Length-incremental Phrase Training for SMT

Joern Wuebker and Hermann Ney
Human Language Technology and Pattern Recognition Group
RWTH Aachen University
Aachen, Germany
{wuebker,ney}@cs.rwth-aachen.de

Abstract

We present an iterative technique to generate phrase tables for SMT, which is based on force-aligning the training data with a modified translation decoder. Different from previous work, we completely avoid the use of a word alignment or phrase extraction heuristics, moving towards a more principled phrase generation and probability estimation. During training, we allow the decoder to generate new phrases on-the-fly and increment the maximum phrase length in each iteration. Experiments are carried out on the IWSLT 2011 Arabic-English task, where we are able to reach moderate improvements on a state-of-the-art baseline with our training method. The resulting phrase table shows only a small overlap with the heuristically extracted one, which demonstrates the restrictiveness of limiting phrase selection by a word alignment or heuristics. By interpolating the heuristic and the trained phrase table, we can improve over the baseline by 0.5% BLEU and 0.5% TER.

1 Introduction

Most state-of-the-art SMT systems get the statistics from which the different component models are estimated via heuristics using a Viterbi word alignment. The word alignment is usually generated with tools like GIZA++ (Och and Ney, 2003), that apply the EM algorithm to estimate the alignment with the HMM or IBM-4 translation models. This is also the case for the phrases or rules which serve as translation units for the decoder. All phrases that do not violate the word alignment are extracted and their probabilities are estimated as relative frequencies (Koehn et al., 2003).

A number of different approaches have tried to do away with the heuristics and close this gap between the phrase table generation and translation decoding. However, most of these approaches either fail to achieve state-of-the-art performance or still make use of the word alignment or the extraction heuristics, e.g. as a prior in discriminative training or to initialize a generative or generatively inspired training procedure and are thus biased by their weaknesses. Here, we aim at moving towards the ideal situation, where a unified framework induces the phrases based on the same models as in decoding.

We train the phrase table without using a word alignment or the extraction heuristics. Different from previous work, we are able to generate all possible phrase pairs on-the-fly during the training procedure. A further advantage of our proposed algorithm is that we use basically the same beam search as in translation. This makes it easy to re-implement by modifying any translation decoder, and makes sure that training and translation are consistent. In principle, we apply the forced decoding approach described in (Wuebker et al., 2010) with cross-validation to prevent over-fitting, but we initialize the phrase table with IBM-1 lexical probabilities (Brown et al., 1993) instead of heuristically extracted relative frequencies. The algorithm is extended with the concept of *back-off phrases*, so that new phrase pairs can be generated at training time. The size of the newly generated phrases is incremented over the training iterations. By introducing *fallback decoding runs*, we are able to successfully align the complete training data. *Local language models* are used for better phrase pair pre-selection.

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 309–319,
Sofia, Bulgaria, August 8-9, 2013

The experiments are carried out on the IWSLT 2011 Arabic-English shared task. We are able to show that it is possible and feasible to reach state-of-the-art performance without the need to word-align the bilingual training data. The small overlap of 18.5% between the trained and the heuristically extracted phrase table demonstrates the limitations of previous work, where training is initialized by the baseline phrase table or phrase selection is restricted by a word alignment. With a linear interpolation of phrase tables an improvement of 0.5% BLEU and 0.5% TER over the baseline can be achieved. The result in BLEU is statistically significant on the test set with 90% confidence. Further, we can confirm the observation of previous work, that phrases with near-zero entropies seem to be a disadvantage for translation quality. Although we use a phrase-based decoder here, the principles of our work can be applied to any statistical machine translation paradigm. The software used for our experiments is available under a non-commercial open source licence.

The paper is organized as follows. We review related work in Section 2. The decoder and its features are described in Section 3 and we give an overview of the training procedure in Section 4. The complete algorithm is described in Section 5 and experiments are presented in Section 6. We conclude with Section 7.

2 Related Work

Marcu and Wong (2002) present a joint probability model, which is trained with a hill-climbing technique based on break, merge, swap and move operations. Due to the computational complexity they are only able to consider phrases, which appear at least five times in the data. The model is shown to slightly underperform heuristic extraction in (Koehn et al., 2003). For higher efficiency, it is constrained by a word alignment in (Birch et al., 2006). DeNero et al. (2008) introduce a different training procedure for this model based on a Gibbs sampler. They make use of the word alignment for initialization.

A generative phrase model trained with the Expectation-Maximization (EM) algorithm is shown in (DeNero et al., 2006). It also does not reach the same top performance as heuristic extraction. The authors identify the hidden segmentation variable, which results in over-fitting, as the main problem.

