

BETTER STATISTICAL ESTIMATION CAN BENEFIT ALL PHRASES IN PHRASE-BASED STATISTICAL MACHINE TRANSLATION

Khalil Sima'an and Markos Mylonakis

Institute for Logic, Language and Computation
Faculty of Science, University of Amsterdam
Amsterdam, The Netherlands
{k.simaan, m.mylonak}@uva.nl

ABSTRACT

The heuristic estimates of conditional phrase translation probabilities are based on frequency counts in a word-aligned parallel corpus. Earlier attempts at more principled estimation using Expectation-Maximization (EM) underperform this heuristic. This paper shows that a recently introduced novel estimator based on smoothing might provide a good alternative. When *all phrase pairs* are estimated (no length cut-off), this estimator slightly outperforms the heuristic estimator.

Index Terms— Transduction, Parameter Estimation, Smoothing Methods

1. MOTIVATION

The conditional probabilities of phrase translation pairs constitute a major component in phrase-based statistical Machine translation (PBSMT) [1, 2]. It is currently standard practice to extract a multi-set of phrase pairs of length less than an experimentally set upperbound (e.g., seven words) from a word-aligned parallel corpus [2]. The phrase probabilities are estimated by using the counts in the multi-set of extracted phrases as relative frequencies [2], leading to a heuristic estimator.

It has proven difficult to date to outperform the heuristic with a more principled estimation method, e.g., [3, 4]. DeNero et al [3] explore EM estimation under a conditional model. The model involves a latent segmentation probability, set uniformly or to prefer shorter phrases over longer ones, and a reordering component akin to IBM model 3. The heuristic estimator remains superior because "EM learns overly determined segmentations and translation parameters, overfitting the training data and failing to generalize". More recently, [4] devise a model without segmentation variables and employ a heuristic estimation procedure. Again, the translation results remain inferior to the heuristic. The alternative approaches based on [5] or [6] are related but for space reasons we will not discuss them here (see e.g., [7]).

Based on our earlier work [8], here we also start out from a standard definition of phrase pairs and aim at principled estimation of the conditional phrase probabilities. Contrary to the common approach based on findings in [2], we extract *all phrase pairs* from the training corpus (i.e., without limit on length). In this paper we consider the outstanding question of whether this choice pays off in terms of performance, relative to the heuristic estimator. Furthermore, while [8] used French-English as benchmark evaluation data, here we exhibit new experiments on German-English EuroParl data. We compare our own phrase pair probability estimates to a state-of-the-art baseline system (based on Moses [9]) by substituting these estimates instead of the heuristics (both directions). When all phrase pairs are estimated, our system performs better than the heuristic. These results on German-English provide support to our earlier results on French-English [8]. Based on these outcomes, we conclude that the principled estimation of phrase probabilities without length cut-off holds the promise of improved performance.

2. THE MODEL

The phrase pairs are extracted from a word-aligned parallel corpus using a standard method (cf. [2, 10]). Based on findings in [2], PBSMT practitioners constrain the phrase length to a certain maximum because longer phrases do not improve performance. In this work we employ **all phrase-pairs** that can be extracted from the word-aligned training data. This avoids implicit, accidental biases due to length cut-off and has a positive impact on performance as we will show empirically in the next sections.

We employ the translation model described in [8] (the target language model is estimated separately). Given a word-aligned sentence pair $\langle \mathbf{f}, \mathbf{e}, \mathbf{a} \rangle$, our model works as follows¹:

¹A container $\sigma_j = \langle l_f, r_f, l_e, r_e \rangle$ consists of the start l_f and end r_f positions for a phrase in \mathbf{f} and the start l_e and end r_e positions for an aligned phrase in \mathbf{e} . The bilingual containers are akin to the concepts in [5].

1. Abiding by word-alignments \mathbf{a} , segment 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 $\sigma_1^I(\mathbf{f}, \mathbf{e}) = \{\langle f_j, e_j \rangle\}_{j=1}^I$.
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:

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

Where $\Sigma(\mathbf{a})$ is the set of *binarizable* segmentations that are eligible under the alignments \mathbf{a} between \mathbf{f} and \mathbf{e} , and $P(\sigma_1^I)$ is the prior probability over segmentations. Both entities are defined next.

