Phrase Translation Probabilities with ITG Priors and Smoothing as Learning Objective

Markos Mylonakis
Language and Computation, ILLC
Faculty of Science
University of Amsterdam
m.mylonakis@uva.nl

Khalil Sima'an
Language and Computation, ILLC
Faculty of Science
University of Amsterdam
k.simaan@uva.nl

Abstract

The conditional phrase translation probabilities constitute the principal components of phrase-based machine translation systems. These probabilities are estimated using a heuristic method that does not seem to optimize any reasonable objective function of the word-aligned, parallel training corpus. Earlier efforts on devising a better understood estimator either do not scale to reasonably sized training data, or lead to deteriorating performance. In this paper we explore a new approach based on three ingredients (1) A generative model with a prior over latent segmentations derived from Inversion Transduction Grammar (ITG), (2) A phrase table containing all phrase pairs without length limit, and (3) Smoothing as learning objective using a novel Maximum-A-Posteriori version of Deleted Estimation working with Expectation-Maximization. Where others conclude that latent segmentations lead to overfitting and deteriorating performance, we show here that these three ingredients give performance equivalent to the heuristic method *on reasonably sized training data*.

1 Motivation

A major component in phrase-based statistical Machine translation (PBSMT) (Zens et al., 2002; Koehn et al., 2003) is the table of conditional probabilities of phrase translation pairs. The pervading method for estimating these probabilities is a simple heuristic based on the relative frequency of the phrase pair *in the multi-set of the phrase pairs extracted from the word-aligned corpus* (Koehn et al., 2003). While this heuristic estimator gives good empirical results, it does not seem to optimize any intuitively reasonable objective function of the (word-aligned) parallel corpus (see e.g., (DeNero et al., 2006)) The mounting number of efforts attacking this problem over the last few years (DeNero et al., 2006; Marcu and Wong, 2002; Birch et al., 2006; Moore and Quirk, 2007; Zhang et al., 2008) exhibits its difficulty. So far, none has lead to an alternative method that performs as well as the heuristic on *reasonably sized data (approx. 1000k sentence pair)*.

Given a parallel corpus, an estimator for phrase-tables in PBSMT involves two interacting decisions (1) which phrase pairs to extract, and (2) how to assign probabilities to the extracted pairs. The heuristic estimator employs word-alignment (Giza++) (Och and Ney, 2003) and a few thumb rules for defining phrase pairs, and then extracts a multi-set of phrase pairs and estimates their conditional probabilities based on the counts in the multi-set. Using this method for extracting a set of phrase pairs, (DeNero et al., 2006; Moore and Quirk, 2007) aim at defining a better estimator for the probabilities. Generally speaking, both efforts report deteriorating translation performance relative to the heuristic.

Instead of employing word-alignment to guide phrase pair extraction, it is theoretically more appealing to aim at phrase alignment as part of the estimation process (Marcu and Wong, 2002; Birch et al., 2006). This way, phrase pair extraction goes hand-in-hand with estimating the probabilities. However, in practice, due to the huge number of possible phrase pairs, this task is rather challenging, both computationally and statistically. It is hard to define

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 630–639,
Honolulu, October 2008.

both a manageable phrase pair translation model and a well-founded training regime that would scale up to reasonably sized parallel corpora (see e.g., (Birch et al., 2006)). It remains to be seen whether this theoretically interesting approach will lead to improved phrase probability estimates.

In this paper we also start out from a standard phrase extraction procedure based on word-alignment and aim solely at estimating the conditional probabilities for the phrase pairs and their reverse translation probabilities. Unlike preceding work, we extract *all phrase pairs* from the training corpus and estimate their probabilities, i.e., without limit on length. We present a novel formulation of a conditional translation model that works with a *prior over segmentations* and a bag of conditional phrase pairs. We use binary Synchronous Context-Free Grammar (bSCFG), based on Inversion Transduction Grammar (ITG) (Wu, 1997; Chiang, 2005a), to define the set of eligible segmentations for an aligned sentence pair. We also show how the number of spurious derivations per segmentation in this bSCFG can be used for devising a prior probability over the space of segmentations, capturing the bias *in the data* towards monotone translation. The heart of the estimation process is a new *smoothing estimator*, a penalized version of Deleted Estimation, which averages the temporary **probability estimates** of multiple parallel EM processes at each joint iteration.

For evaluation we use a state-of-the-art baseline system (Moses) (Hoang and Koehn, 2008) which works with a log-linear interpolation of feature functions optimized by MERT (Och, 2003). We simply substitute our own estimates for the heuristic phrase translation estimates (both directions and the phrase penalty score) and compare the two within the Moses decoder. While our estimates differ substantially from the heuristic, their performance is on par with the heuristic estimates. This is remarkable given the fact that comparable previous work (DeNero et al., 2006; Moore and Quirk, 2007) did not match the performance of the heuristic estimator using large training sets. We find that smoothing is crucial for achieving good estimates. This is in line with earlier work on consistent estimation for similar models (Zollmann and Sima'an, 2006), and agrees with the most up-to-date work that employs Bayesian priors over the estimates (Zhang et al., 2008).

