker and **Arne Mauser** and **Hermann Ney**
Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Germany
<surname>@cs.rwth-aachen.de

Abstract

Several attempts have been made to learn phrase translation probabilities for phrase-based statistical machine translation that go beyond pure counting of phrases in word-aligned training data. Most approaches report problems with over-fitting. We describe a novel leaving-one-out approach to prevent over-fitting that allows us to train phrase models that show improved translation performance on the WMT08 Europarl German-English task. In contrast to most previous work where phrase models were trained separately from other models used in translation, we include all components such as single word lexica and reordering models in training. Using this consistent training of phrase models we are able to achieve improvements of up to 1.4 points in BLEU. As a side effect, the phrase table size is reduced by more than 80%.

1 Introduction

A phrase-based SMT system takes a source sentence and produces a translation by segmenting the sentence into phrases and translating those phrases separately (Koehn et al., 2003). The phrase translation table, which contains the bilingual phrase pairs and the corresponding translation probabilities, is one of the main components of an SMT system. The most common method for obtaining the phrase table is heuristic extraction from automatically word-aligned bilingual training data (Och et al., 1999). In this method, all phrases of the sentence pair that match constraints given by the alignment are extracted. This includes overlapping phrases. At extraction time it does not matter, whether the phrases are extracted from a highly probable phrase alignment or from an unlikely one.

Phrase model probabilities are typically defined as relative frequencies of phrases extracted from word-aligned parallel training data. The joint counts $C(\tilde{f}, \tilde{e})$ of the source phrase $\tilde{f}$ and the target phrase $\tilde{e}$ in the entire training data are normalized by the marginal counts of source and target phrase to obtain a conditional probability

$$p_H(\tilde{f}|\tilde{e}) = \frac{C(\tilde{f}, \tilde{e})}{C(\tilde{e})}. \quad (1)$$

The translation process is implemented as a weighted log-linear combination of several models $h_m(e_1^I, s_1^K, f_1^J)$ including the logarithm of the phrase probability in source-to-target as well as in target-to-source direction. The phrase model is combined with a language model, word lexicon models, word and phrase penalty, and many others. (Och and Ney, 2004) The best translation $\hat{e}_1^{\hat{I}}$ as defined by the models then can be written as

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \quad (2)$$

In this work, we propose to directly train our phrase models by applying a forced alignment procedure where we use the decoder to find a phrase alignment between source and target sentences of the training data and then updating phrase translation probabilities based on this alignment. In contrast to heuristic extraction, the proposed method provides a way of consistently training and using phrase models in translation. We use a modified version of a phrase-based decoder to perform the forced alignment. This way we ensure that all models used in training are identical to the ones used at decoding time. An illustration of the basic

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 475–484,
Uppsala, Sweden, 11-16 July 2010.

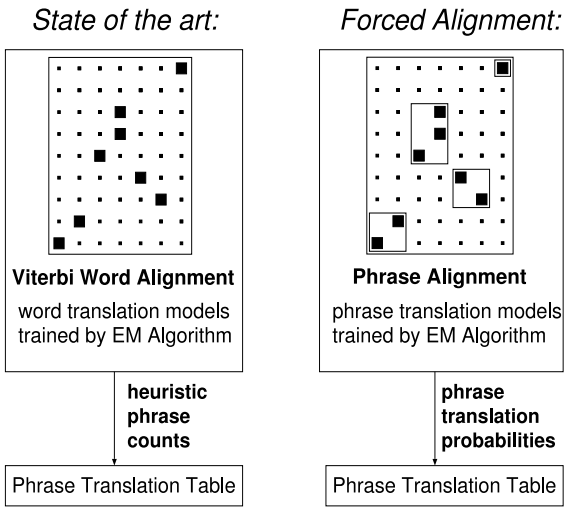

Figure 1: Illustration of phrase training with forced alignment.

idea can be seen in Figure 1. In the literature this method by itself has been shown to be problematic because it suffers from over-fitting (DeNero et al., 2006), (Liang et al., 2006). Since our initial phrases are extracted from the same training data, that we want to align, very long phrases can be found for segmentation. As these long phrases tend to occur in only a few training sentences, the EM algorithm generally overestimates their probability and neglects shorter phrases, which better generalize to unseen data and thus are more useful for translation. In order to counteract these effects, our training procedure applies leaving-one-out on the sentence level. Our results show, that this leads to a better translation quality.