Liang et al. (2006) present a discriminative translation system. One of the proposed strategies for training, which the authors call bold updating, is similar to our training scheme. They use heuristically extracted phrase translation probabilities as blanket features in all setups.

Another iteratively-trained phrase model is described by Moore and Quirk (2007). Their model is segmentation-free and, confirming the findings in (DeNero et al., 2006), can close the gap to phrase tables induced from surface heuristics. It relies on word alignment for phrase selection.

Mylonakis and Sima'an (2008) present a phrase model, whose training procedure uses prior probabilities based on Inversion Transduction Grammar and smoothing as learning objective to prevent over-fitting. They also rely on the word alignment to select phrase pairs.

Blunsom et al. (2009) perform inference over latent synchronous derivation trees under a non-parametric Bayesian model with a Gibbs sampler. Training is also initialized by extracting rules from a word alignment, but the authors let the sampler diverge from the initial value for 1000 passes over the data, before the samples are used. However, as the model is to weak for actual translation, the usual extraction heuristics are applied on the hierarchical alignments to infer a distribution over rule tables.

Wuebker et al. (2010) use a forced decoding training procedure, which applies a leave-one-out technique to prevent over-fitting. They are able to show improvements over a heuristically extracted phrase table, which is used for initialization of the training.

In (Saers and Wu, 2011), the EM algorithm is applied for principled induction of bilexica based on linear inversion transduction grammar. The model itself underperforms the baseline, but the authors show moderate improvements by combining it with the baseline phrase table, which is similar to our results.

(Neubig et al., 2011) also propose a probabilistic model based on inversion transduction grammar, which allows for direct phrase table extraction from unaligned data. They show results similar to the heuristic baseline on several tasks.

A number of different models that can be trained from forced derivation trees are shown in (Duan et al., 2012), including a re-estimated translation model, two reordering models and a rule se-

quence model. For inference, they optimize their parameters towards alignment F-score. The forced derivations are initialized with the standard heuristic extraction scheme.

He and Deng (2012) describe a discriminative phrase training procedure, where n-best translations are produced by the decoder on the whole training data. The heuristically extracted relative frequencies serve as a prior, and the probabilities are updated with a maximum BLEU criterion based on the n-best lists.

3 Translation Model

We use the standard phrase-based translation decoder from the open source toolkit *Jane 2* (Wuebker et al., 2012a) for both the training procedure and the translation experiments. It makes use of the usual features: Translation channel models in both directions, lexical smoothing models in both directions, an n-gram language model (LM), phrase and word penalty and a jump-distance-based distortion model. Formally, the best translation $\hat{e}_1^{\hat{I}}$ as defined by the models $h_m(e_1^I, s_1^K, f_1^J)$ can be written as (Och and Ney, 2004)

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

where $f_1^J = f_1 \dots f_J$ is the source sentence, $e_1^I = e_1 \dots e_I$ the target sentence and $s_1^K = s_1 \dots s_K$ their phrase segmentation and alignment. We define $s_k := (i_k, b_k, j_k)$, where i_k is the 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. Different from many standard systems, the lexical smoothing scores are not estimated by extracting counts from a word alignment, but with IBM-1 model scores trained on the bilingual data with GIZA++. They are computed as (Zens, 2008)

$$h_{lex}(e_1^I, s_1^K, f_1^J) = \sum_{k=1}^{K} \sum_{j=b_k}^{j_k} log \left(p(f_j|e_0) + \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \right) \quad (2)$$

Here, e_0 denotes the empty target word. The lexical smoothing model for the inverse direction is computed analogously. The log-linear feature weights λ_m are optimized on a development data set with minimum error rate training (MERT) (Och, 2003). As optimization criterion we use BLEU (Papineni et al., 2001).

4 Training

4.1 Overview

In this work we employ a training procedure inspired by the Expectation-Maximization (EM) algorithm.

The **E-step** corresponds to force-aligning the training data with a modified translation decoder, which yields a distribution over possible phrasal segmentations and their alignment. Different from original EM, we make use of not only the two translation channel models that are being learned, but the full log-linear combination of models as in translation decoding. Formally, we are searching for the best phrase segmentation and alignment for the given sentence pair, which is defined by

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

To force-align the training data, the translation decoder is constrained to the given target sentence. The translation candidates applicable for each sentence pair are selected through a bilingual phrase matching before the actual search.

In the **M-step**, we re-estimate the phrase table from the phrase alignments. The translation probability of a phrase pair $(\tilde{f}, \tilde{e})$ is estimated as

$$p_{FA}(\tilde{f}|\tilde{e}) = \frac{C_{FA}(\tilde{f}, \tilde{e})}{\sum_{\tilde{f}'} C_{FA}(\tilde{f}', \tilde{e})} \quad (4)$$

where $C_{FA}(\tilde{f}, \tilde{e})$ is the count of the phrase pair $(\tilde{f}, \tilde{e})$ in the phrase-aligned training data.