2 Related work

Marcu and Wong (Marcu and Wong, 2002) realize that the problem of extracting phrase pairs should be intertwined with the method of probability estimation. They formulate a joint phrase-based model in which a source-target sentence pair is generated jointly. However, the huge number of possible *phrase-alignments* prohibits scaling up the estimation by Expectation-Maximization (EM) (Dempster et al., 1977) to large corpora. Birch et al (Birch et al., 2006) provide soft measures for including word-alignments in the estimation process and obtain improved results only on small data sets.

Coming up-to-date, (Blunsom et al., 2008) attempt a related estimation problem to (Marcu and Wong, 2002), using the expanded phrase pair set of (Chiang, 2005a), working with an exponential model and concentrating on marginalizing out the latent segmentation variable. Also most up-to-date, (Zhang et al., 2008) report on a multi-stage model, *without* a latent segmentation variable, but with a strong prior preferring sparse estimates embedded in a Variational Bayes (VB) estimator and concentrating the efforts on pruning both the space of phrase pairs and the space of (ITG) analyses. The latter two efforts report improved performance, albeit again on a limited training set (approx. 140k sentences up to a certain length).

DeNero et al (2006) have explored estimation using EM of phrase pair probabilities under a conditional translation model based on the original source-channel formulation. This model involves a hidden segmentation variable that is set uniformly (or to prefer shorter phrases over longer ones). Furthermore, the model involves a reordering component akin to the one used in IBM model 3. Despite this, the heuristic estimator remains superior because "EM learns overly determinized segmentations and translation parameters, overfitting the training data and failing to generalize". More recently, (Moore and Quirk, 2007) devise a estimator working with a model that does not include a hidden segmentation variable but works with a heuristic iterative procedure (rather than MLE or EM). The

translation results remain inferior to the heuristic but the authors note an interesting trade-off between decoding speed and the various settings of this estimator.

Our work expands on the general approach taken by (DeNero et al., 2006; Moore and Quirk, 2007) but arrives at insights similar to those of the most recent work (Zhang et al., 2006), albeit in a completely different manner. The present work differs from all preceding work in that it employs the set of *all phrase pairs* during training. It differs from (Zhang et al., 2008) in that it does postulate a latent segmentation variable and puts the prior directly over that variable rather than over the ITG synchronous rule estimates. Our method neither excludes phrase pairs before estimation nor does it prune the space of possible segmentations/analyses during training/estimation. As well as smoothing, we find (in the same vein as (Zhang et al., 2008)) that setting effective priors/smoothing is crucial for EM to arrive at better estimates.

3 The Translation Model

Given a word-aligned parallel corpus of source-target sentences, it is common practice to extract a set of phrase pairs using extraction heuristics (cf. (Koehn et al., 2003; Och and Ney, 2004)). These heuristics define a phrase pair to consist of a source and target ngrams of a word-aligned source-target sentence pair such that if one end of an alignment is in the one ngram, the other end is in the other ngram (and there is at least one such alignment) (Och and Ney, 2004; Koehn et al., 2003). For efficiency and sparseness, the practitioners of PBSMT constrain the length of the source phrase to a certain maximum number of words.

An All Phrase Pairs Model: In this work we train a phrase-translation table that consists of *all phrase-pairs* that can be extracted from the word-aligned training data according to the standard phrase extraction heuristic. After training, we can still limit the set of phrase pairs to those selected by a cut-off on phrase length. The reason for using all phrase pairs during training is that it gives a clear point of reference for an estimator, without implicit, accidental biases that might emerge due to length cut-off[1].

The Generative Model: Given a word-aligned source-target sentence pair $\langle \mathbf{f}, \mathbf{e}, \mathbf{a} \rangle$, the generative story underlying our model goes as follows:

1. Abiding by the word-alignments in $\mathbf{a}$, segment the source-target sentence pair $\langle \mathbf{f}, \mathbf{e} \rangle$ into a sequence of I containers σ_1^I, and a bag of I phrase pairs $\sigma_1^I(\mathbf{f}, \mathbf{e}) = \{\langle f_j, e_j \rangle\}_{j=1}^I$. Each container $\sigma_j = \langle l_f, r_f, l_e, r_e \rangle$ consists of the start l_f and end r_f positions[2] for a phrase in $\mathbf{f}$ and the start l_e and end r_e positions for an aligned phrase in $\mathbf{e}$.

2. For a given segmentation σ_1^I, for every container σ_j ($1 \leq j \leq I$) generate the phrase-pair $\langle f_j, e_j \rangle$, independently from all other phrase-pairs.

