Monte Carlo inference and maximization for phrase-based translation

Abhishek Arun[*] a.arun@sms.ed.ac.uk
Chris Dyer[†] redpony@umd.edu
Barry Haddow[*] bhaddow@inf.ed.ac.uk
Phil Blunsom[*] pblunsom@inf.ed.ac.uk
Adam Lopez[*] alopez@inf.ed.ac.uk
Philipp Koehn[*] pkoehn@inf.ed.ac.uk

[*]Department of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK

[†]Department of Linguistics
University of Maryland
College Park, MD 20742, USA

Abstract

Recent advances in statistical machine translation have used beam search for approximate NP-complete inference within probabilistic translation models. We present an alternative approach of sampling from the posterior distribution defined by a translation model. We define a novel Gibbs sampler for sampling translations given a source sentence and show that it effectively explores this posterior distribution. In doing so we overcome the limitations of heuristic beam search and obtain theoretically sound solutions to inference problems such as finding the maximum probability translation and minimum expected risk training and decoding.

1 Introduction

Statistical machine translation (SMT) poses the problem: given a foreign sentence f, find the translation e^* that maximises the conditional posterior probability $p(e|f)$. This probabilistic formulation of translation has driven development of state-of-the-art systems which are able to learn from parallel corpora which were generated for other purposes — a direct result of employing a mathematical framework that we can reason about independently of any particular model.

For example, we can train SMT models using maximum likelihood estimation (Brown et al., 1993; Och and Ney, 2000; Marcu and Wong, 2002). Alternatively, we can train to minimise probabilistic conceptions of *risk* (expected loss) with respect to translation metrics, thereby obtaining better results for those metrics (Kumar and Byrne, 2004; Smith and Eisner, 2006; Zens and Ney, 2007). We can also use Bayesian inference techniques to avoid resorting to heuristics that damage the probabilistic interpretation of the models (Zhang et al., 2008; DeNero et al., 2008; Blunsom et al., 2009).

Most models define multiple derivations for each translation; the probability of a translation is thus the sum over all of its derivations. Unfortunately, finding the maximum probability translation is NP-hard for all but the most trivial of models in this setting (Sima'an, 1996). It is thus necessary to resort to approximations for this sum and the search for its maximum e^*.

The most common of these approximations is the max-derivation approximation, which for many models can be computed in polynomial time via dynamic programming (DP). Though effective for some problems, it has many serious drawbacks for probabilistic inference:

1. It typically differs from the true model maximum.

2. It often requires additional approximations in search, leading to further error.

3. It introduces restrictions on models, such as use of only local features.

4. It provides no good solution to compute the normalization factor $Z(f)$ required by many probabilistic algorithms.

In this work, we solve these problems using a Monte Carlo technique with none of the above drawbacks. Our technique is based on a novel Gibbs sampler that draws samples from the posterior distribution of a phrase-based translation model (Koehn et al., 2003) but operates in linear time with respect to the number of input words (Section 2). We show

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), pages 102–110,
Boulder, Colorado, June 2009.

that it is effective for both decoding (Section 3) and minimum risk training (Section 4).

2 A Gibbs sampler for phrase-based translation models

We begin by assuming a phrase-based translation model in which the input sentence, f, is segmented into phrases, which are sequences of adjacent words.[1] Each foreign phrase is translated into the target language, to produce an output sentence e and an alignment a representing the mapping from source to target phrases. Phrases are allowed to be reordered during translation.

The model is defined with a log-linear form, with feature function vector $\mathbf{h}$ and parametrised by weight vector $\boldsymbol{\theta}$, as described in Koehn et al. (2003).

$$P(e, a|f; \boldsymbol{\theta}) = \frac{\exp\left[\boldsymbol{\theta} \cdot \mathbf{h}(e, a, f)\right]}{\sum_{\langle e', a' \rangle} \exp\left[\boldsymbol{\theta} \cdot \mathbf{h}(e', a', f)\right]} \quad (1)$$

The features $\mathbf{h}$ of the model are usually few and are themselves typically probabilistic models indicating e.g, the relative frequency of a target phrase translation given a source phrase (translation model), the fluency of the target phrase (language model) and how phrases reorder with respect to adjacent phrases (reordering model). There is a further parameter Λ that limits how many source language words may intervene between two adjacent target language phrases. For the experiments in this paper, we use $\Lambda = 6$.