In contrast to original EM, this is done by taking the phrase counts from a uniformly weighted n-best list. The limitation to n phrase alignments helps keeping the number of considered phrases reasonably small. Because the log-linear feature weights have been tuned in a discriminative fashion to optimize the ranking of translation hypotheses, rather than their probability distribution, posterior probabilities received by exponentiation and renormalization need to be scaled similar to (Wuebker et al., 2012b). Uniform weights can alleviate this mismatch between the discriminatively

trained log-linear feature weights and the actual probability distribution, without having to resort to an arbitrarily chosen global scaling factor. This corresponds to the *count model* in (Wuebker et al., 2010) and was shown by the authors to perform similar or better than using actual posterior probabilities. In our experiments, we set the size of the n-best list to $n = 1000$.

The first iteration of phrase training is initialized with an empty phrase table. We use the notion of *backoff phrases* to generate new phrase pairs on-the-fly. To avoid over-fitting, we apply the cross-validation technique presented in (Wuebker et al., 2010) with a batch-size of 2000 sentences. This means that for each batch the phrase and marginal counts from the full phrase table are reduced by the statistics taken from the same batch in the previous iteration. The phrase translation probabilities are then estimated from these updated counts. Phrase pairs only appearing in a single batch are assigned a fixed penalty.

4.2 Backoff Phrases

Backoff phrases are phrase pairs that are generated on-the-fly by the decoder at training time. When aligning a sentence pair, for a given maximum phrase length m, the decoder inserts all combinations of source m_s-grams and target m_t-grams into the translation options, that are present in the sentence pair and with $m_s, m_t \leq m$. Formally, for the sentence pair (f_1^J, e_1^I), $f_1^J = f_1 \dots f_J$, $e_1^I = e_1 \dots e_I$, and maximum length m, we generate all phrase pairs $(\tilde{f}, \tilde{e})$ where

$$\begin{aligned} &\exists m_s, m_t, j, i : \\ &1 \leq m_s, m_t \leq m \wedge 1 \leq j \leq J - m_s + 1 \\ &\wedge 1 \leq i \leq I - m_t + 1 \\ &\wedge \tilde{f} = f_j^{(j+m_s-1)} \wedge \tilde{e} = e_i^{(i+m_t-1)}. \end{aligned} \quad (5)$$

These generated phrase pairs are given a fixed penalty pen_p per phrase, pen_s per source word and pen_t per target word, which are summed up and substituted for the two channel models. The lexical smoothing scores are computed in the usual way based on an IBM-1 table. Note that this table is not extracted from a word alignment, but contains the real probabilities trained with the IBM-1 model by GIZA++.

We use backoff phrases in two different contexts. In the first $m_{max} = 6$ iterations, they are applied as a means to generate new phrase pairs on the fly. We increase the maximum phrase length m in each iteration and always generate all possible backoff phrases before aligning each sentence. Later, when a sufficient number of phrases have been generated in the previous iterations, they are used as a last resort in order to avoid alignment failures.

At the later stage of the length-incremental training, we also make use of a modified version, where we only allow new phrase pairs $(\tilde{f}, \tilde{e})$ to be generated, if no translation candidates exist for $\tilde{f}$ after the bilingual phrase matching. However, in this case, backoff phrases are only used if a first decoding run fails and we have to resort to *fallback runs*, which are described in the next Section.

4.3 Fallback Decoding Runs

To maximize the number of successfully aligned sentences, we allow for *fallback decoding runs* with slightly altered parameterization, whenever constrained decoding fails. In this work, we only change the parameterization of the backoff phrases. After $m_{max} = 6$ iterations, we no longer generate any backoff phrases in the first decoding run. If it fails, a second run is performed, where we allow to generate backoff phrases for all source phrases, which have no target candidates after the bilingual phrase matching. Finally, if this one also fails, all possible phrases are generated in the third run. Here, the maximum backoff phrase length is fixed to $m = 1$. We denote the number of fallback runs with $n_{fb} = 2$. In our experiments, the two fallback runs enable us to align every sentence pair of the training data after the sixth iteration.

4.4 Local Language Models

To make the training procedure feasible, it is parallelized by splitting the training data into batches of 2000 sentences. The batches are aligned independently. For each batch, we produce a *local language model*, which is a unigram LM trained on the target side of the current batch. We presort the phrases before search by their log-linear model score, which uses the phrase-internal unigram LM costs as one feature function. One effect of this is that the order in which phrase candidates are considered is adjusted to the local part of the data, which has a positive effect on decoding speed. Secondly, we limit the number of translation candidates for each source phrase to the best scoring 500 before the bilingual phrase matching.