This leads to the following probabilistic model:

$$P(\mathbf{f} \mid \mathbf{e}; \mathbf{a}) = \sum_{\sigma_1^I \in \Sigma(\mathbf{a})} P(\sigma_1^I) \prod_{\langle f_j, e_j \rangle \in \sigma_1^I(\mathbf{f}, \mathbf{e})} P(f_j \mid e_j) \quad (1)$$

Where $\Sigma(\mathbf{a})$ is the set of *binarizable* segmentations (defined next) that are eligible according to the word-alignments $\mathbf{a}$ between $\mathbf{f}$ and $\mathbf{e}$. These segmentations into bilingual containers (where segmentations are taken inside the containers) are different from the monolingual segmentations used in earlier comparable conditional models (e.g., (DeNero et al., 2006)) which must generate the alignment on top of the segmentations. Note how the different phrase pairs $\langle f_j, e_j \rangle$ are generated from their bilingual containers in the given segmentation σ_1^I. We will discuss our choice of prior probability over segmentations $P(\sigma_1^I)$ after we discuss the definition of the binarizable segmentations $\Sigma(\mathbf{a})$.

3.1 Binarizable segmentations $\Sigma(\mathbf{a})$

Following (Zhang et al., 2006; Huang et al., 2008), every sequence of *phrase alignments* can be viewed

[1] For example, if the cut-off on phrase pairs is ten words, all sentence pairs smaller than ten words in the training data will be included as phrase pairs as well. These sentences are treated differently from longer sentences, which are not allowed to be phrase pairs.

[2] The NULL alignments (word-to-NULL) in the training data can also be marked with actual positions on both sides in order to allow for this definition of containers.

as a sequence of integers $1, \ldots I$ together with a permuted version of this sequence $\pi(1), \ldots, \pi(I)$, where the two copies of an integer in the two sequences are assumed aligned/paired together. For example, possible permutations of $\{1, 2, 3, 4\}$ are $\{2, 1, 3, 4\}$ and $\{2, 4, 1, 3\}$. Because a segmentation σ_1^I of a sentence pair is also a sequence of aligned phrases, it also constitutes a permuted sequence. A binarizable permutation $\mathbf{x}$ is either of length one, or can be **properly split** into two binarizable sub-sequences $\mathbf{y}$ and $\mathbf{z}$ such that either[3] $\mathbf{z} < y$ or $\mathbf{y} < z$. For example, one way to binarize the permutation $\{2, 1, 3, 4\}$ is to introduce a proper split into $\{2, 1; 3, 4\}$, then recursively another proper split of $\{2, 1\}$ into $\{2; 1\}$ and $\{3, 4\}$ into $\{3; 4\}$. In contrast, the permutation $\{2, 4, 1, 3\}$ is non-binarizable.

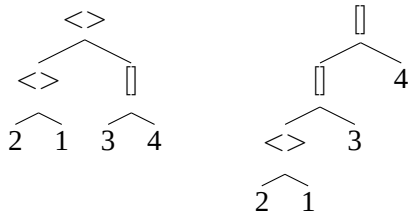

Figure 1: Multiple ways to binarize a permutation

Graphically speaking, the recursive definition of binarizable permutations can be depicted as a binary tree structure where the nodes correspond to recursive proper splits of the permutation, and the leaves are decorated with the naturals. Figure 1 exhibits two possible binarizations of the same permutation where $<>$ and [] denote inverted and monotone proper splits respectively. Note that the number of possible binarizations of a binarizable permutation is a recursive function of the number of possible proper splits and reaches its maximum for fully monotone permutations (all binary trees, which is a factorial function of the length of the permutation).

By definition (cf. (Zhang et al., 2006; Huang et al., 2008)), a binarizable segmentation/permutation can be recognized by a binarized Synchronous Context-Free Grammar (SCFG), i.e., an SCFG in which the right hand sides of all non-lexical rules constitute binarizable permutations. In particular, this holds for the SCFG implementing Inversion Transduction Grammar (Wu, 1997). This SCFG (Chiang, 2005b) has two binary synchronous rules that correspond resp. to the contiguous monotone and inverted alignments:

$$\begin{aligned} XP &\rightarrow XP^{\boxed{1}}\, XP^{\boxed{2}}, \;\; XP^{\boxed{1}}\, XP^{\boxed{2}} \\ XP &\rightarrow XP^{\boxed{1}}\, XP^{\boxed{2}}, \;\; XP^{\boxed{2}}\, XP^{\boxed{1}} \end{aligned} \tag{2}$$

The boxed integers in the superscripts on the nonterminal XP denote synchronized rewritings. In this work, we employ a binary SCFG (bSCFG) working with these two synchronous rules together with a set of lexical rules $\{XP \rightarrow f,\ e \mid \langle f, e\rangle\ is\ a\ phrase\ pair\}$.

In this bSCFG, every derivation corresponds to a binarization of a segmentation of the input. Note that the bSCFG defined in equation 2 generates all possible binarizations for every segmentation of the input. It is possible to constrain this bSCFG such that it generates a single, canonical derivation per segmentation. However, in section 3.2 we show that the number of such derivations is a good measure of phrase pair productivity.