Ideally, we would produce all possible segmentations and alignments during training. However, this has been shown to be infeasible for real-world data (DeNero and Klein, 2008). As training uses a modified version of the translation decoder, it is straightforward to apply pruning as in regular decoding. Additionally, we consider three ways of approximating the full search space:

1. the single-best Viterbi alignment,
2. the n-best alignments,
3. all alignments remaining in the search space after pruning.

The performance of the different approaches is measured and compared on the German-English Europarl task from the *ACL 2008 Workshop on Statistical Machine Translation* (WMT08). Our results show that the proposed phrase model training improves translation quality on the test set by 0.9 BLEU points over our baseline. We find that by interpolation with the heuristically extracted phrases translation performance can reach up to 1.4 BLEU improvement over the baseline on the test set.

After reviewing the related work in the following section, we give a detailed description of phrasal alignment and leaving-one-out in Section 3. Section 4 explains the estimation of phrase models. The empirical evaluation of the different approaches is done in Section 5.

2 Related Work

It has been pointed out in literature, that training phrase models poses some difficulties. For a generative model, (DeNero et al., 2006) gave a detailed analysis of the challenges and arising problems. They introduce a model similar to the one we propose in Section 4.2 and train it with the EM algorithm. Their results show that it can not reach a performance competitive to extracting a phrase table from word alignment by heuristics (Och et al., 1999).

Several reasons are revealed in (DeNero et al., 2006). When given a bilingual sentence pair, we can usually assume there are a number of equally correct phrase segmentations and corresponding alignments. For example, it may be possible to transform one valid segmentation into another by splitting some of its phrases into sub-phrases or by shifting phrase boundaries. This is different from word-based translation models, where a typical assumption is that each target word corresponds to only one source word. As a result of this ambiguity, different segmentations are recruited for different examples during training. That in turn leads to over-fitting which shows in overly determinized estimates of the phrase translation probabilities. In addition, (DeNero et al., 2006) found that the trained phrase table shows a highly peaked distribution in opposition to the more flat distribution resulting from heuristic extraction, leaving the decoder only few translation options at decoding time.

Our work differs from (DeNero et al., 2006) in a number of ways, addressing those problems.

To limit the effects of over-fitting, we apply the leaving-one-out and cross-validation methods in training. In addition, we do not restrict the training to phrases consistent with the word alignment, as was done in (DeNero et al., 2006). This allows us to recover from flawed word alignments.

In (Liang et al., 2006) a discriminative translation system is described. For training of the parameters for the discriminative features they propose a strategy they call *bold updating*. It is similar to our forced alignment training procedure described in Section 3.

For the hierarchical phrase-based approach, (Blunsom et al., 2008) present a discriminative rule model and show the difference between using only the viterbi alignment in training and using the full sum over all possible derivations.

Forced alignment can also be utilized to train a phrase segmentation model, as is shown in (Shen et al., 2008). They report small but consistent improvements by incorporating this segmentation model, which works as an additional prior probability on the monolingual target phrase.

In (Ferrer and Juan, 2009), phrase models are trained by a semi-hidden Markov model. They train a conditional "inverse" phrase model of the target phrase given the source phrase. Additionally to the phrases, they model the segmentation sequence that is used to produce a phrase alignment between the source and the target sentence. They used a phrase length limit of 4 words with longer phrases not resulting in further improvements. To counteract over-fitting, they interpolate the phrase model with IBM Model 1 probabilities that are computed on the phrase level. We also include these word lexica, as they are standard components of the phrase-based system.

It is shown in (Ferrer and Juan, 2009), that Viterbi training produces almost the same results as full Baum-Welch training. They report improvements over a phrase-based model that uses an inverse phrase model and a language model. Experiments are carried out on a custom subset of the English-Spanish Europarl corpus.

Our approach is similar to the one presented in (Ferrer and Juan, 2009) in that we compare Viterbi and a training method based on the Forward-Backward algorithm. But instead of focusing on the statistical model and relaxing the translation task by using monotone translation only, we use a full and competitive translation system as starting point with reordering and all models included.