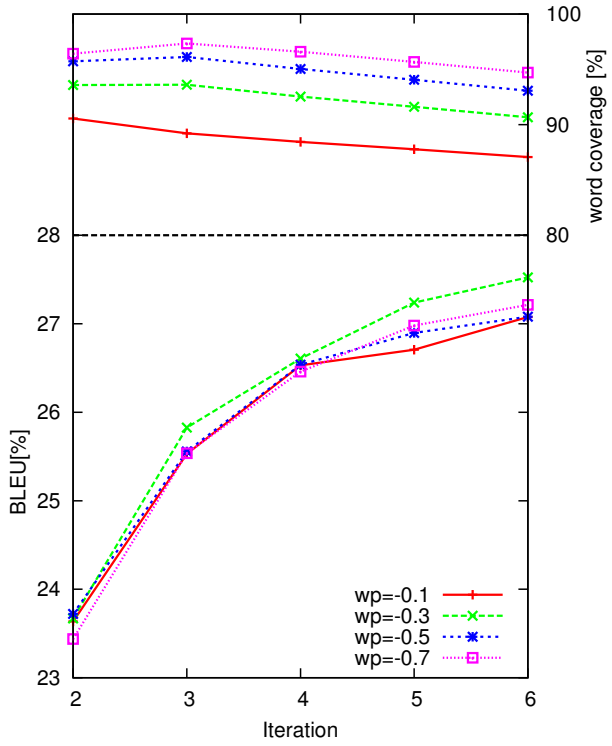

Figure 1: BLEU scores and word coverages on dev over the first 6 training iterations with different word penalties (wp).

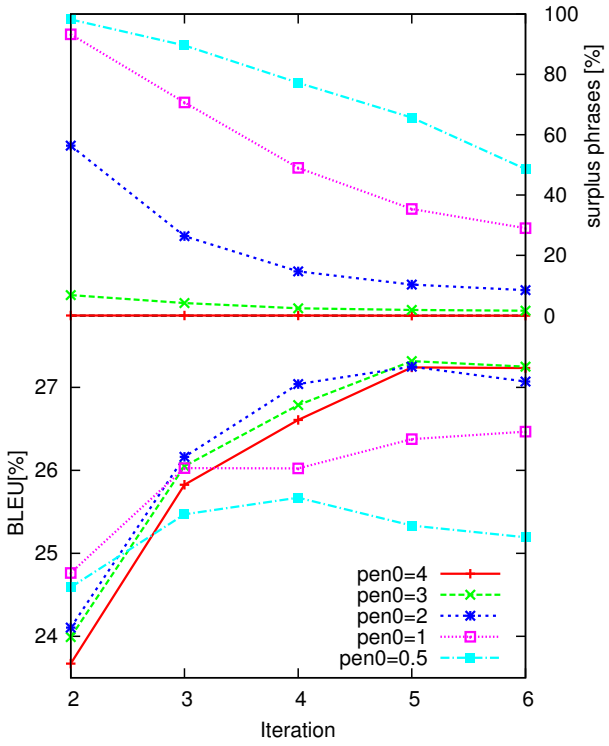

Figure 2: BLEU scores and percentage of surplus phrases on dev over the first 6 training iterations with different backoff phrase penalties pen_0.

Using the local LM for this means that the preselection better suits the current data batch. As a result, the number of phrases remaining after the phrase matching is increased as compared to the same setup without a local language model.

4.5 Parameterization

The training procedure has a number of hyper parameters, most of which do not seem to have a strong impact on the results. This section describes the parameters that have to be chosen carefully. To successfully align a sentence pair, our decoder is required to fully cover the source sentence. However, in order to achieve a good success rate in terms of number of aligned sentence pairs, we allow for incompletely aligned target sentences. We denote the percentage of successfully aligned sentence pairs as *sentence coverage*. Note that we count a sentence pair as successfully aligned, even if the target sentence is not fully covered. the **word penalty** (wp) feature weight λ_{wp} needs to be adjusted carefully. A high value leads to a high sentence coverage, but many of their target sides may be incompletely aligned. A low word penalty can decrease the sentence coverage, while aligning larger parts of the target sentences. We denote the total percentage of successfully aligned target words as *word coverage*. Please note the distincton to the sentence coverage, which is defined above. Figure 1 shows the word coverages and BLEU scores for training iterations 2 through 6 with different word penalties. In the first iteration, the results are identical, as only one-to-one phrases are allowed and the number of aligned target words is therefore predetermined. For $\lambda_{wp} = -0.1$, the word coverages are continuously decreasing with each iteration, although not by much. For $\lambda_{wp} = -0.3$ to $\lambda_{wp} = -0.7$ the word coverage slightly increases from iteration 2 to 3 and then decreases again. In terms of BLEU score, $\lambda_{wp} = -0.3$ has a slight advantage over the other values and we decided to continue using this value in all subsequent experiments.