It is well known that there are alignments and segmentations that this bSCFG does not cover (see (Huang et al., 2008)). Recently, strong evidence emerged (e.g., (Huang et al., 2008)) showing that most word-alignments of actual parallel corpora can be covered by a binarized SCFG of the ITG type. Furthermore, because our model employs the set of *all phrase-pairs* that can be extracted from a given training set, it will always find segmentations that cover every sentence pair in the training data[4]. This implies that while our model might discard non-binarizable segmentations for certain complex word alignments, we do manage to train the model on the binarizable segementations of all sentence pairs.

Up to the prior over segmentations (see next), we implement the above model using a weighted version of the binary SCFG as follows:

- The weight for lexical rules is given by $P(XP \rightarrow f, e) := P(f \mid e)$, where $\langle f, e\rangle$ is a phrase-pair. These are the trainable parameters of our model.

[3]For two sequences of numbers, the notation $\mathbf{y} < z$ stands for $\forall y \in \mathbf{y}, \forall z \in \mathbf{z} : y < z$.

[4]In the worst case the whole sentence pair is a phrase pair with a trivial segmentation.

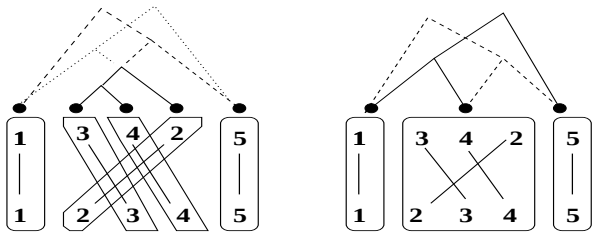

Figure 2: Two segmentations of an alignment/permutation. Both segmentations have the same number of binarizations despite differences in container sizes.

- The weights for the two non-lexical rules in equation 2 are fixed at 1.0. These weights are not trained at all.

Where we use the notation $P(.)$ for the weight of a synchronous rule.

3.2 Prior over segmentations

As it has been found out by (DeNero et al., 2006), it is not easy to come up with a simple, effective prior distribution over segmentations that allows for improved phrase pair estimates. Within a Maximum-Likelihood estimator, preference for segmentations σ_1^I consisting of longer containers could lead to overfitting as we will explain in section 4. Alternatively, it is tempting to have preference for segmentations σ_1^I that consist of shorter containers, because (generally speaking) shorter containers have higher expected coverage of new sentence pairs. However, mere bias for shorter containers will not give better estimates as observed by (DeNero et al., 2006). One case where this bias clearly fails is the case of a contiguous sequence of containers with a complex alignment structure (crossing alignments). For example (see figure 2), for the alignment $\{1, 3, 4, 2, 5\}$ there is a segmentation into five containers $\{1; 3; 4; 2; 5\}$, and another into three $\{1; 3, 4, 2; 5\}$. The first segmentation involves shorter containers that have crossing brackets among them, while the second one consists of three containers including a longer container $\{3, 4, 2\}$. In the first segmentation, due to their crossing alignments, each of the containers $\{3\}$, $\{4\}$ and $\{2\}$ will not combine with the surrounding context ($\{1\}$ and $\{5\}$) on its own, i.e., without the other two containers. Furthermore, there is only a single binarization of $\{3, 4, 2\}$. Hence, while the first segmentation involves shorter containers than the second one, these shorter containers are as *productive* as the large container $\{3, 4, 2\}$, i.e., they combine with surrounding containers in the same number of ways as the large container. In such and similar cases, there are no grounds for the bias towards shorter phrases/containers.

The notion of *container productivity* (the number of ways in which it combines with surrounding containers during training) seems to correlate with the expected number of ways a container can be used during decoding, which should be correlated with expected coverage. During training, containers that are often surrounded by other, monotonically aligned containers are expected to be more productive than alternative containers that are often surrounded by crossing alignments. Hence, the number of binarizations that a segmentation has under the bSCFG is a direct function of the ways in which the containers combine among themselves (monotone vs. inverted/crossing) within segmentations, and provides a more accurate measure of container productivity than container length. Hence, the final model we employ is the following:

$$P(\mathbf{f} \mid \mathbf{e}; \mathbf{a}) = \sum_{\sigma_1^I \in \Sigma(\mathbf{a})} \frac{N(\sigma_1^I)}{Z(\Sigma(\mathbf{a}))} \prod_{\langle f_j, e_j \rangle \in \sigma_1^I(\mathbf{f},\mathbf{e})} P(f_j \mid e_j) \quad (3)$$

Where $N(\sigma_1^I)$ is the number of binary derivations/trees that σ_1^I has in the binary SCFG (bSCFG), and $Z(\Sigma(\mathbf{a})) = \sum_{\sigma_1^J \in \Sigma(\mathbf{a})} N(\sigma_1^J)$, i.e., this prior is the ratio of number of derivations of σ_1^I to the total number of derivations that $\langle \mathbf{f}, \mathbf{e}, \mathbf{a} \rangle$ has under the bSCFG.