In (Marcu and Wong, 2002), a joint probability phrase model is presented. The learned phrases are restricted to the most frequent n-grams up to length 6 and all unigrams. Monolingual phrases have to occur at least 5 times to be considered in training. Smoothing is applied to the learned models so that probabilities for rare phrases are non-zero. In training, they use a greedy algorithm to produce the Viterbi phrase alignment and then apply a hill-climbing technique that modifies the Viterbi alignment by merge, move, split, and swap operations to find an alignment with a better probability in each iteration. The model shows improvements in translation quality over the single-word-based IBM Model 4 (Brown et al., 1993) on a subset of the Canadian Hansards corpus.

The joint model by (Marcu and Wong, 2002) is refined by (Birch et al., 2006) who use high-confidence word alignments to constrain the search space in training. They observe that due to several constraints and pruning steps, the trained phrase table is much smaller than the heuristically extracted one, while preserving translation quality.

The work by (DeNero et al., 2008) describes a method to train the joint model described in (Marcu and Wong, 2002) with a Gibbs sampler. They show that by applying a prior distribution over the phrase translation probabilities they can prevent over-fitting. The prior is composed of IBM1 lexical probabilities and a geometric distribution over phrase lengths which penalizes long phrases. The two approaches differ in that we apply the leaving-one-out procedure to avoid over-fitting, as opposed to explicitly defining a prior distribution.

3 Alignment

The training process is divided into three parts. First we obtain all models needed for a normal translations system. We perform minimum error rate training with the downhill simplex algorithm (Nelder and Mead, 1965) on the development data to obtain a set of scaling factors that achieve a good BLEU score. We then use these models and scaling factors to do a forced alignment, where we compute a phrase alignment for the training data. From this alignment we then estimate new phrase models, while keeping all other models un-

changed. In this section we describe our forced alignment procedure that is the basic training procedure for the models proposed here.

3.1 Forced Alignment

The idea of forced alignment is to perform a phrase segmentation and alignment of each sentence pair of the training data using the full translation system as in decoding. What we call segmentation and alignment here corresponds to the "concepts" used by (Marcu and Wong, 2002). We apply our normal phrase-based decoder on the source side of the training data and constrain the translations to the corresponding target sentences from the training data.

Given a source sentence f_1^J and target sentence e_1^I, we search for the best phrase segmentation and alignment that covers both sentences. A segmentation of a sentence into K phrase is defined by

$$k \rightarrow s_k := (i_k, b_k, j_k), \text{ for } k = 1, \ldots, K$$

where for each segment i_k is last position of kth target phrase, and (b_k, j_k) are the start and end positions of the source phrase aligned to the kth target phrase. Consequently, we can modify Equation 2 to define the best segmentation of a sentence pair as:

$$\hat{s}_1^{\hat{K}} = \underset{K, s_1^K}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K, f_1^J) \right\} \quad (3)$$

The identical models as in search are used: conditional phrase probabilities $p(\tilde{f}_k|\tilde{e}_k)$ and $p(\tilde{e}_k|\tilde{f}_k)$, within-phrase lexical probabilities, distance-based reordering model as well as word and phrase penalty. A language model is not used in this case, as the system is constrained to the given target sentence and thus the language model score has no effect on the alignment.

In addition to the phrase matching on the source sentence, we also discard all phrase translation candidates, that do not match any sequence in the given target sentence.

Sentences for which the decoder can not find an alignment are discarded for the phrase model training. In our experiments, this is the case for roughly 5% of the training sentences.

3.2 Leaving-one-out

As was mentioned in Section 2, previous approaches found over-fitting to be a problem in phrase model training. In this section, we describe a leaving-one-out method that can improve the phrase alignment in situations, where the probability of rare phrases and alignments might be overestimated. The training data that consists of N parallel sentence pairs f_n and e_n for $n = 1, \ldots, N$ is used for both the initialization of the translation model $p(\tilde{f}|\tilde{e})$ and the phrase model training. While this way we can make full use of the available data and avoid unknown words during training, it has the drawback that it can lead to over-fitting. All phrases extracted from a specific sentence pair f_n, e_n can be used for the alignment of this sentence pair. This includes longer phrases, which only match in very few sentences in the data. Therefore those long phrases are trained to fit only a few sentence pairs, strongly overestimating their translation probabilities and failing to generalize. In the extreme case, whole sentences will be learned as phrasal translations. The average length of the used phrases is an indicator of this kind of over-fitting, as the number of matching training sentences decreases with increasing phrase length. We can see an example in Figure 2. Without leaving-one-out the sentence is segmented into a few long phrases, which are unlikely to occur in data to be translated. Phrase boundaries seem to be unintuitive and based on some hidden structures. With leaving-one-out the phrases are shorter and therefore better suited for generalization to unseen data.