The **backoff phrase penalties** directly affect the learning rate of the training procedure. With low penalties, only few, very good phrases get an advantage over the ones generated on-the-fly, which corresponds to a slow learning rate. In-

1. Initialize with empty phrase table

2. Set backoff phrase penalties to $pen_0 = 3$ and $m = 1$

3. Until $m = m_{max}$, iterate:

   - If iteration > 1: set $m = m + 1$, $\lambda_{s2t} = \lambda_{s2t} + \delta$, $\lambda_{t2s} = \lambda_{t2s} + \delta$
   - Force-align training data and re-estimate phrase table

4. Set $m = 1$ and $n_{fb} = 2$

5. Iterate:

   - Force-align training data and re-estimate phrase table

Figure 3: The complete training algorithm.

creasing the penalties means that a larger percentage of the phrase pairs generated in the previous iterations will be favored over new backoff phrases, which corresponds to a faster learning rate. We denote phrase pairs that are more expensive than their backoff phrase counterparts as *surplus phrases*. Figure 2 shows the behavior over the training iterations 2 through 6 with different penalties pen_0 in terms of percentage of surplus phrase pairs and BLEU score. Here we set $pen_s = pen_t = pen_0$ and $pen_p = 5pen_0$. We can see that $pen_0 = 4$ yields less than 0.1% surplus phrases through all iterations, whereas $pen_0 = 0.5$ starts off with 98.2% surplus phrases and goes down to 55.9% in iteration 6. In terms of BLEU, a fast learning rate seems to be preferable. The best results are achieved with $pen_0 = 3$, where the rate of surplus phrases starts at 6.8% and decreases to 1.7% until iteration 6. In all subsequent experiments, we set $pen_0 = 3$.

5 Length-incremental Training

In this section we describe the complete training algorithm. The first training iteration is initialized with an empty phrase table. The phrases used in alignment are backoff phrases, which are generated on-the-fly. The maximum backoff phrase length is set to $m = 1$. Then the forced alignment is iterated, increasing m by 1 in each iteration, up to a maximum of $m_{max} = 6$.

After $m_{max} = 6$ iterations, we have created a sufficient number of phrase pairs and continue iterating the training procedure with new parameters. Now, we do not allow usage of any backoff phrases in the first decoding run. If the first run fails, we allow a fallback decoding run, where backoff phrases are generated only for source phrases without translation candidates. If this one also fails, in a final fallback run all possible phrases are generated. Here we allow a maximum backoff phrase length of $m = 1$.

The log-linear feature weights λ_i used for training are mostly standard values. Only λ_{wp} for the word penalty is adjusted as described in Section 4.5, and λ_{s2t},λ_{t2s} for the two phrasal channel models are incremented with each iteration. We start off with $\lambda_{s2t} = \lambda_{t2s} = 0$ and increment the weights by $\delta = 0.02$ in each iteration, until the standard value $\lambda_{s2t} = \lambda_{t2s} = 0.1$ is reached in iteration 6, after which the values are kept fixed. MERT is not part of the training procedure, but only used afterwards for evaluation. The full algorithm is illustrated in Figure 3.

		Arabic	English
train	Sentences	305K	
	Running Words	6.5M	6.5M
	Vocabulary	104K	74K
dev	Sentences	934	
	Run. Words	19K	20K
	Vocabulary	4293	3182
	OOVs (run. words)	445	182
test	Sentences	1664	
	Run. Words	31K	32K
	Vocabulary	5415	3650
	OOVs (run. words)	658	159

Table 1: Statistics for the IWSLT 2011 Arabic-English data. The out-of-vocabulary words are denoted as OOVs.