3.3 Contrast with similar models:

In contrast with the model of (DeNero et al., 2006), who define the segmentations over the source sentence $\mathbf{f}$ alone, our model employs bilingual containers thereby segmenting both source and target sides simultaneously. Therefore, unlike (DeNero et al., 2006), our model does not need to generate the word-alignments explicitly, as these are embedded in the segmentations. Similarly, our model does not include *explicit* penalty terms for reorder-

ing/inversion but includes a related bias in the prior probabilities over segmentations $P(\sigma_1^I)$.

In a way, the segmentations and bilingual containers we use can be viewed as similar to the concepts used in the Joint Model of Marcu and Wong (Marcu and Wong, 2002). Unlike (Marcu and Wong, 2002), however, our model works with conditional probabilities and starts out from the word-alignments.

The novel aspects of our model are three (1) It defines the set of segmentations using a bSCFG, (2) It includes a novel, refined prior probability over segmentations, and (3) It employs all phrase pairs that can be extracted from a word-aligned training parallel corpus. For these novel elements to produce reasonable estimates, we devise our own estimator.

4 Estimation by Smoothing

In principle, we are dealing here with a translation model that employs all phrase pairs (of unbounded size), extracted from a word-aligned parallel corpus. Under this model, where a phrase pair and its sub-phrase pairs are included in the model, the MLE can be expected to overfit the data[5] unless a suitable prior probability over segmentations is employed. Indeed, the prior over segmentations defined in the preceding section prevents the MLE from completely overfitting the training data. However, we find empirical evidence that this prior is insufficient for avoiding overfitting.

Our model behaves like a *memory-based model* because it memorizes all extractable phrase pairs found in the training data including the training sentence pairs themselves. Such memory-based models are related to nonparametric models such as K-NN and kernel methods (Hastie et al., 2001). For memory-based models, consistent estimation for novel instances proceeds by local density estimation from the surroundings of the instance, which is akin to smoothing for parametric models. Hence, next we describe our own version of a *smoothed* Maximum-Likelihood estimator for phrase translation probabilities.

[5]One trivial MLE solution would give the longest container, consisting of the longest phrase pairs, a probability of one, at the cost of all shorter alternatives. A similar problem arises in Data-Oriented Parsing, see (Sima'an and Buratto, 2003; Zollmann and Sima'an, 2006). Note that models that employ an upperbound on phrase pair length will still risk overfitting training sentences of lengths that fall within this upperbound.

INPUT: Word-aligned parallel training data T
OUTPUT: Estimates π for all $P(f \mid e)$
{
Split training data T into equal parts $H_1, \ldots, H_{10}$.
For $1 \leq i \leq 10$ **do**
  Extract from $E_i = \cup_{j \neq i} H_j$ all phrase pairs π_i
  Initialize $\widehat{\pi}_i^0$ to uniform conditional probs
Let $j = 0$
Repeat
  Let $j = j + 1$ // EM iteration counter
  For $1 \leq i \leq 10$ **do**
    E-step: calculate expected counts for pairs in π_i^j on H_i using counts from $\widehat{\pi}_i^{j-1}$.
    M-step: calculate probabilities for pairs in π_i^j from the expected counts
  For $1 \leq i \leq 10$ **do** $\widehat{\pi}_i^j := \frac{1}{10} \sum_{i=1}^{10} \pi_i^j$
Until $\pi := \{\widehat{\pi}_1^j, \ldots, \widehat{\pi}_{10}^j\}$ has converged
}

Figure 3: Penalized Deleted Estimation

For a latent variable model, it is usually common to employ Expectation-Maximization (EM) (Dempster et al., 1977) as a search method for a (local) maximum-likelihood estimate (MLE) of the training data. Instead of mere EM we opt for a *smoothed* version: we present a new method, that combines Deleted Estimation (Jelinek and Mercer, 1980) with the Jackknife (Duda et al., 2001) as the core estimator.

Figure 3 shows the pseudo-code for our estimator. Like in Deleted Estimation, we split the training data into ten equal portions. This way we create ten different splits of *extraction/heldout sets* of respectively 90%/10% of the training set. For every split $1 \leq i \leq 10$, we extract a set of phrase pairs π_i from the *extraction* set E_i and train it (under our model) on the *heldout set* H_i. Naturally, the phrase pair sets π_i ($1 \leq i \leq 10$) are subsets of (or equal to) the set of phrase pairs $\pi = \cup_i \pi_i$ extracted from the total training data (i.e., π is the set of model parameters). The training of the different π_i's, each on its corresponding heldout set H_i, is done by ten separate EM processes, which are synchronized in their initializa-

tion, their iterations as well as stop condition. The EM processes start out from uniform conditional estimates of the phrase pairs in all π_i. After every EM iteration j, when the M-step has finished, the estimates in all π_i^j ($1 \leq i \leq 10$) are set to the average (over $1 \leq i \leq 10$) of the estimates in π_i^j leading to $\widehat{\pi}_i^j$ (following the Jackknife method). The resulting averaged probabilities in $\widehat{\pi}_i^j$ are then used as the current phrase pair estimates, which feed into the next iteration $j+1$ of the different EM processes (each working on a different heldout set H_i with a different set of phrase pairs π_i).

