Minimum Error Training of Log-Linear Translation Models

Mauro Cettolo and Marcello Federico

ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
I-38100 Povo - Trento, Italy
{cettolo,federico}@itc.it

Abstract

Recent work on training of log-linear interpolation models for statistical machine translation reported performance improvements by optimizing parameters with respect to translation quality, rather than to likelihood oriented criteria. This work presents an alternative and more direct training procedure for log-linear interpolation models. In addition, we point out the subtle interaction between log-linear models and the beam search algorithm. Experimental results are reported on two Chinese-English evaluation sets, C-Star 2003 and Nist 2003, by using a statistical phrase-based model derived from Model 4. By optimizing parameters with respect to the BLUE score, performance relative improvements by 9.6% and 2.8% were achieved, respectively.

1. Introduction

Log-linear interpolation models, which can be formally derived within the maximum entropy framework [1], have been only recently applied to statistical machine translation (SMT) [2]. In addition, and similarly to what proposed for speech recognition [3], optimization of interpolation parameters can directly address translation quality, rather than the usual maximum likelihood criterion [4].

This paper goes along the direction of [4], and proposes an alternative and more direct training procedure, but computationally more intensive. Moreover, a subtle relationship between the parameter optimization and the beam search algorithm is pointed out, which might have an important impact on the choice of optimal parameters.

2. Log-Linear Model for SMT

Given a source string $\mathbf{f}$ and a target string $\mathbf{e}$, the framework of maximum entropy [5] provides a mean to directly address the posterior probability $\Pr(\mathbf{e} \mid \mathbf{f})$. By introducing the hidden *alignment* variable $\mathbf{a}$, the usual SMT optimization criterion is expressed by:

$$\begin{aligned} \mathbf{e}^* &= \arg\max_{\mathbf{e}} \sum_{\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \\ &\approx \arg\max_{\mathbf{e},\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \end{aligned} \quad (1)$$

The conditional distribution $\Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f})$ is determined through suitable real valued features functions $h_i(\mathbf{e}, \mathbf{f}, \mathbf{a}), i = 1 \ldots M$, and takes the parametric form:

$$p_{\boldsymbol{\lambda}}(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) = \frac{\exp\{\sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f}, \mathbf{a})\}}{\sum_{\mathbf{e},a} \exp\{\sum_i \lambda_i h_i(\mathbf{e}, \mathbf{f}, \mathbf{a})\}} \quad (2)$$

The maximum entropy criterion suggests to compute values λ_i, which maximize the log-likelihood over a training sample T:

$$\boldsymbol{\lambda}_* = \arg\max_{\boldsymbol{\lambda}} \sum_{(\mathbf{e},\mathbf{f},\mathbf{a}) \in T} \log p_{\boldsymbol{\lambda}}(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \quad (3)$$

An interesting log-linear model results if the following feature functions derived from Model 4 [6] are used:

$$\begin{aligned} h_1(\mathbf{e}, \mathbf{f}, \mathbf{a}) &= \log \Pr(\mathbf{e}) \\ h_2(\mathbf{e}, \mathbf{f}, \mathbf{a}) &= \log \Pr(\boldsymbol{\phi} \mid \mathbf{e}) \\ h_3(\mathbf{e}, \mathbf{f}, \mathbf{a}) &= \log \Pr(\boldsymbol{\tau} \mid \mathbf{e}, \boldsymbol{\phi}) \\ h_4(\mathbf{e}, \mathbf{f}, \mathbf{a}) &= \log \Pr(\boldsymbol{\pi} \mid \mathbf{e}, \boldsymbol{\phi}, \boldsymbol{\tau}), \end{aligned}$$

which explain $\mathbf{f}$ and $\mathbf{a}$ for $\mathbf{e}$ in terms of fertilities $\boldsymbol{\phi}$, tablets $\boldsymbol{\tau}$ and permutations $\boldsymbol{\pi}$. In fact, after simple manipulations, the usual decoding criterion for Model 4 results, with the addition of four scaling factors:

$$\mathbf{e}^* \approx \arg\max_{\mathbf{e},\mathbf{a}} Q(\mathbf{e}, \mathbf{a}; \boldsymbol{\lambda}) \quad (4)$$

$$= \arg\max_{\mathbf{e},\mathbf{a}} \Pr(\mathbf{e})^{\lambda_1} \cdot \Pr(\boldsymbol{\phi} \mid \mathbf{e})^{\lambda_2} \cdot \Pr(\boldsymbol{\tau} \mid \mathbf{e}, \boldsymbol{\phi})^{\lambda_3} \cdot \Pr(\boldsymbol{\pi} \mid \mathbf{e}, \boldsymbol{\phi}, \boldsymbol{\tau})^{\lambda_4} \quad (5)$$