6 Experiments

6.1 Data

We carry out our experiments on the IWSLT 2011 Arabic-English shared task[1]. It focuses on the translation of TED talks, a collection of lectures on a variety of topics ranging from science to culture. Our bilingual training data is composed of all available in-domain (TED) data and a selection of the out-of-domain MultiUN data provided for the evaluation campaign. The bilingual data selection

[1] www.iwslt2011.org

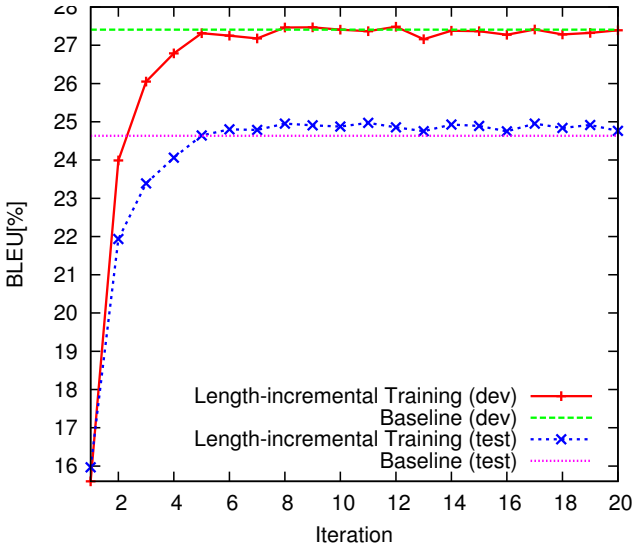

Figure 4: BLEU scores on dev and test over 20 training iterations.

is based on (Axelrod et al., 2011). Data statistics are given in Table 1. The language model is a 4-gram LM trained on all provided in-domain monolingual data and a selection based on (Moore and Lewis, 2010) of the out-of-domain corpora. To account for statistical variation, all reported results are average scores over three independent MERT runs.

6.2 Results

To build the baseline phrase table, we perform the standard phrase extraction from a symmetrized word alignment created with the IBM-4 model by GIZA++. The length of the extracted phrases is limited to a maximum of six words. The lexical smoothing scores are computed from IBM-1 probabilities. We run MERT on the development set (dev) and evaluate on the test set (test). A second baseline is the technique described in (Wuebker et al., 2010), which we denote as *leave-one-out*. It is initialized with the heuristically extracted table and run for one iteration, which the authors have shown to be sufficient.

Length-incremental training is performed as described in Section 5. After each iteration, we run MERT on dev using the resulting phrase table and evaluate. The set of models used here is identical to the baseline.

The results in BLEU are plotted in Figure 4. We can see that the performance increases up to iteration 5, after which only small changes can be observed. The performance on dev is similar to the baseline, on test the trained phrase tables are consistently slightly above the baseline. The optimum on dev is reached in iteration 12. Exact BLEU and TER (Snover et al., 2006) scores of the optimum on dev and the baseline are given in Table 2. The phrase table trained with leave-one-out (Wuebker et al., 2010) performs simlar to the heuristic baseline. Length-incremental training is slightly superior to the baseline, yielding an improvement of 0.3% BLEU and 0.4% TER on test. Similar to results observed in (DeNero et al., 2006) and (Wuebker et al., 2010), a linear interpolation with the baseline containing all phrase pairs from either of the two tables yields a moderate improvement of 0.5% BLEU and 0.5% TER both data sets. The BLEU improvement on test is statistically significant with 90% confidence based on bootstrap resampling as described by Koehn (2004).

	dev		test	
	BLEU [%]	TER [%]	BLEU [%]	TER [%]
baseline	27.4	54.0	24.6	57.8
leave-one-out	27.3	54.2	24.6	57.7
length-increm.	27.5	53.8	24.9	57.4
lin. interp.	**27.9**	**53.5**	**25.1**†	**57.3**

Table 2: BLEU and TER scores of the baseline, phrase training with leave-one-out and length-incremental training after 12 iterations, as well as a linear interpolation of the baseline with length-incremental phrase table. Results marked with † are statistically significant with 90% confidence.

6.3 Analysis

In Figure 5 we plot the number of phrase pairs present in the phrase tables after each iteration. In the first 6 iterations, we keep generating new phrase pairs via backoff phrases. The maximum of 14.4M phrase pairs is reached after three iterations. For comparison, the size of the heuristically extracted table is 19M phrase pairs. Afterwards, backoff phrases are only used in fallback decoding runs, which leads to drop in the number of phrase pairs that are being used. It levels out at 10.4M phrases.

When we take a look at the phrase length distributions in both the baseline and the trained phrase table shown in Figure 6, we can see that in the latter the phrases are generally shorter, which con-

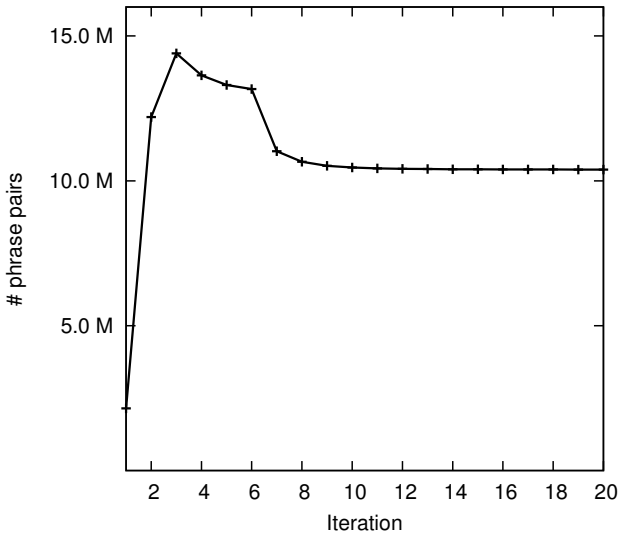

Figure 5: Number of generated phrase pairs over 20 training iterations.