There are two special boundary cases which demand special attention during estimation:

Sparse distributions: For a phrase e that does occur both in H_i and E_i, there could be a phrase pair $\langle f, e \rangle$ that does occur in H_i but does *not* occur in π_i. To prevent EM from giving the extra probability mass to all other pairs $\langle f, e' \rangle$ unjustifiably, we apply smoothing. We add the missing pair $\langle f, e \rangle$ to π_i and set its probability to a fixed number 10^{-5*len}, where len is the length of the phrase pair. In effect, we backoff our model (equation 1) to a word-level model with fixed word translation probability (10^{-5}).

Zero distributions: When a phrase e does not occur in H_i, all its pairs $\langle f, e \rangle$ in π_i will have zero counts. During each EM iteration, when the M-step is applied, the distribution $P(\cdot \mid e)$ is undefined by MLE, since it is irrelevant for the likelihood of H_i. In this case any choice of proper distribution $P(\cdot \mid e)$ will constitute an MLE solution. We choose to set this case to a uniform distribution every time again.

Since our model and estimator are implemented within the bSCFG framework, we use a bilingual CYK parser (Younger, 1967) under the grammar in equation 2. This parser builds for every input $\langle \mathbf{f}, \mathbf{a}, \mathbf{e} \rangle$ all binarizations/derivations for every segmentation in $\Sigma(\mathbf{a})$. For implementing EM, we employ the Inside-Outside algorithm (Lari and Young, 1990; Goodman, 1998). During estimation, because the input, output and word-alignment are known in advance, the time and space requirements remain manageable despite the worst-case complexity $O(n^6)$ in target sentence length n.

Penalized Deleted Estimation: In contrast with our method, Deleted Estimation sums the *expected counts* (rather than probabilities) obtained from the different splits before applying the M-step (normalization). While the rationale behind Deleted Estimation comes from MLE over the original training data, our method has a smoothing objective (inspired by the Jackknife): generally speaking, the averages over different heldout sets (under different subsets of the model) give less sharp estimates than MLE. By averaging the different heldout estimates, this estimator employs a penalty term that depends on the marginal count of e in the heldout set[6]. Interestingly, when the phrase e is very frequent[7], it will approximately occur almost as often in the different heldout sets. In this case, our method reduces to Deleted Estimation, where it effectively sums the counts[8]. Yet, when the target phrase e does occur only very few times, it is likely that its count in some splits will be zero. In our method, at every EM iteration, during the Maximization step, we set such cases back to uniform. By averaging the probabilities from the different splits over many EM iterations, setting these cases to uniform constitutes a kind of prior that prevents the final estimates from falling too far from uniform. In contrast, in Deleted Interpolation the zero counts are simply summed with the other corresponding counts of the same phrase pair, which leads to sharper probability distributions. In all experiments that we conducted, our method (which we call *Penalized Deleted Estimation*) gave more successful estimates than mere Deleted Estimation.

On the theoretical side, the choice for a fixed

[6]Define $count_y(x)$ to be the count of event x in data y. The Deleted Estimation (DE) estimate is $\sum_H count_H(f,e)/count_T(e)$, which can be written as $\sum_H [count_H(f,e)/count_H(e)][count_H(e)/count_T(e)] = \sum_H \pi_H(f|e)[count_H(e)/count_T(e)]$ where $\pi_H(f|e)$ is the estimate from heldout set H. Hence, DE linearly interpolated π_H with factors $count_H(e)/count_T(e)$. Our estimator employs uniform interpolation factors instead, thereby penalizing the DI counts (hence Penalized DI).

[7]Theoretically speaking, when the training data is unboundedly large, our estimator will converge to the same estimates as the Deleted Estimation. When the data is still sparse, our estimator is biased, unlike the MLE which will overfit.

[8]When calculating the conditional probabilities, the denominators used are approximately equal to one another.

prior over segmentations (ITG prior) implies that our model cannot be estimated to converge (in probability) to the relative frequency estimates (RFE) of source-target sentence pairs in the limit of the training data (a sufficiently large parallel corpus). A prior probability over segmentations that would allow our estimator to converge in the limit to the RFE must gradually prefer segmentations consisting of larger containers as the data grows large. We set the design and estimation of such a prior aside for future work.