2.1 Gibbs sampling

We use Markov chain Monte Carlo (MCMC) as an alternative to DP search (Geman and Geman, 1984; Metropolis and Ulam, 1949). MCMC probabilistically generates sample derivations from the complete search space. The probability of generating each sample is conditioned on the previous sample, forming a Markov chain. After a long enough interval (referred to as the burn-in) this chain returns samples from the desired distribution.

Our MCMC sampler uses Gibbs sampling, which obtains samples from the joint distribution of a set of random variables $X = \{X_1, \ldots, X_n\}$. It starts with some initial state $(X_1 = x_{10}, \ldots, X_n = x_{n0})$, and generates a Markov chain of samples, where each sample is the result of applying a set of *Gibbs operators* to the previous sample. Each operator is defined by specifying a subset of the random variables $Y \subset X$, which the operator updates by sampling from the conditional distribution $P(Y|X \setminus Y)$. The set $X \setminus Y$ is referred to as the Markov blanket and is unchanged by the operator.

In the case of translation, we require a Gibbs sampler that produces a sequence of samples, $\mathcal{S}_1^N = (e_1, a_1) \ldots (e_N, a_N)$, that are drawn from the distribution $P(e, a|f)$. These samples can thus be used to estimate the expectation of a function $h(e, a, f)$ under the distribution as follows:

$$\mathbb{E}_{P(a,e|f)}[h] = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} h(a_i, e_i, f) \quad (2)$$

Taking h to be an indicator function $h = \delta(a, \hat{a})\delta(e, \hat{e})$ provides an estimate of $P(\hat{a}, \hat{e}|f)$, and using $h = \delta(e, \hat{e})$ marginalises over all derivations a', yielding an estimate of $P(\hat{e}|f)$.

2.2 Gibbs operators

Our sampler consists of three operators. Examples of these are depicted in Figure 1.

The RETRANS operator varies the translation of a single source phrase. Segmentation, alignment, and all other translations are held constant.

The MERGE-SPLIT operator varies the source segmentation at a single word boundary. If the boundary is a split point in the current hypothesis, the adjoining phrases can be merged, provided that the corresponding target phrases are adjacent and the phrase table contains a translation of the merged phrase. If the boundary is not a split point, the covering phrase may be split, provided that the phrase table contains a translation of both new phrases. Remaining segmentation points, phrase alignment and phrase translations are held constant.

The REORDER operator varies the target phrase order for a pair of source phrases, provided that the new alignment does not violate reordering limit Λ. Segmentation, phrase translations, and all other alignments are held constant.

To illustrate the RETRANS operator, we will assume a simplified model with two features: a bigram language model P_{lm} and a translation model P_{tm}. Both features are assigned a weight of 1.

[1] These phrases are not necessarily linguistically motivated.

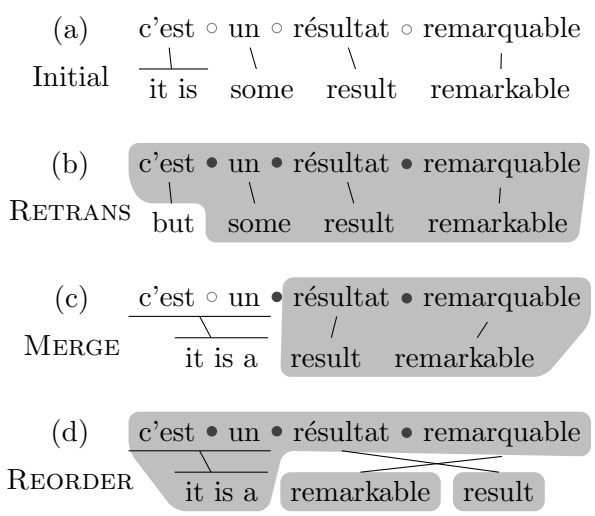

Figure 1: Example evolution of an initial hypothesis via application of several operators, with Markov blanket indicated by shading.