We follow [11] and use Inversion Transduction Grammar (ITG) [12] for defining the binarizable segmentations. The binarizable segmentations $\Sigma(\mathbf{a})$ are those derivable by the binary Synchronous Context-Free Grammar (bSCFG) implementing ITG [12]. This bSCFG has a set of synchronous lexical rules $\{XP \rightarrow f, e \mid f, e \text{ is a phrase pair}\}$ and only two binary glue rules: monotone $XP \rightarrow [XP XP]$ and inverted $XP \rightarrow \langle XP XP \rangle$. In this bSCFG, every derivation corresponds to a binarization of a segmentation of the input (see Figure 1). Note that this bSCFG generates all binarizations for every segmentation of the input. It is possible to constrain this bSCFG such that it generates a single, canonical derivation/binarization per segmentation. However, in the sequel (this section) we show that the number of such derivations is a good measure of phrase pair productivity.

We implement the above model using a weighted version of the bSCFG. For lexical rules the weight $P(XP \rightarrow f, e) := P(f | e)$, where $\langle f, e \rangle$ is a phrase-pair (these are the trainable parameters). We do not train the two non-lexical rules and fix their weights at 1.0.

It is tempting to have preference for segmentations σ_1^I that consist of shorter containers because those give higher coverage of new data. However, this will not give better estimates as found empirically in [3]. For example, consider the alignment $\{1, 3, 4, 2, 5\}$ (aligned with $\{1, 2, 3, 4, 5\}$) that has one segmentation into five containers $\{1; 3; 4; 2; 5\}$ and another into three $\{1; 3, 4, 2; 5\}$ (see figure 1). In the first segmenta-

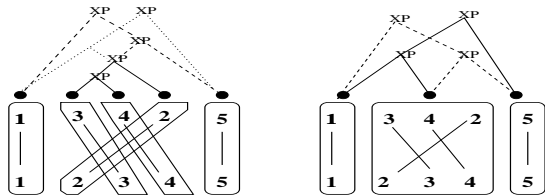


Fig. 1. Two segmentations of an alignment: both are derivable by two binarizations/derivations.

tion, due to crossing alignments, each of the containers $\{3\}$,

INPUT: Word-aligned parallel training data T

OUTPUT: Estimates π for all $P(f | e)$

Split data T into equal parts H_1, \dots, 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 $\hat{\pi}_i^0$ 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 $\hat{\pi}_i^{j-1}$.

M-step: calculate probabilities for pairs in π_i^j from the expected counts

For $1 \leq i \leq 10$ **do** $\hat{\pi}_i^j := \frac{1}{10} \sum_{i=1}^{10} \pi_i^j$

Until $\pi := \{\hat{\pi}_1^j, \dots, \hat{\pi}_{10}^j\}$ has converged

Fig. 2. Penalized Deleted Estimation

$\{4\}$ and $\{2\}$ will not combine with the surrounding context ($\{1\}$ and $\{5\}$) without the other two. Thus, there is only a single binarization of $\{3, 4, 2\}$. Hence, the shorter containers are not more productive than the single long container!

We define our prior based on the ITG spurious derivations. We observe that the *number of possible binarizations* that a segmentation has under the Wu97 bSCFG is a direct function of the ways in which the containers combine among themselves (monotone vs. inverted/crossing) within segmentations. This number provides a more accurate measure of productivity than container length. Hence, we define $P(\sigma_1^I) := \frac{N(\sigma_1^I)}{Z(\Sigma(\mathbf{a}))}$, where $N(\sigma_1^I)$ is the number of binary derivations/trees that σ_1^I has in the binary SCFG (bSCFG), and $Z(\Sigma(\mathbf{a})) = \sum_{\sigma_1^I \in \Sigma(\mathbf{a})} N(\sigma_1^I)$. 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. ESTIMATION BY SMOOTHING

For a latent variable model, Expectation-Maximization (EM) [13] is usually used for finding a (local) maximum-likelihood estimate (MLE). However, under models like ours, where a *phrase pair and its sub phrase pairs* are included in the model, the MLE can be expected to overfit the data. Instead of mere EM we opt for a *smoothed* version: we combine Deleted Estimation [14] with the Jackknife.