Previous attempts have dealt with the over-fitting problem by limiting the maximum phrase length (DeNero et al., 2006; Marcu and Wong, 2002) and by smoothing the phrase probabilities by lexical models on the phrase level (Ferrer and Juan, 2009). However, (DeNero et al., 2006) experienced similar over-fitting with short phrases due to the fact that the same word sequence can be segmented in different ways, leading to specific segmentations being learned for specific training sentence pairs. Our results confirm these findings. To deal with this problem, instead of simple phrase length restriction, we propose to apply the leaving-one-out method, which is also used for language modeling techniques (Kneser and Ney, 1995).

When using leaving-one-out, we modify the phrase translation probabilities for each sentence pair. For a training example f_n, e_n, we have to remove all phrases $C_n(\tilde{f}, \tilde{e})$ that were extracted from this sentence pair from the phrase counts that

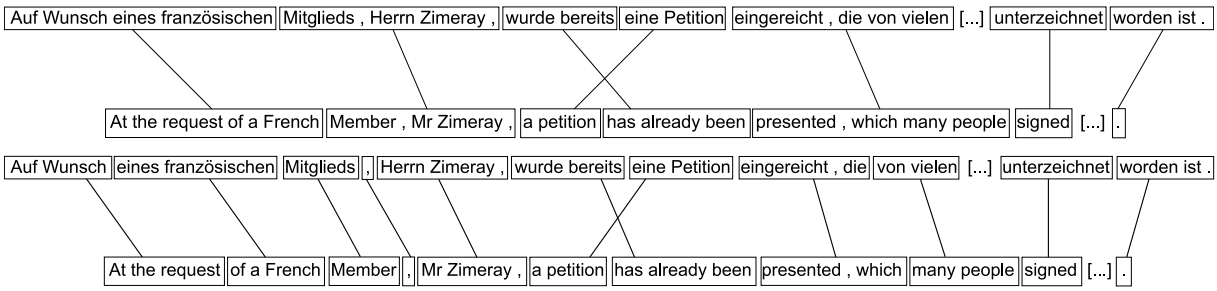

Figure 2: Segmentation example from forced alignment. Top: without leaving-one-out. Bottom: with leaving-one-out.

we used to construct our phrase translation table. The same holds for the marginal counts $C_n(\tilde{e})$ and $C_n(\tilde{f})$. Starting from Equation 1, the leaving-one-out phrase probability for training sentence pair n is

$$p_{l1o,n}(\tilde{f}|\tilde{e}) = \frac{C(\tilde{f},\tilde{e}) - C_n(\tilde{f},\tilde{e})}{C(\tilde{e}) - C_n(\tilde{e})} \quad (4)$$

To be able to perform the re-computation in an efficient way, we store the source and target phrase marginal counts for each phrase in the phrase table. A phrase extraction is performed for each training sentence pair separately using the same word alignment as for the initialization. It is then straightforward to compute the phrase counts after leaving-one-out using the phrase probabilities and marginal counts stored in the phrase table.

While this works well for more frequent observations, singleton phrases are assigned a probability of zero. We refer to singleton phrases as phrase pairs that occur only in one sentence. For these sentences, the decoder needs the singleton phrase pairs to produce an alignment. Therefore we retain those phrases by assigning them a positive probability close to zero. We evaluated with two different strategies for this, which we call standard and length-based leaving-one-out. Standard leaving-one-out assigns a fixed probability α to singleton phrase pairs. This way the decoder will prefer using more frequent phrases for the alignment, but is able to resort to singletons if necessary. However, we found that with this method longer singleton phrases are preferred over shorter ones, because fewer of them are needed to produce the target sentence. In order to better generalize to unseen data, we would like to give the preference to shorter phrases. This is done by length-based leaving-one-out, where singleton phrases are assigned the probability $\beta^{(|\tilde{f}|+|\tilde{e}|)}$ with the source and target phrase lengths $|\tilde{f}|$ and $|\tilde{e}|$ and fixed $\beta < 1$. In our experiments we set $\alpha = e^{-20}$ and $\beta = e^{-5}$. Table 1 shows the decrease in average source phrase length by application of leaving-one-out.