We denote the start of the sentence with S and the language model context with C. Assuming the French phrase *c'est* can be translated either as *it is* or *but*, the RETRANS operator at step (b) stochastically chooses an English phrase, $\hat{e}$ in proportion to the phrases' conditional probabilities.

$$P(\textit{but}|\textit{c'est}, C) = \frac{P_{tm}(\textit{but}|\textit{c'est}) \cdot P_{lm}(\textit{S but some})}{Z}$$

and

$$P(\textit{it is}|\textit{c'est}, C) = \frac{P_{tm}(\textit{it is}|\textit{c'est}) \cdot P_{lm}(\textit{S it is some})}{Z}$$

where

$$\begin{aligned} Z &= P_{tm}(\textit{but}|\textit{c'est}) \cdot P_{lm}(\textit{S but some}) + \\ &\quad P_{tm}(\textit{it is}|\textit{c'est}) \cdot P_{lm}(\textit{S it is some}) \end{aligned}$$

Conditional distributions for the MERGE-SPLIT and REORDER operators can be derived in an analogous fashion.

A complete iteration of the sampler consists of applying each operator at each possible point in the sentence, and a sample is collected after each operator has performed a complete pass.

2.3 Algorithmic complexity

Since both the RETRANS and MERGE-SPLIT operators are applied by iterating over source side word positions, their complexity is linear in the size of the input.

The REORDER operator iterates over the positions in the input and for the source phrase found at that position considers swapping its target phrase with that of every other source phrase, *provided* that the reordering limit is not violated. This means that it can only consider swaps within a fixed-length window, so complexity is linear in sentence length.

2.4 Experimental verification

To verify that our sampler was behaving as expected, we computed the KL divergence between its inferred distribution $\hat{q}(e|f)$ and the true distribution over a single sentence (Figure 2). We computed the true posterior distribution $p(e|f)$ under an Arabic-English phrase-based translation model with parameters trained to maximise expected BLEU (Section 4), summing out the derivations for identical translations and computing the partition term $Z(f)$. As the number of iterations increases, the KL divergence between the distributions approaches zero.

3 Decoding

The task of decoding amounts to finding the single translation e^* that maximises or minimises some criterion given a source sentence f. In this section we consider three common approaches to decoding, maximum translation (MaxTrans), maximum derivation (MaxDeriv), and minimum risk decoding (MinRisk):

$$e^* = \begin{cases} \arg\max_{(e,a)} p(e, a|f) & \text{(MaxDeriv)} \\ \arg\max_e p(e|f) & \text{(MaxTrans)} \\ \arg\min_e \sum_{e'} \ell_{e'}(e) p(e'|f) & \text{(MinRisk)} \end{cases}$$

In the minimum risk decoder, $\ell_{e'}(e)$ is any real-valued loss (error) function that computes the error of one hypothesis e with respect to some reference e'. Our loss is a sentence-level approximation of $(1 - \text{BLEU})$.

As noted in section 2, the Gibbs sampler can be used to provide an estimate of the probability distribution $P(a, e|f)$ and therefore to determine the maximum of this distribution, in other words the most likely derivation. Furthermore, we can marginalise over the alignments to estimate $P(e|f)$

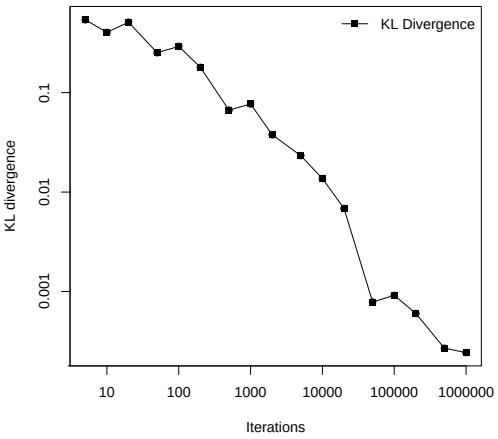

Figure 2: The KL divergence of the true posterior distribution and the distribution estimated by the Gibbs sampler at different numbers of iterations for the Arabic source sentence *r}ys wzrA' mAlyzyA yzwr Alflbyn* (in English, *The prime minister of Malaysia visits the Philippines*).