Figure 2 shows the pseudo-code for our estimator. Like in Deleted Estimation, we split the training data into ten different splits of *extraction/holdout sets* of respectively 90%/10% of the training set. For every split $1 \leq i \leq 10$, we extract the set of all phrase pairs π_i from the *extraction* set E_i and train

it (under our model) on the *heldout set* H_i . The set of phrase pairs $\pi = \cup_{i=1}^{10} \pi_i$ extracted from the total training data is the set of model parameters. Each set π_i is trained on its corresponding heldout set H_i by EM. The resulting ten separate EM processes are synchronized in their initialization, their iterations as well as stop condition. The EM processes start out from uniform conditional estimates in all π_i . After every EM iteration j , when the M-steps 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 $\hat{\pi}_i^j$. The resulting averaged probabilities in $\hat{\pi}_i^j$ are the current phrase pair estimates, which feed into the next iteration $j+1$ of the different EM processes.

There are two special boundary cases which demand special attention during estimation: (1) **Sparse distributions:** A phrase e that does occur both in H_i and E_i could have a pair $\langle f, e \rangle$ that occurs in H_i but *not* in E_i (i.e., not in π_i). We add the missing pair $\langle f, e \rangle$ to π_i and set its probability to $10^{-5 \cdot len}$, where len is the length of the phrase pair. And (2) **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. We set this to a uniform distribution every time again. We use a bilingual CYK parser to parse the bSCFG. For implementing EM, we employ the Inside-Outside algorithm [15]. 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 .

Note that standard 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: generally speaking, the averages over different heldout sets 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 set². Theoretically speaking, when the training data is unboundedly large, our estimator will converge to the same estimates as Deleted Estimation. When the data is still sparse, our estimator is biased, unlike the MLE which overfit the data. In all experiments, our method (dubbed Penalized Deleted Estimation) outperforms Deleted Estimation.

4. EMPIRICAL EXPERIMENTS

We employ an existing decoder, Moses, which defines a log-linear model, $\mathbf{e}^* = \arg \max_{\mathbf{e}} \sum_{f \in \Phi} \lambda_f H_f(\mathbf{f}, \mathbf{e})$, inter-

²Define $count_y(x)$ 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).

Phrases	System	BLEU
≤ 7 (std setting)	Baseline PBSMT	0.2818
≤ 10	Baseline PBSMT	0.2834
All	Baseline PBSMT	0.2827
≤ 10	Penalized EM + ITG Prior	0.2846
All	Penalized EM + ITG Prior	0.2830

Table 1. Results: Europarl German-English train/dev/test splits from ACL07 2nd Workshop on SMT

polating feature functions H_f (defined next), where λ_f are the interpolation weights. The set Φ consists of the following feature functions (see Moses): 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 $P(f | e)$ and $P(e | f)$. Keeping the other five feature functions fixed, we compare our estimates of $P(f | e)$ and $P(e | f)$ (and the phrase penalty) to the commonly used heuristic estimates.

The training, development and test data all come from the German-English translation shared task of the ACL 2007 Second Workshop on Statistical Machine Translation³. After pruning sentence pairs with word length more than 40 on either side, we are left with 996K sentence pairs as **training set**. The **development and test data** are composed of 2K sentence pairs each. All data sets are lowercased. 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 [2]. 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 these systems.

We compare different versions of our system against the baseline system using the heuristic estimator. Performance is measured by computing the BLEU scores [16] of the system’s translations, when compared against a single reference translation per sentence.

Table 1 exhibits the BLEU scores for the systems. The table shows two sets of results: systems decoding with phrase pairs up to maximum length seven (Moses settings) or ten on both sides and systems that decode with all phrase pairs. For all version of our own system, we train the table containing all phrase pairs (over 95 million phrase pairs) using our estimator, and in the cases where we use phrase length cut-off (seven/ten) during decoding we simply discard phrase pairs

³<http://www.statmt.org/wmt07>

longer than the length cut-off (without re-normalization).