Table 1: Avg. source phrase lengths in forced alignment without leaving-one-out and with standard and length-based leaving-one-out.

	avg. phrase length
without l1o	2.5
standard l1o	1.9
length-based l1o	1.6

3.3 Cross-validation

For the first iteration of the phrase training, leaving-one-out can be implemented efficiently as described in Section 3.2. For higher iterations, phrase counts obtained in the previous iterations would have to be stored on disk separately for each sentence and accessed during the forced alignment process. To simplify this procedure, we propose a cross-validation strategy on larger batches of data. Instead of recomputing the phrase counts for each sentence individually, this is done for a whole batch of sentences at a time. In our experiments, we set this batch-size to 10000 sentences.

3.4 Parallelization

To cope with the runtime and memory requirements of phrase model training that was pointed out by previous work (Marcu and Wong, 2002; Birch et al., 2006), we parallelized the forced alignment by splitting the training corpus into blocks of 10k sentence pairs. From the initial phrase table, each of these blocks only loads the phrases that are required for alignment. The align-

ment and the counting of phrases are done separately for each block and then accumulated to build the updated phrase model.

4 Phrase Model Training

The produced phrase alignment can be given as a single best alignment, as the n-best alignments or as an alignment graph representing all alignments considered by the decoder. We have developed two different models for phrase translation probabilities which make use of the force-aligned training data. Additionally we consider smoothing by different kinds of interpolation of the generative model with the state-of-the-art heuristics.

4.1 Viterbi

The simplest of our generative phrase models estimates phrase translation probabilities by their relative frequencies in the Viterbi alignment of the data, similar to the heuristic model but with counts from the phrase-aligned data produced in training rather than computed on the basis of a word alignment. The translation probability of a phrase pair $(\tilde{f}, \tilde{e})$ is estimated as

$$p_{FA}(\tilde{f}|\tilde{e}) = \frac{C_{FA}(\tilde{f}, \tilde{e})}{\sum_{\tilde{f}'} C_{FA}(\tilde{f}', \tilde{e})} \quad (5)$$

where $C_{FA}(\tilde{f}, \tilde{e})$ is the count of the phrase pair $(\tilde{f}, \tilde{e})$ in the phrase-aligned training data. This can be applied to either the Viterbi phrase alignment or an n-best list. For the simplest model, each hypothesis in the n-best list is weighted equally. We will refer to this model as the *count* model as we simply count the number of occurrences of a phrase pair. We also experimented with weighting the counts with the estimated likelihood of the corresponding entry in the the n-best list. The sum of the likelihoods of all entries in an n-best list is normalized to 1. We will refer to this model as the *weighted count* model.

4.2 Forward-backward

Ideally, the training procedure would consider all possible alignment and segmentation hypotheses. When alternatives are weighted by their posterior probability. As discussed earlier, the run-time requirements for computing all possible alignments is prohibitive for large data tasks. However, we can approximate the space of all possible hypotheses by the search space that was used for the alignment. While this might not cover all phrase translation probabilities, it allows the search space and translation times to be feasible and still contains the most probable alignments. This search space can be represented as a graph of partial hypotheses (Ueffing et al., 2002) on which we can compute expectations using the Forward-Backward algorithm. We will refer to this alignment as the *full* alignment. In contrast to the method described in Section 4.1, phrases are weighted by their posterior probability in the word graph. As suggested in work on minimum Bayes-risk decoding for SMT (Tromble et al., 2008; Ehling et al., 2007), we use a global factor to scale the posterior probabilities.

4.3 Phrase Table Interpolation

As (DeNero et al., 2006) have reported improvements in translation quality by interpolation of phrase tables produced by the generative and the heuristic model, we adopt this method and also report results using log-linear interpolation of the estimated model with the original model.