and so obtain the most likely translation. The Gibbs sampler can therefore be used as a decoder, either running in max-derivation and max-translation mode. Using the Gibbs sampler in this way makes max-translation decoding tractable, and so will help determine whether max-translation offers any benefit over the usual max-derivation. Using the Gibbs sampler as a decoder also allows us to verify that it is producing valid samples from the desired distribution.

3.1 Training data and preparation.

The experiments in this section were performed using the French-English and German-English parallel corpora from the WMT09 shared translation task (Callison-Burch et al., 2009), as well as 300k parallel Arabic-English sentences from the NIST MT evaluation training data.[2] For all language pairs, we constructed a phrase-based translation model as described in Koehn et al. (2003), limiting the phrase length to 5. The target side of the parallel corpus was used to train a 3-gram language model.

[2]The Arabic-English training data consists of the eTIRR corpus (LDC2004E72), the Arabic news corpus (LDC2004T17), the Ummah corpus (LDC2004T18), and the sentences with confidence $c > 0.995$ in the ISI automatically extracted web parallel corpus (LDC2006T02).

For the German and French systems, the DEV2006 set was used for model tuning and the TEST2007 (in-domain) and NEWS-DEV2009B (out-of-domain) sets for testing. For the Arabic system, the MT02 set (10 reference translations) was used for tuning and MT03 (4 reference translations) was used for evaluation. To reduce the size of the phrase table, we used the association-score technique suggested by Johnson et al. (2007a). Translation quality is reported using case-insensitive BLEU (Papineni et al., 2002).

3.2 Translation performance

For the experiments reported in this section, we used feature weights trained with minimum error rate training (MERT; Och, 2003) . Because MERT ignores the denominator in Equation 1, it is invariant with respect to the scale of the weight vector θ — the Moses implementation simply normalises the weight vector it finds by its ℓ_1-norm. However, when we use these weights in a true probabilistic model, the scaling factor affects the behaviour of the model since it determines how peaked or flat the distribution is. If the scaling factor is too small, then the distribution is too flat and the sampler spends too much time exploring unimportant probability regions. If it is too large, then the distribution is too peaked and the sampler may concentrate on a very narrow probability region. We optimised the scaling factor on a 200-sentence portion of the tuning set, finding that a multiplicative factor of 10 worked best for fr-en and a multiplicative factor of 6 for de-en. [3]

The first experiment shows the effect of different initialisations and numbers of sampler iterations on max-derivation decoding performance of the sampler. The Moses decoder (Koehn et al., 2007) was used to generate the starting hypothesis, either in full DP max-derivation mode, or alternatively with restrictions on the features and reordering, or with zero weights to simulate a random initialisation, and the number of iterations varied from 100 to 200,000, with a 100 iteration burn-in in each case. Figure 3 shows the variation of model score with sampler iteration, for the different starting points, and for both language pairs.

[3]We experimented with *annealing*, where the scale factor is gradually increased to sharpen the distribution while sampling. However, we found no improvements with annealing.