5 Empirical experiments

Decoding and Baseline Model: In this work we employ an existing decoder, Moses (Hoang and Koehn, 2008), which defines a log-linear model interpolating feature functions, with interpolation scores λ_f $\mathbf{e}^* = \arg\max_{\mathbf{e}} \sum_{f \in \Phi} \lambda_f H_f(\mathbf{f}, \mathbf{e})$. The λ_f are optimized by Minimum-Error Training (MERT) (Och, 2003). The set Φ consists of the following feature functions (see (Hoang and Koehn, 2008)): a 5-gram target language model, the standard reordering scores, the word and phrase penalty scores, the conditional lexical estimates obtained from the word-alignment in both directions, and the conditional phrase translation estimates in both directions $P(f \mid e)$ and $P(e \mid f)$. Keeping the other five feature functions fixed, we compare our estimates of $P(f \mid e)$ and $P(e \mid f)$ (and the phrase penalty) to the commonly used heuristic estimates.

Because our model employs a latent segmentation variable, this variable should be marginalized out during decoding to allow selecting the highest probability translation given the input. This turns out crucial for improved results (cf. (Blunsom et al., 2008)). However, such a marginalization can be NP-Complete, in analogy to a similar problem in Data-Oriented Parsing (Sima'an, 2002)[9]. We do not have a decoder yet that can approximate this marginalization efficiently and we employ the standard Moses decoder for this work.

Experimental Setup: The training, development and test data all come from the French-English translation shared task of the ACL 2007 Second Workshop on Statistical Machine Translation [10]. After pruning sentence pairs with word length more than 40 on either side, we are left with 949K sentence pairs for training. The development and test data are composed of 2K sentence pairs each. All data sets are lower-cased.

For both the baseline system and our method, we produce word-level alignments for the parallel training corpus using GIZA++. We use 5 iterations of each IBM Model 1 and HMM alignment models, followed by 3 iterations of each Model 3 and Model 4. From this aligned training corpus, we extract the phrase pairs according to the heuristics in (Koehn et al., 2003). The baseline system extracts all phrase-pairs upto a certain maximum length on both sides and employs the heuristic estimator. The language model used in all systems is a 5-gram language model trained on the English side of the parallel corpus. Minimum-Error Rate Training (MERT) is applied on the development set to obtain optimal log-linear interpolation weights for all systems. Performance is measured by computing the BLEU scores (Papineni et al., 2002) of the system's translations, when compared against a single reference translation per sentence.

Phrases	System	BLEU
≤ 7	Baseline PBSMT	33.03
≤ 10	Baseline PBSMT	33.03
All	Baseline PBSMT	33.00
≤ 7	EM + ITG Prior	32.50
≤ 7	EM + Del. Est.	32.67
≤ 7	EM + Del. Est. + ITG Prior	32.73
≤ 7	EM + Pen. Del. Est. + ITG Prior	33.02
≤ 10	EM + Pen. Del. Est. + ITG Prior	**33.14**
All	EM + Pen. Del. Est. + ITG Prior	32.98

Table 1: Results: data from ACL07 2^{nd} Wkshp on SMT

Results: We compare different versions of our system against the baseline system using the heuristic estimator. We observe the effects of the ITG prior in the translation model as well as the method of estimation (Deleted Estimation vs. Penalized Deleted Estimation).

Table 1 exhibits the BLEU scores for the sys-

[9] A reduction of simple instances of the first problem to instances of the latter problem should be possible.

[10] http://www.statmt.org/wmt07

tems. Our own system (with ITG prior and Penalized Deleted Estimation and maximum phrase-length ten words) scores (33.14), slightly outperforming the best baseline system (33.03). When using straight Deleted Estimation over EM, this leads to deterioration (32.73). When also the ITG prior is excluded (by having a single derivation per segmentation) this leads to further deterioration (32.67). By using mere EM with an ITG prior, performance goes down to 32.50, exhibiting the crucial role of the estimation by smoothing. Clearly, Penalized Deleted Estimation and the ITG prior are important for the improved phrase translation estimates.

As table 1 shows we also varied the phrase length cutoff (seven, ten or none=all phrase pairs). The length cutoff pertains to both sides of a phrase-pair. For our estimator, we always train all phrase pairs, applying the length cutoff only after training (no renormalization is applied at that point).

Interestingly, we find out that the heuristic estimator cannot benefit performance by including longer phrase pairs. Our estimator does benefit performance by including phrase pairs of length upto ten words, but then it degrades again when including all phrase pairs. We take the latter finding to signal remaining overfitting that proved resistant to the smoothing applied by our estimator. The heuristic estimator exhibits a similar degradation.

We also tried to vary the treatment of Sparse Distributions (section 4, page 7) during heldout estimation from fixed word-translation probabilities to the lexical model probabilities. This lead to slight deterioration of results (32.94). It is unclear whether this deterioration is meaningful or not. We did not explore mere EM without any smoothing or ITG prior, as we expect it will directly overfit the training data as reported by (DeNero et al., 2006).

We note that for French-English translation it is hard to outperform the heuristic within the PBSMT framework, since it already performs very well. Preliminary, most recent experiments on German-English (also WMT07 data) exhibit that our estimator outperforms the heuristic.