To tackle the optimization problem of eq. (5), a search algorithm can be devised which incrementally extends partial translation hypotheses $(\tilde{\mathbf{e}}, \tilde{\mathbf{a}})$ of the source string, until an optimal complete translation is found. A translation is said partial if its corresponding alignment $\tilde{\mathbf{a}}$ does not cover all positions in $\mathbf{f}$. The complexity of the search algorithm mainly depends on the number of possible translations and of target positions to be considered for each source word. To avoid exponential complexity, constraints on both factors are generally introduced. Moreover, the so-called pruning of hypotheses

is deployed, too. Hence, at each step (or target string length), only a *beam* with the most "promising" hypotheses is considered for extension. The following are two very popular pruning methods, which are usually applied to partial translations of the same length and/or covering the same source positions:

- *threshold pruning*: partial hypotheses $(\tilde{\mathbf{e}}, \tilde{\mathbf{a}})$ whose score $Q(\cdot)$ is smaller than the (local) optimum score Q^* times a given factor T, i.e.

$$\frac{Q(\tilde{\mathbf{e}}, \tilde{\mathbf{a}}; \boldsymbol{\lambda})}{Q^*} < T \ , \tag{6}$$

  are eliminated;

- *histogram pruning*: hypotheses not among the top N best scoring ones are pruned.

3. Minimum Error Training

In place of the criterion (3), [4] recently proposed to estimate parameters by minimizing the number of translation errors. We assume that a function $E_D(\boldsymbol{\lambda})$ is available, which measures the translation errors made by running a model defined by parameter values $\boldsymbol{\lambda}$ on a development set D. Hence, parameters are searched by:

$$\boldsymbol{\lambda}_* = \arg\min_{\boldsymbol{\lambda}} E_D(\boldsymbol{\lambda}) \tag{7}$$

Unlike the log-likelihood criterion (3), the objective function $E_D(\cdot)$ might have many local minima. Hence, finding an optimal solution can be very hard. In this work, we use the *simplex method* [7], an algorithm for multivariate function minimization which requires relatively few function evaluations. The same algorithm was already applied for the same task in [8] and for training log-linear language models in [1].

3.1. Interaction with Beam Search

The optimization process, besides tuning the parameters of the statistical model, may also interfere with the beam search. The reason is in the following property of the scoring function (4):

$$Q(\tilde{\mathbf{e}}, \tilde{\mathbf{a}}; \alpha\boldsymbol{\lambda}) = Q(\tilde{\mathbf{e}}, \tilde{\mathbf{a}}; \boldsymbol{\lambda})^{\alpha} \tag{8}$$

for any positive real number α. As a consequence, the threshold criterion (6) is affected by any change of the parameter vector $\boldsymbol{\lambda}$ which corresponds to a scaling transformation. For instance, a contraction of the parameter vector by a factor $\alpha = 0.5$ would implicitly determine the search to prune hypotheses according to the more relaxed constraint:

$$\frac{Q(\tilde{\mathbf{e}}, \tilde{\mathbf{a}}; \boldsymbol{\lambda})}{Q^*} < T^2 \tag{9}$$

Hence, we can expect that any optimization algorithm would be easily attracted by parameter values which relax the pruning threshold, and reduce the error rate at the expense of more computations.

3.2. Simplex Initialization

To remove the impact of the pruning threshold, the simplex method is started from a parameter configuration inducing a loose threshold, so that any further widening of it does not give tangible effect on performance. A potential problem of this approach could be its high computational cost. In our implementation, the optimization remains feasible because the cost of search is also bounded by the histogram pruning. Moreover, optimization of parameters is no more influenced by the beam search, given that there is no relationship between the histogram pruning and the parameters.

Another possibility could be to normalize the parameter vector (like in [2]), or to fix one important parameter and let vary only the others.