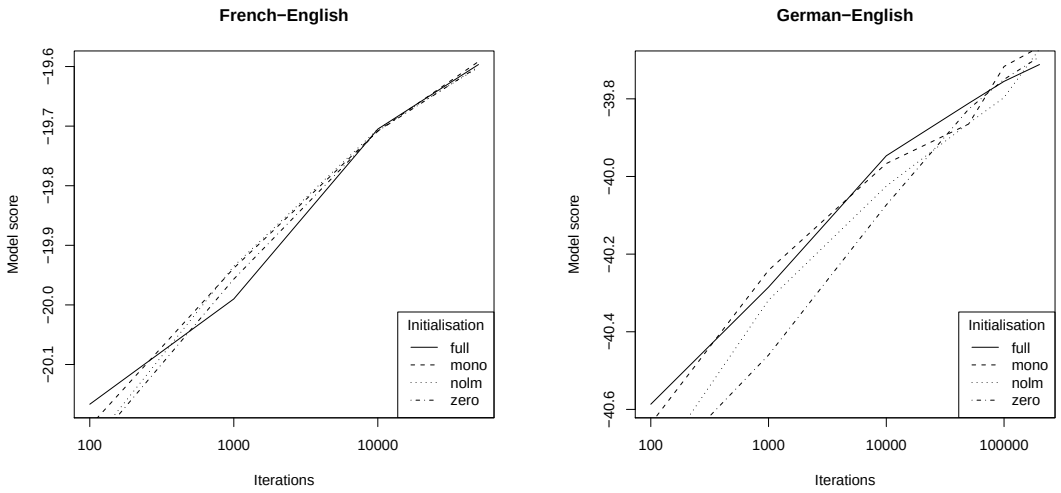

Figure 3: Mean maximum model score, as a function of iteration number and starting point. The starting point can either be the full max-derivation translation (**full**), the monotone translation (**mono**), the monotone translation with no language model (**nolm**) or the monotone translation with all weights set to zero (**zero**).

Comparing the best model scores found by the sampler, with those found by the Moses decoder with its default settings, we found that around 50,000 sampling iterations were required for fr-en and 100,000 for de-en, for the sampler to give equivalent model scores to Moses. From Figure 3 we can see that the starting point did not have an appreciable effect on the model score of the best derivation, except with low numbers of iterations. This indicates that the sampler is able to move fairly quickly towards the maximum of the distribution from any starting point, in other words it has good mobility. Running the sampler for 100,000 iterations took on average 1670 seconds per sentence on the French-English data set and 1552 seconds per sentence on German-English.

A further indication of the dependence of sampler accuracy on the iteration count is provided by Figure 4. In this graph, we show the mean Spearman's rank correlation between the nbest lists of derivations when ranked by (i) model score and (ii) the posterior probability estimated by the sampler. This measure of sampler accuracy also shows a logarithmic dependence on the sample size.

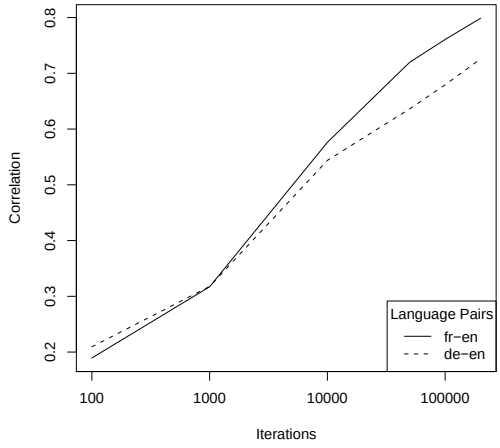

Figure 4: Mean Spearman's rank correlation of 1000-best list of derivations ranked according to (i) model score and (ii) posterior probability estimated by sampler. This was measured on a 200 sentence subset of DEV2006.

3.3 Minimum risk decoding

The sampler also allows us to perform minimum Bayes risk (MBR) decoding, a technique introduced by Kumar and Byrne (2004). In their work, as an approximation of the model probability distribution, the expected loss of the decoder is calculated by summing over an n-best list. With the Gibbs sampler, however, we should be able to obtain a much more accurate view of the model probability distribution. In order to compare max-translation, max-derivation and MBR decoding with the Gibbs sampler, and the Moses baseline, we ran experiments

	fr-en		de-en	
	in	out	in	out
Moses	32.7	19.1	27.4	15.9
MaxD	32.6	19.1	27.0	15.5
MaxT	32.6	19.1	27.4	16.0
MBR	32.6	19.2	27.3	16.0

Table 1: Comparison of the BLEU score of the Moses decoder with the sampler running in max-derivation (MaxD), max-translation (MaxT) and minumum Bayes risk (MBR) modes. The test sets are TEST2007 (in) and NEWS-DEV2009B (out)

on both European language pairs, using both the in-domain and out-of-domain test sets. The sampler was initialised with the output of Moses with the feature weights set to zero and restricted to monotone, and run for 100,000 iterations with a 100 iteration burn-in. The scale factors were set to the same values as in the previous experiment. The relative translation quality (measured according to BLEU) is shown in Table 1.