6 Discussion and Future Research

The most similar efforts to ours, mainly (DeNero et al., 2006), conclude that segmentation variables in the generative translation model lead to overfitting while attaining higher likelihood of the training data than the heuristic estimator. Based on this advise (Moore and Quirk, 2007) exclude the latent segmentation variables and opt for a heuristic training procedure. In this work we also start out from a generative model with latent segmentation variables. However, we find out that concentrating the learning effort on smoothing is crucial for good performance. For this, we devise ITG-based priors over segmentations and employ a penalized version of Deleted Estimation working with EM at its core. The fact that our results (at least) match the heuristic estimates on a reasonably sized data set (947k parallel sentence pairs) is rather encouraging.

The work in (Zhang et al., 2008) has a similar flavor to our work, yet the two differ substantially. Both depart from Maximum-Likelihood towards non-overfitting estimators. Where Zhang et al choose for sparse priors (leading to sharp phrase distributions) and put the smoothing burden on the ITG rule parameters and a pruning strategy, we choose for a prior over segmentations determined by the ITG derivation space and smooth the MLE directly with a penalized version of Deleted Estimation. It remains to be seen how the two biases compare to one another on the same task.

There are various strands of future research. Firstly, we plan to explore our estimator on other language pairs in order to obtain more evidence on its behavior. Secondly, as (Blunsom et al., 2008) show, marginalizing out the different segmentations during decoding leads to improved performance. We plan to build our own decoder (based on ITG) where different ideas can be tested including tractable ways for achieving a marginalization effect. Apart from a new decoder, it will be worthwhile adapting the prior probability in our model to allow for consistent estimation. Finally, it would be interesting to study properties of the penalized Deleted Estimation used in this paper.

Acknowledgments: Both authors are supported by a VIDI grant (nr. 639.022.604) from The Netherlands Organization for Scientific Research (NWO). David Chiang and Andy Way are acknowledged for stimulating discussions on machine translation and parsing.

References

A. Birch, Ch. Callison-Burch, M. Osborne, and Ph. Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 154–157. Association for Computational Linguistics.

P. Blunsom, T. Cohn, and M. Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 200–208. Association for Computational Linguistics.

D. Chiang. 2005a. A hierarchical phrase-based model for statistical machine translation. In *In Proceedings of ACL 2005*, pages 263–270.

D. Chiang. 2005b. An introduction to synchronous grammars. Technical report, Univeristy of Maryland.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

J. DeNero, D. Gillick, J. Zhang, and D. Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City. Association for Computational Linguistics.

R.O. Duda, P.E. Hart, and D.G. Stork. 2001. *Pattern Classification*. John Wiley & Sons, NY, USA.

J.T. Goodman. 1998. *Parsing Inside-Out*. PhD thesis, Departement of Computer Science, Harvard University, Cambridge, Massachusetts.

T. Hastie, R. Tibshirani, and J. H. Friedman. 2001. *The Elements of Statistical Learning*. Springer.

H. Hoang and Ph. Koehn. 2008. Design of the moses decoder for statistical machine translation. In *ACL Workshop on Software engineering, testing, and quality assurance for NLP 2008*.

L. Huang, H. Zhang, D. Gildea, and K. Knight. 2008. Binarization of synchronous context-free grammars. *Submitted to Computational Linguistics. http://www.cis.upenn.edu/ lhuang3/opt.pdf*.

F. Jelinek and R. L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *In Proceedings of the Workshop on Pattern Recognition in Practice*.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL*.

K. Lari and S.J. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer, Speech and Language*, 4:35–56.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of Empirical methods in natural language processing*, pages 133–139. Association for Computational Linguistics.

R. Moore and Ch. 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, Czech Republic. Association for Computational Linguistics.

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

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

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

K. Sima'an and L. Buratto. 2003. Backoff Parameter Estimation for the DOP Model. In H. Blockeel N. LavraĈ, D. Gamberger and L. Todorovski, editors, *Proceedings of the 14th European Conference on Machine Learning (ECML'03), Lecture Notes in Artificial Intelligence (LNAI 2837)*, pages 373–384, Cavtat-Dubrovnik, Croatia. Springer.

K. Sima'an. 2002. Computational complexity of probabilistic disambiguation. *Grammars*, 5(2):125–151.

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

D.H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.

R. Zens, F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In Matthias Jarke, Jana Koehler, and Gerhard Lakemeyer, editors, *KI 2002: Advances in Artificial Intelligence, 25th Annual German Conference on AI (KI 2002)*, volume 2479 of *Lecture Notes in Computer Science*, pages 18–32. Springer.

H. Zhang, L. Huang, D. Gildea, and K. Knight. 2006. Synchronous binarization for machine translation. In *HLT-NAACL*.

H. Zhang, Ch. Quirk, R. C. Moore, and D. Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

A. Zollmann and K. Sima'an. 2006. An efficient and consistent estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics (JALC)*, 10 (2005) Number 2/3:367–388.