The log-linear interpolations $p_{int}(\tilde{f}|\tilde{e})$ of the phrase translation probabilities are estimated as

$$p_{int}(\tilde{f}|\tilde{e}) = \Big(p_H(\tilde{f}|\tilde{e})\Big)^{1-\omega} \cdot \Big(p_{gen}(\tilde{f}|\tilde{e})\Big)^{(\omega)} \quad (6)$$

where ω is the interpolation weight, p_H the heuristically estimated phrase model and p_{gen} the count model. The interpolation weight ω is adjusted on the development corpus. When interpolating phrase tables containing different sets of phrase pairs, we retain the intersection of the two.

As a generalization of the fixed interpolation of the two phrase tables we also experimented with adding the two trained phrase probabilities as additional features to the log-linear framework. This way we allow different interpolation weights for the two translation directions and can optimize them automatically along with the other feature weights. We will refer to this method as *feature-wise combination*. Again, we retain the intersection of the two phrase tables. With good log-linear feature weights, feature-wise combination should perform at least as well as fixed interpolation. However, the results presented in Table 5

Table 2: Statistics for the Europarl German-English data

		German	English
TRAIN	Sentences	1 311 815	
	Run. Words	34 398 651	36 090 085
	Vocabulary	336 347	118 112
	Singletons	168 686	47 507
DEV	Sentences	2 000	
	Run. Words	55 118	58 761
	Vocabulary	9 211	6 549
	OOVs	284	77
TEST	Sentences	2 000	
	Run. Words	56 635	60 188
	Vocabulary	9 254	6 497
	OOVs	266	89

show a slightly lower performance. This illustrates that a higher number of features results in a less reliable optimization of the log-linear parameters.

5 Experimental Evaluation

5.1 Experimental Setup

We conducted our experiments on the German-English data published for the *ACL 2008 Workshop on Statistical Machine Translation* (WMT08). Statistics for the Europarl data are given in Table 2.

We are given the three data sets $TRAIN$, DEV and $TEST$. For the heuristic phrase model, we first use GIZA++ (Och and Ney, 2003) to compute the word alignment on $TRAIN$. Next we obtain a phrase table by extraction of phrases from the word alignment. The scaling factors of the translation models have been optimized for BLEU on the DEV data.

The phrase table obtained by heuristic extraction is also used to initialize the training. The forced alignment is run on the training data $TRAIN$ from which we obtain the phrase alignments. Those are used to build a phrase table according to the proposed generative phrase models. Afterward, the scaling factors are trained on DEV for the new phrase table. By feeding back the new phrase table into forced alignment we can reiterate the training procedure. When training is finished the resulting phrase model is evaluated on DEV and $TEST$. Additionally, we can apply smoothing by interpolation of the new phrase table with the original one estimated heuristically, retrain the scaling factors and evaluate afterwards.

Table 3: Comparison of different training setups for the count model on DEV.

leaving-one-out	max phr.len.	BLEU	TER
baseline	6	25.7	61.1
none	2	25.2	61.3
	3	25.7	61.3
	4	25.5	61.4
	5	25.5	61.4
	6	25.4	61.7
standard	6	26.4	60.9
length-based	6	26.5	60.6

The baseline system is a standard phrase-based SMT system with eight features: phrase translation and word lexicon probabilities in both translation directions, phrase penalty, word penalty, language model score and a simple distance-based reordering model. The features are combined in a log-linear way. To investigate the generative models, we replace the two phrase translation probabilities and keep the other features identical to the baseline. For the feature-wise combination the two generative phrase probabilities are added to the features, resulting in a total of 10 features. We used a 4-gram language model with modified Kneser-Ney discounting for all experiments. The metrics used for evaluation are the case-sensitive BLEU (Papineni et al., 2002) score and the translation edit rate (TER) (Snover et al., 2006) with one reference translation.

5.2 Results

In this section, we investigate the different aspects of the models and methods presented before. We will focus on the proposed leaving-one-out technique and show that it helps in finding good phrasal alignments on the training data that lead to improved translation models. Our final results show an improvement of 1.4 BLEU over the heuristically extracted phrase model on the test data set.