3.4 Discussion

These results show very little difference between the decoding methods, indicating that the Gibbs sampling decoder can perform as well as a standard DP based max-derivation decoder with these models, and that there is no gain from doing max-translation or MBR decoding. However it should be noted that the model used for these experiments was optimised by MERT, for max-derivation decoding, and so the experiments do not rule out the possibility that max-translation and MBR decoding will offer an advantage on an appropriately optimised model.

4 Minimum risk training

In the previous section, we described how our sampler can be used to search for the best translation under a variety of decoding criteria (max derivation, translation, and minimum risk). However, there appeared to be little benefit to marginalizing over the latent derivations. This is almost certainly a side effect of the MERT training approach that was used to construct the models so as to maximise the performance of the model on its single best derivation, without regard to the shape of the rest of the distribution (Blunsom et al., 2008). In this section we describe a further application of the Gibbs sampler: to do *unbiased* minimum risk training.

While there have been at least two previous attempts to do minimum risk training for MT, both approaches relied on biased k-best approximations (Smith and Eisner, 2006; Zens and Ney, 2007). Since we sample from the whole distribution, we will have a more accurate risk assessment.

The risk, or expected loss, of a probabilistic translation model on a corpus $\mathcal{D}$, defined with respect to a particular loss function $\ell_{\hat{e}}(e)$, where $\hat{e}$ is the reference translation and e is a hypothesis translation

$$\mathcal{L} = \sum_{\langle \hat{e}, f \rangle \in \mathcal{D}} \sum_{e} p(e|f) \ell_{\hat{e}}(e) \tag{3}$$

This value can be trivially computed using equation (2). In this section, we are concerned with finding the parameters $\boldsymbol{\theta}$ that minimise (3). Fortunately, with the log-linear parameterization of $p(e|f)$, $\mathcal{L}$ is differentiable with respect to $\boldsymbol{\theta}$:

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = \sum_{\langle \hat{e}, f \rangle \in \mathcal{D}} \sum_{e} p(e|f) \ell_{\hat{e}}(e) \left(h_k - \mathbb{E}_{p(e|f)}[h_k] \right) \tag{4}$$

Equation (4) is slightly more complicated to compute using the sampler since it requires the feature expectation in order to evaluate the final term. However, this can be done simply by making two passes over the samples, computing the feature expectations on the first pass and the gradient on the second.

We have now shown how to compute our objective (3), the expected loss, and a gradient with respect to the model parameters we want to optimise, (4), so we can use any standard first-order optimization technique. Since the sampler introduces stochasticity into the gradient and objective, we use stochastic gradient descent methods which are more robust to noise than more sophisticated quasi-Newtonian methods like L-BFGS (Schraudolph et al., 2007). For the experiments below, we updated the learning rate after each step proportionally to difference in successive gradients (Schraudolph, 1999).

For the experiments reported in this section, we used sample sizes of 8000 and estimated the gradient on sets of 100 sentences drawn randomly (with replacement) from the development corpus. For a

Training	Decoder	MT03
MERT	Moses Max Derivation	44.6
	Moses MBR	44.8
	Gibbs MBR	44.9
MinRisk	Moses Max Derivation	40.6
	MaxTrans	41.8
	Gibbs MBR	42.9

Table 2: Decoding with minimum risk trained systems, compared with decoding with MERT-trained systems on Arabic to English MT03 data

loss function we use 4-gram $(1 - \text{BLEU})$ computed individually for each sentence[4]. By examining performance on held-out data, we find the model converges typically in fewer than 20 iterations.

4.1 Training experiments

During preliminary experiments with training, we observed on a held-out data set (portions of MT04) that the magnitude of the weights vector increased steadily (effectively sharpening the distribution), but without any obvious change in the objective. Since this resulted in poor generalization we added a regularization term of $||\vec{\theta} - \vec{\mu}||^2/2\sigma^2$ to $\mathcal{L}$. We initially set the means to zero, but after further observing that the translations under all decoding criteria tended to be shorter than the reference (causing a significant drop in performance when evaluated using BLEU), we found that performance could be improved by setting $\mu_{WP} = -0.5$, indicating a preference for a lower weight on this parameter.