4. Experiments

4.1. Baseline System

The core of the translation system is a statistical model, based on the IBM Model 4 and extended to deal with *phrases* rather than with single words [9]. The corresponding log-linear model is similar to that shown in eq. (5) with the addition of two terms, which explicitly scale the fertility and distortion probabilities of the null word. Search is performed by a decoder based on dynamic programming. Both in training and testing, sentences are pre-processed in order to reduce data sparseness. Pre-processing includes: Chinese word segmentation, separation of words from punctuation, handling of acronyms and abbreviations, number extraction, case normalization, etc.

In the following, we will refer to the *baseline* system when uniform parameters are assumed, which can be possibly scaled up or down to modify the beam-search pruning.

4.2. Data

We evaluated our approach on two Chinese-English translation tasks: the Nist 2003 MT evaluation task[1], large-data case-insensitive conditions, and the C-Star 2003 evaluation campaign [2]. The first task concerns with translation of new agencies, while the second task concerns with basic traveling expressions [10]. Test sentences are provided with 4 and 16 human translations, respectively. Tables 1 and 2 report detailed statistics about the used training and test data. For parameter optimization, the Nist 2002 MT evaluation data and 1,000 sentences extracted from the C-Star training data were used, respectively.

It is worth noticing that the C-Star 2003 test set has been used as development set for the IWSLT-2004 evalu-

[1] www.nist.gov/speech/tests/mt

[2] www.c-star.org

Table 1: Statistics of training data.

		Nist	C-Star
vocabulary	source	148K	12K
	target	110K	11K
#words	source	13.1M	434K
	target	13.5M	45K
#sentence pairs		687K	48K
#phrase pairs		4.4M	381K

Table 2: Statistics of test data.

	#sentences	#words	OOV rate
Nist 2003	919	27,254	1.89%
C-Star 2003	506	3,770	1.62%

ation campaign (see [9]).

4.3. Performance Evaluation

Minimum error training was based on two well established MT objective quality measures, that adequately correlate with human subjective evaluations: namely the BLEU [11] and NIST[3] scores. Moreover, in order to compute time requirements independently from the hardware platform and CPU load, the number of hypotheses generated by the search algorithm was assumed as a reliable measure.

5. Results

Best achieved performance for each task are reported in Table 3 and Table 4. In both cases consistent performance improvements were achieved with respect to baseline systems with comparable search complexity. On the Nist 2003 evaluation set, the BLEU score improved by 2.8% relative, while the NIST score by 1.7% relative. More significant gains were achieved on the C-Star task: 9.6% relative improvement on the BLEU score and 1.8% on the NIST score.

Interestingly, for the C-Star task, a slightly better NIST score was obtained by optimizing parameters with respect to the BLEU score (see Table 4). In general, optimizing the NIST score results much harder than optimizing the BLEU score.

A comparison considering many beam-search working points is shown, for the C-Star 2003 task, in Figure 1. The two plotted curves represent performance of the baseline (using uniform parameters) and the log-linear model with parameters optimizing the BLEU score. In both cases, different working points of the search algorithm were obtained by linearly scaling the parameter

[3] www.nist.gov/speech/tests/mt

Table 3: Best translation results on the Nist 2003 task with different parameter settings: optimizing BLEU, optimizing NIST, and uniform.

Criterion	BLEU	NIST	# hyp
BLEU	0.1854	7.2882	116M
NIST	0.1840	7.3362	115M
Baseline	0.1803	7.2115	116M

Table 4: Best translation results on the C-Star 2003 task with different parameter settings: optimizing BLEU, optimizing NIST, and uniform.

Criterion	BLEU	NIST	# hyp
BLEU	0.4614	8.4945	14.9M
NIST	0.4581	8.4675	14.9M
Baseline	0.4208	8.3169	14.8M

vector. Figure 2 is the analogous of Figure 1, but refers to parameters optimal with respect to the NIST score.

At equal computational costs, we observe performance gains between 7-10% on the BLEU score, and 1-2% on the NIST score, when optimized parameters are used instead of uniform ones. These achievements are significant, since parameter optimization on the test set allows for an improvement of 12.8% and 3.7%, respectively, which are fair approximations of the best achievable scores on that test set.

In addition, we notice that best BLEU performance obtained by the baseline (best score of the baseline in Figure 1) is reached by the minimum error trained system with almost 50% less search cost.

6. Previous Work and Discussion

The estimation of the translation probability $\Pr(\mathbf{e} \mid \mathbf{f})$ in the framework of maximum entropy was suggested by [2]. In [4], parameter estimation of a log-linear translation model was done by optimizing the error rate instead of the likelihood. Error minimization relied on the availability of a set of n-best candidate translations for each input sentence, produced by the search algorithm. During training, optimal parameters were searched through the Powell's algorithm [7]. Since the n-best list can significantly change by modifying the parameters, the procedure is iterated until the n-best list remains stable. The author claims that, in practice, 5-7 iteration are enough for convergence.