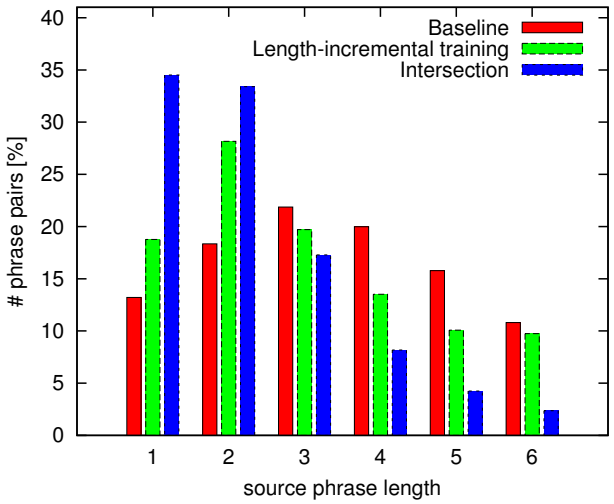

Figure 6: Histogram of the phrase lengths present in the phrase tables.

firms observations from previous work. In the trained phrase table, phrases of length one and two make up 47% of all phrases. In the heuristically extracted table it is only 32%. This is even more pronounced in the intersection of the two tables, where 68% of the phrases are of length one and two.

Interestingly, the total overlap between the two phrase tables is rather small. Only about 18.5% of the phrases from the trained table also appear in the heuristically extracted one. This shows that, by generating phrases on-the-fly without restrictions based on a word alignment or a bias from intialization, our training procedure strongly diverges from the baseline phrase table. We conclude that most previous work in this area, which adhered to the above mentioned restrictions, was only able to explore a fraction of the full potential of real phrase training.

Following (DeNero et al., 2006), we compute the entropy of the distributions within the phrase tables to quantify the 'smoothness' of the distribution. For a given source phrase $\tilde{f}$, it is defined as

$$H(\tilde{f}) = \sum_{\tilde{e}} p(\tilde{e}|\tilde{f}) log(p(\tilde{e}|\tilde{f})). \quad (6)$$

A flat distribution with a high level of uncertainty yields a high entropy, whereas a peaked distribution with little uncertainty produces a low entropy. We analyze the phrase tables filtered towards the `dev` and `test` sets. The average entropy, weighted by frequency, is 3.1 for the table learned with length-incremental training, compared to 2.7 for the heuristically extracted one. However, the interpolated table, which has the best performance, lies in between with an average entropy of 2.9. When we consider the histogram of entropies for the phrase tables in Figure 7, we can see that in the baseline phrase table 3.8% of the phrases haven an entropy below 0.5, compared to 0.90% for length-incremental training and 0.16% for the linear interpolation. Therefore, we can confirm the observation in (DeNero et al., 2006), that phrases with a near-zero entropy are undesirable for decoding. The distribution of the higher entropies, however, does not seem to matter for translation quality. This also gives us a handle for understanding, why phrase table interpolation often improves results: It largely seems to eliminate near-zero entropies from either table.

6.4 Training time

The training was not run under controlled conditions, so we can only give a rough estimate of how the training times between the different methods compare. Also, some of the steps were parallelized while others are not. To account for the computational resources needed, we report the trainig times on a single machine by summing the times for all parallel and sequential processes.

Heuristc phrase extraction from the word alignment took us about 1.7 hours. A single iteration of standard phrase training (leave-one-out) needs about 24 hours. The first iteration of length-

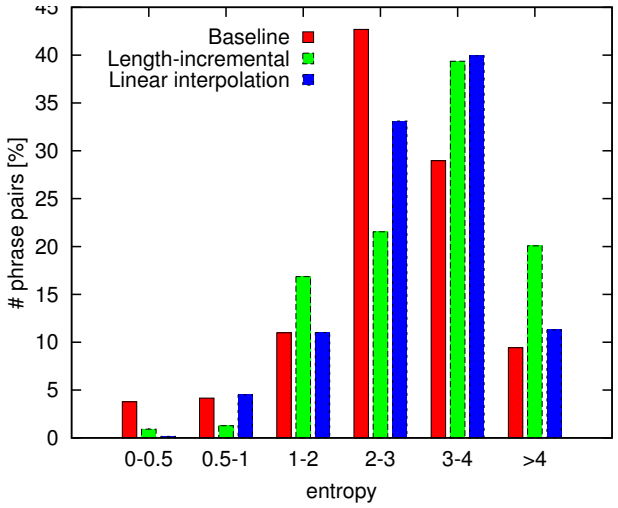

Figure 7: Histogram of entropies present in the phrase tables.