Table 2 compares the performance on Arabic to English translation of systems tuned with MERT (maximizing corpus BLEU) with systems tuned to maximise expected sentence-level BLEU. Although the performance of the minimum risk model under all decoding criteria is lower than that of the original MERT model, we note that the positive effect of marginalizing over derivations as well as using minimum risk decoding for obtaining good results on this model. A full exploration of minimum risk training is beyond the scope of this paper, but these initial experiments should help emphasise the versatility of the sampler and its utility in solving a variety of problems. In the conclusion, we will, however, discuss some possible future directions that can be taken to make this style of training more competitive with standard baseline systems.

[4]The ngram precision counts are smoothed by adding 0.01 for n $>$ 1

5 Discussion and future work

We have described an algorithmic technique that solves certain problems, but also verifies the utility of standard approximation techniques. For example, we found that on standard test sets the sampler performs similarly to the DP max-derivation solution and equally well regardless of how it is initialised. From this we conclude that at least for MERT-trained models, the max-derivation approximation is adequate for finding the best translation.

Although the training approach presented in Section 4 has a number of theoretical advantages, its performance in a one-best evaluation falls short when compared with a system tuned for optimal one-best performance using MERT. This contradicts the results of Zens and Ney (2007), who optimise the same objective and report improvements over a MERT baseline. We conjecture that the difference is due to the biased k-best approximation they used. By considering only the most probable derivations, they optimise a smoothed error surface (as one does in minimum risk training), but not one that is indicative of the true risk. If our hypothesis is accurate, then the advantage is accidental and ultimately a liability. Our results are in line with those reported by Smith and Eisner (2006) who find degradation in performance when minimizing risk, but compensate by "sharpening" the model distribution for the final training iterations, effectively maximising one-best performance rather minimising risk over the full distribution defined by their model. In future work, we will explore possibilities for artificially sharpening the distribution during training so as to better anticipate the one-best evaluation conditions typical of MT. However, for applications which truly do require a distribution over translations, such as re-ranking, our method for minimising expected risk would be the objective of choice.

Using sampling for model induction has two further advantages that we intend to explore. First, although MERT performs quite well on models with

small numbers of features (such as those we considered in this paper), in general the algorithm severely limits the number of features that can be used since it does not use gradient-based updates during optimization, instead updating one feature at a time. Our training method (Section 4) does not have this limitation, so it can use many more features.

Finally, for the DP-based max-derivation approximation to be computationally efficient, the features characterizing the steps in the derivation must be either computable independently of each other or with only limited local context (as in the case of the language model or distortion costs). This has led to a situation where entire classes of potentially useful features are not considered because they would be impractical to integrate into a DP based translation system. With the sampler this restriction is mitigated: any function of $h(e, f, a)$ may participate in the translation model subject only to its own computability. Freed from the rusty manacles of dynamic programming, we anticipate development of many useful features.

6 Related work

Our sampler is similar to the decoder of Germann et al. (2001), which starts with an approximate solution and then incrementally improves it via operators such as RETRANS and MERGE-SPLIT. It is also similar to the estimator of Marcu and Wong (2002), who employ the same operators to search the alignment space from a heuristic initialisation. Although the operators are similar, the use is different. These previous efforts employed their operators in a greedy hill-climbing search. In contrast, our operators are applied probabilistically, making them theoretically well-founded for a variety of inference problems.

Our use of Gibbs sampling follows from its increasing use in Bayesian inference problems in NLP (Finkel et al., 2006; Johnson et al., 2007b). Most closely related is the work of DeNero et al. (2008), who derive a Gibbs sampler for phrase-based alignment, using it to infer phrase translation probabilities. The use of Monte Carlo techniques to calculate posteriors is similar to that of Chappelier and Rajman (2000) who use those techniques to find the best parse under models where the derivation and the parse are not isomorphic.