These results show that phrase pairs of maximum length 10 give the best results, followed by *all* phrase pairs, and only after that comes the standard setting with length cut-off 7. This holds both for heuristic estimates and our own estimates. Our estimator, penalized-EM with ITG prior, yields improved BLEU scores over the heuristic in both cases (cutoff=10 and all). While the improvement is modest it should be taken in light of all earlier, less successful attempts at matching the heuristic performance. Given our earlier results on French-English [8], we conclude that more principled estimation of all phrase pairs holds the promise of producing improved system performance. However, we must also note that the room for improvement over the heuristic is small especially when using a decoder that does not allow marginalization (e.g., by sampling) over the different segmentations.

5. DISCUSSION AND FUTURE RESEARCH

In this work we aim at more principled phrase translation probability estimates. We show that estimating *all phrase pairs* with our estimator can be beneficial. The generative model we use assumes latent segmentations and employs ITG-based priors over segmentations. The goal of estimation is a smooth Maximum-Likelihood estimate; we achieve this by embedding EM in a penalized deleted interpolation estimator. The fact that our estimator improves over the heuristic estimator on a reasonably sized data set is rather encouraging.

Our model is far simpler than the model of [3], and can be related to the joint phrase model [5]: it does not need to generate alignments from segmentations because the bilingual containers/concepts preserve the alignments between phrases. Furthermore, we do not use heuristics for pruning the space of segmentations or possible analyses during estimation.

In future research we intend to explore our estimator on other models, such as as the joint phrase model [5]. Based on recent findings, we will attempt marginalizing out the different segmentations during decoding. For this we should build our own decoder in order to experiment with tractable ways for achieving a marginalization effect. Finally, if we view our conditional model as an alternative to the Marcu and Wong joint model, we might be able to explore new ways for inducing phrase alignments from parallel corpora without assuming that word-alignment is given.

6. REFERENCES

- [1] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *25th Annual German Conference on AI (KI 2002)*, M. Jarke et al., Ed. 2002, vol. 2479 of *LNCS*, pp. 18–32, Springer.
- [2] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *HLT-NAACL*, 2003.
- [3] J. DeNero, D. Gillick, J. Zhang, and D. Klein, "Why generative phrase models underperform surface heuristics," in *Proceedings of the workshop on SMT*, 2006, pp. 31–38.
- [4] R. Moore and Ch. Quirk, "An iteratively-trained segmentation-free phrase translation model for statistical machine translation," in *Proceedings of Workshop on SMT*, 2007, pp. 112–119.
- [5] D. Marcu and W. Wong, "A phrase-based, joint probability model for statistical machine translation," in *Proceedings of EMNLP'02*, 2002, pp. 133–139.
- [6] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proceedings of ACL 2005*, 2005, pp. 263–270.
- [7] C. Cherry and D. Lin, "Inversion transduction grammar for joint phrasal translation modeling," in *Proceedings Workshop on Syntax and Structure in Statistical Translation, NAACL-HLT 2006*, 2006.
- [8] M. Mylonakis and K. Sima'an, "Phrase translation probabilities with itg priors and smoothing as learning objective," in *Proceedings of EMNLP-08*, 2008.
- [9] P. Koehn et al, "Moses: Open source toolkit for statistical machine translation," in *ACL demo session*, 2007.
- [10] F. J. Och and H. Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.
- [11] H. Zhang, L. Huang, D. Gildea, and K. Knight, "Synchronous binarization for machine translation," in *HLT-NAACL*, 2006.
- [12] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora.," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, 1997.
- [13] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [14] F. Jelinek and R. L. Mercer, "Interpolated estimation of markov source parameters from sparse data," in *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.
- [15] J.T. Goodman, *Parsing Inside-Out*, PhD thesis, Department of Computer Science, Harvard University, Cambridge, Massachusetts, 1998.
- [16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation.," in *ACL*, 2002, pp. 311–318.