Unlike that in [4], our optimization method exploits all possible solutions of the search algorithm and does not limit the search to n-best lists. As a consequence, our procedure is computationally more expensive. Each

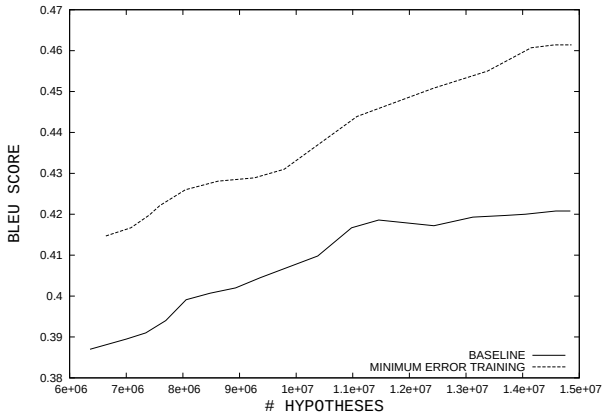

Figure 1: BLEU score as function of the number of generated hypotheses.

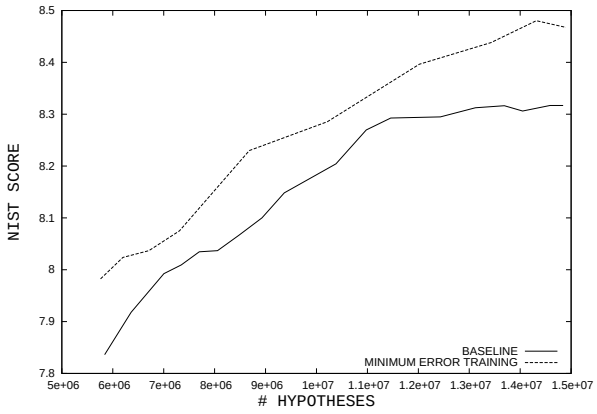

Figure 2: NIST score as function of the number of generated hypotheses.

iteration of the simplex algorithm requires translating the full development set. Practically, each iteration on the Nist 2003 development data takes about 7 minutes with 12 CPUs, while a solution is found in about 100 steps.

Although we measured consistent and stable improvements with two different MT scores and along many different beam-search configurations, our performance gains are significantly lower than those shown in [4]. Nevertheless, as pointed out in [8], the simplex "method has the advantage that it is not limited to the model scaling factors as the method described in" [4], but it also allows to optimize parameters of the models to be interpolated. Then, a direct comparison of the two approaches would be desirable: with this goal in mind, we are currently developing optimization methods exploiting n-best lists.

7. Acknowledgments

This work was partially financed by the European Commission under the project PF-STAR - Preparing Future Speech Translation Research (IST-2001-37599, http://pfstar.itc.it), and by the Province of Trento (FDR-PAT program) under the project WebFAQ.

8. References

[1] D. Klakow, "Log-linear interpolation of language models," in *Proc. of ICSLP*, Sidney, Australia, 1998, pp. 1695–1698.

[2] F.J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. of the ACL*, Philadelphia, PA, 2002, pp. 295–302.

[3] P. Beyerlein, "Discriminative model combination," in *Proc. of IEEE ASRU*, S. Barbara, CA, 1997, pp. 238–245.

[4] F.J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the ACL*, Sapporo, Japan, 2003, pp. 160–167.

[5] A.L. Berger, S.A. Della Pietra, and V.J. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[6] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and Robert L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–312, 1993.

[7] W.M. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical recipes in C: the art of scientific computing*, Cambridge Univ. Press, 1992.

[8] R. Zens, and H. Ney, "Improvements in phrase-based statistical machine translation", in *Proc. of HLT/NAACL*, Boston, MA, 2004, pp. 257–264.

[9] N. Bertoldi, R. Cattoni, M. Cettolo, and M. Federico, "The ITC-irst statistical machine translation system for IWSLT-2004", in these proceedings.

[10] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world," in *Proc. of LREC*, Las Palmas, Spain, 2002, pp. 147–152.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," Research Report RC22176, IBM Research Division, Thomas J. Watson Research Center, 2001.