incremental training as well as all iterations after the sixth also took roughly 24 hours. The iterations two through six of length-incremental training are considerably more expensive due to the larger size of backoff phrases. Iteration six, with a maximum backoff phrase size of six words on source and target side, was the slowest with around 740 hours.

7 Conclusion

In this work we presented a training procedure for phrase or rule tables in statistical machine translation. It is based on force-aligning the training data with a modified version of the translation decoder. Different from previous work, we completely avoid the use of a word alignment on the bilingual training corpus. Instead, we initialize the procedure with an empty phrase table and generate all possible phrases on-the-fly through the concept of *backoff phrases*. Starting with a maximum phrase length of $m = 1$, we increment m in each iteration, until we reach m_{max}. Then, we continue training in a more conventional fashion, allowing creation of new phrases only in fallback runs. As additional extensions to previous work we introduce *fallback decoding runs* for higher coverage of the data and *local language models* for better pre-selection of phrases. The effects of the most important hyper parameters of our procedure are discussed and we show how they were selected in our setup. The experiments are carried out with a phrase-based decoder on the IWSLT 2011 Arabic-English shared task. The trained phrase table slightly outperforms our state-of-the-art baseline and a linear interpolation yields an improvement of 0.5% BLEU and 0.5% TER. The BLEU improvement on `test` is statistically significant with 90% confidence. The small overlap of 18.5% between the trained and the heuristically extracted phrase table shows how initialization or restrictions based on word alignments would have biased the training procedure. We also analyzed the distribution of entropies within the phrase tables, confirming the previous observation that fewer near-zero entropy phrases are advantageous for decoding. We also showed that, in our setup, near-zero entropies are largely eliminated by phrase table interpolation.

In future work we plan to apply this technique as a more principled way to train a wider range of models similar to (Duan et al., 2012). But even for the phrase models, we have only scratched the surface of its potential. We hope that by finding a meaningful way to set the hyper parameters of our training procedure, better and smaller phrase tables can be created.

Acknowledgments

This work was partially realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The material is also partially based upon work supported by the DARPA BOLT project under Contract No. HR0011- 12-C-0015. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Alexandra Birch, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the Phrase-Based, Joint Probability Statistical Translation Model. In *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, pages 154–157, New York City, NY, June.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of*

the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2, ACL '09, pages 782–790, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June.

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.

Nan Duan, Mu Li, and Ming Zhou. 2012. Forced derivation tree based model training to statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 445–454, Jeju Island, Korea, July. Association for Computational Linguistics.

Xiaodong He and Li Deng. 2012. Maximum Expected BLEU Training of Phrase and Lexicon Translation Models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 292–301, Jeju, Republic of Korea, Jul.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-03)*, pages 127–133, Edmonton, Alberta.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain, July.

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 *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pages 133–139, Philadelphia, PA, July.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden, July. Association for Computational Linguistics.

Robert C. Moore and Chris Quirk. 2007. An Iteratively-Trained Segmentation-Free Phrase Translation Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 112–119, Prague, June.

Markos Mylonakis and Khalil Sima'an. 2008. Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 630–639, Honolulu, October.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 632–641, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Franz J. Och. 2003. Minimum Error Rate Training for Statistical Machine Translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. IBM Research Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, September.

Markus Saers and Dekai Wu. 2011. Principled induction of phrasal bilexica. In *Proceedings of the 15th International Conference of the European Association for Machine Translation*, pages 313–320, May.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, pages 475–484, Uppsala, Sweden, July.

Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012a. Jane 2: Open source phrase-based and hierarchical statistical machine translation. In *International Conference on Computational Linguistics*, pages 483–491, Mumbai, India, December.

Joern Wuebker, Mei-Yuh Hwang, and Chris Quirk. 2012b. Leave-One-Out Phrase Model Training for Large-Scale Deployment. In *Proceedings of the NAACL 2012 Seventh Workshop on Statistical Machine Translation*, pages 460–467, Montreal, Canada, June.

Richard Zens. 2008. *Phrase-based Statistical Machine Translation: Models, Search, Training*. Ph.D. thesis, Computer Science Department, RWTH Aachen – University of Technology, Germany, February.