To our knowledge, we are the first to apply Monte Carlo methods to maximum translation and minimum risk translation. Approaches to the former (Blunsom et al., 2008; May and Knight, 2006) rely on dynamic programming techniques which do not scale well without heuristic approximations, while approaches to the latter (Smith and Eisner, 2006; Zens et al., 2007) use biased k-best approximations.

7 Conclusion

We have described a Gibbs sampler for approximating two intractable problems in SMT: maximum translation decoding (and its variant, minimum risk decoding) and minimum risk training. By using Monte Carlo techniques we avoid the biases associated with the more commonly used DP based max-derivation (or k-best derivation) approximation. In doing so we provide a further tool to the translation community that we envision will allow the development and analysis of increasing theoretically well motivated techniques.

Acknowledgments

This research was supported in part by the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-2-001; and by the EuroMatrix project funded by the European Commission (6th Framework Programme). The project made use of the resources provided by the Edinburgh Compute and Data Facility (`http://www.ecdf.ed.ac.uk/`). The ECDF is partially supported by the eDIKT initiative (`http://www.edikt.org.uk/`).

References

P. Blunsom, T. Cohn, and M. Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proc. of ACL-HLT*.

P. Blunsom, T. Cohn, and M. Osborne. 2009. Bayesian synchronous grammar induction. In *Advances in Neural Information Processing Systems 21*, pages 161–168.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder, editors. 2009. *Proc. of Workshop on Machine Translations*, Athens.

J.-C. Chappelier and M. Rajman. 2000. Monte-Carlo sampling for NP-hard maximization problems in the

framework of weighted parsing. In *Natural Language Processing – NLP 2000, number 1835 in Lecture Notes in Artificial Intelligence*, pages 106–117. Springer.

J. DeNero, A. Bouchard, and D. Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proc. of EMNLP*.

J. R. Finkel, C. D. Manning, and A. Y. Ng. 2006. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Proc. of EMNLP*.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of ACL*. Association for Computational Linguistics, July.

J. Johnson, J. Martin, G. Foster, and R. Kuhn. 2007a. Improving translation quality by discarding most of the phrasetable. In *Proc. of EMNLP-CoNLL*, Prague.

M. Johnson, T. Griffiths, and S. Goldwater. 2007b. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of NAACL-HLT*, pages 139–146, Rochester, New York, April.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL*, pages 48–54, Morristown, NJ, USA.

P. Koehn, H. Hoang, A. B. Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL Demonstration Session*, pages 177–180, June.

S. Kumar and W. Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Processings of HLT-NAACL*.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP*, pages 133–139.

J. May and K. Knight. 2006. A better n-best list: Practical determinization of weighted finite tree automata. In *Proc. of NAACL-HLT*.

N. Metropolis and S. Ulam. 1949. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

F. Och and H. Ney. 2000. A comparison of alignment models for statistical machine translation. In *Proc. of COLING*, Saarbrucken, Germany, July.

F. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167, Sapporo, Japan, July.

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

N. N. Schraudolph, J. Yu, and S. Günter. 2007. A stochastic quasi-Newton method for online convex optimization. In *Proc. of Artificial Intelligence and Statistics*.

N. N. Schraudolph. 1999. Local gain adaptation in stochastic gradient descent. Technical Report IDSIA-09-99, IDSIA.

K. Sima'an. 1996. Computational complexity of probabilistic disambiguation by means of tree grammars. In *Proc. of COLING*, Copenhagen.

D. A. Smith and J. Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proc. of COLING-ACL*, pages 787–794.

R. Zens and H. Ney. 2007. Efficient phrase-table representation for machine translation with applications to online MT and speech translation. In *Proc. of NAACL-HLT*, Rochester, New York.

R. Zens, S. Hasan, and H. Ney. 2007. A systematic comparison of training criteria for statistical machine translation. In *Proc. of EMNLP*, pages 524–532, Prague, Czech Republic.

H. Zhang, C. Quirk, R. C. Moore, and D. Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proc. of ACL: HLT*, pages 97–105, Columbus, Ohio.