In Section 3.2 we have discussed several methods which aim to overcome the over-fitting prob-

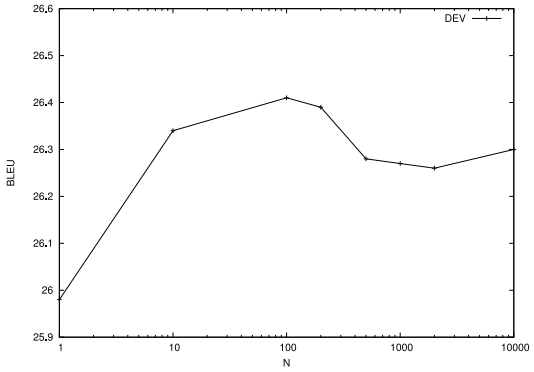

Figure 3: Performance on DEV in BLEU of the count model plotted against size n of n-best list on a logarithmic scale.

lems described in (DeNero et al., 2006). Table 3 shows translation scores of the count model on the development data after the first training iteration for both leaving-one-out strategies we have introduced and for training without leaving-one-out with different restrictions on phrase length. We can see that by restricting the source phrase length to a maximum of 3 words, the trained model is close to the performance of the heuristic phrase model. With the application of leaving-one-out, the trained model is superior to the baseline, the length-based strategy performing slightly better than standard leaving-one-out. For these experiments the count model was estimated with a 100-best list.

The count model we describe in Section 4.1 estimates phrase translation probabilities using counts from the n-best phrase alignments. For smaller n the resulting phrase table contains fewer phrases and is more deterministic. For higher values of n more competing alignments are taken into account, resulting in a bigger phrase table and a smoother distribution. We can see in Figure 3 that translation performance improves by moving from the Viterbi alignment to n-best alignments. The variations in performance with sizes between $n = 10$ and $n = 10000$ are less than 0.2 BLEU. The maximum is reached for $n = 100$, which we used in all subsequent experiments. An additional benefit of the count model is the smaller phrase table size compared to the heuristic phrase extraction. This is consistent with the findings of (Birch et al., 2006). Table 4 shows the phrase table sizes for different n. With $n = 100$ we retain only 17% of the original phrases. Even for the full model, we do not retain all phrase table entries. Due to pruning in the forced alignment step, not all translation options are considered. As a result experiments can be done more rapidly and with less resources than with the heuristically extracted phrase table. Also, our experiments show that the increased performance of the count model is partly derived from the smaller phrase table size. In Table 5 we can see that the performance of the heuristic phrase model can be increased by 0.6 BLEU on $TEST$ by filtering the phrase table to contain the same phrases as the count model and reoptimizing the log-linear model weights. The experiments on the number of different alignments taken into account were done with standard leaving-one-out.

Table 4: Phrase table size of the count model for different n-best list sizes, the full model and for heuristic phrase extraction.

N	# phrases	% of full table
1	4.9M	5.3
10	8.4M	9.1
100	15.9M	17.2
1000	27.1M	29.2
10000	40.1M	43.2
full	59.6M	64.2
heuristic	92.7M	100.0

The final results are given in Table 5. We can see that the count model outperforms the baseline by 0.8 BLEU on DEV and 0.9 BLEU on $TEST$ after the first training iteration. The performance of the filtered baseline phrase table shows that part of that improvement derives from the smaller phrase table size. Application of cross-validation (cv) in the first iteration yields a performance close to training with leaving-one-out (l1o), which indicates that cross-validation can be safely applied to higher training iterations as an alternative to leaving-one-out. The weighted count model clearly under-performs the simpler count model. A second iteration of the training algorithm shows nearly no changes in BLEU score, but a small improvement in TER. Here, we used the phrase table trained with leaving-one-out in the first iteration and applied cross-validation in the second iteration. Log-linear interpolation of the count model with the heuristic yields a further increase, showing an improvement of 1.3 BLEU on DEV and 1.4 BLEU on $TEST$ over the baseline. The interpo-

Table 5: Final results for the heuristic phrase table filtered to contain the same phrases as the count model (baseline filt.), the count model trained with leaving-one-out (l1o) and cross-validation (cv), the weighted count model and the full model. Further, scores for fixed log-linear interpolation of the count model trained with leaving-one-out with the heuristic as well as a feature-wise combination are shown. The results of the second training iteration are given in the bottom row.

	DEV		TEST	
	BLEU	TER	BLEU	TER
baseline	25.7	61.1	26.3	60.9
baseline filt.	26.0	61.6	26.9	61.2
count (l1o)	26.5	60.6	27.2	60.5
count (cv)	26.4	60.7	27.0	60.7
weight. count	25.9	61.4	26.4	61.3
full	26.3	60.0	27.0	60.2
fixed interpol.	**27.0**	**59.4**	**27.7**	**59.2**
feat. comb.	26.8	60.1	27.6	59.9
count, iter. 2	26.4	60.3	27.2	60.0

lation weight is adjusted on the development set and was set to $\omega = 0.6$. Integrating both models into the log-linear framework (feat. comb.) yields a BLEU score slightly lower than with fixed interpolation on both DEV and $TEST$. This might be attributed to deficiencies in the tuning procedure. The full model, where we extract all phrases from the search graph, weighted with their posterior probability, performs comparable to the count model with a slightly worse BLEU and a slightly better TER.

6 Conclusion

We have shown that training phrase models can improve translation performance on a state-of-the-art phrase-based translation model. This is achieved by training phrase translation probabilities in a way that they are consistent with their use in translation. A crucial aspect here is the use of leaving-one-out to avoid over-fitting. We have shown that the technique is superior to limiting phrase lengths and smoothing with lexical probabilities alone.

While models trained from Viterbi alignments already lead to good results, we have demonstrated that considering the 100-best alignments allows to better model the ambiguities in phrase segmentation.

The proposed techniques are shown to be superior to previous approaches that only used lexical probabilities to smooth phrase tables or imposed limits on the phrase lengths. On the WMT08 Europarl task we show improvements of 0.9 BLEU points with the trained phrase table and 1.4 BLEU points when interpolating the newly trained model with the original, heuristically extracted phrase table. In TER, improvements are 0.4 and 1.7 points.

In addition to the improved performance, the trained models are smaller leading to faster and smaller translation systems.

Acknowledgments

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001-06-C-0023. Any opinions, ndings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reect the views of the DARPA.

References

Alexandra Birch, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *smt2006*, pages 154–157, Jun.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 200–208, Columbus, Ohio, June. Association for Computational Linguistics.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–312, June.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 25–28, Morristown, NJ, USA. Association for Computational Linguistics.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why Generative Phrase Models Underperform Surface Heuristics. In *Proceedings of the*

Workshop on Statistical Machine Translation, pages 31–38, New York City, June.

John DeNero, Alexandre Buchard-Côté, and Dan Klein. 2008. Sampling Alignment Structure under a Bayesian Translation Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, October.

Nicola Ehling, Richard Zens, and Hermann Ney. 2007. Minimum bayes risk decoding for bleu. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 101–104, Morristown, NJ, USA. Association for Computational Linguistics.

Jesús-Andrés Ferrer and Alfons Juan. 2009. A phrase-based hidden semi-markov approach to machine translation. In *Procedings of European Association for Machine Translation (EAMT)*, Barcelona, Spain, May. European Association for Machine Translation.

Reinhard Kneser and Hermann Ney. 1995. Improved Backing-Off for M-gram Language Modelling. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 181–184, Detroit, MI, May.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Percy Liang, Alexandre Buchard-Côté, Dan Klein, and Ben Taskar. 2006. An End-to-End Discriminative Approach to Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia.

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

J.A. Nelder and R. Mead. 1965. A Simplex Method for Function Minimization. *The Computer Journal)*, 7:308–313.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, December.

F.J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99)*, pages 20–28, University of Maryland, College Park, MD, USA, June.

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 on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Wade Shen, Brian Delaney, Tim Anderson, and Ray Slyh. 2008. The MIT-LL/AFRL IWSLT-2008 MT System. In *Proceedings of IWSLT 2008*, pages 69–76, Hawaii, U.S.A., October.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*, pages 223–231, Aug.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629, Honolulu, Hawaii, October. Association for Computational Linguistics.

N. Ueffing, F.J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing*, pages 156–163, Philadelphia, PA, USA, July.