Mach Translat (2010) 24:103–121
DOI 10.1007/s10590-010-9080-7

Monte Carlo techniques for phrase-based translation

Abhishek Arun · Barry Haddow · Philipp Koehn · Adam Lopez · Chris Dyer · Phil Blunsom

Received: 2 November 2009 / Accepted: 25 May 2010 / Published online: 24 June 2010
© Springer Science+Business Media B.V. 2010

Abstract Recent advances in statistical machine translation have used approximate beam search for 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 risk training and decoding.

This paper extends work presented in Arun et al. (2009).

A. Arun (✉) · B. Haddow · P. Koehn · A. Lopez
University of Edinburgh, Edinburgh, UK
e-mail: a.arun@sms.ed.ac.uk

B. Haddow
e-mail: bhaddow@inf.ed.ac.uk

P. Koehn
e-mail: pkoehn@inf.ed.ac.uk

A. Lopez
e-mail: alopez@inf.ed.ac.uk

C. Dyer
University of Maryland, College Park, MD, USA
e-mail: redpony@umd.edu

P. Blunsom
Oxford University Computing Laboratory, Oxford, UK
e-mail: pblunsom@comlab.ox.ac.uk

Keywords Statistical machine translation · Gibbs sampling · Machine learning · MCMC

1 Introduction

Statistical machine translation (SMT) poses the problem: given a foreign sentence f, find the translation e^* that maximises the posterior probability $p(e|f)$. Translation models, such as the phrase-based translation model that we focus on in this paper (Koehn et al. 2003), define multiple derivations for each translation, making the probability of a translation the sum over all of its derivations. Unfortunately, finding the maximum probability translation is NP-hard for this model (Casacuberta and Higuera 2000), making approximations necessary. The most common of these approximations is the Viterbi approximation, which can be computed in polynomial time via dynamic programming (DP). While fast and effective for many problems, it has two serious drawbacks for probabilistic inference. First, the error incurred by the Viterbi maximum with respect to the true model maximum is unbounded. Second, the DP solution requires substantial pruning and restricts the use of non-local features. The latter problem persists even in the variational approximations of Li et al. (2009), who attempt to solve the former.

We address these problems with Monte Carlo techniques. Our solution is a Gibbs sampler that draws samples from the posterior distribution of a phrase-based translation model (Sect. 2). Experiments reveal that our sampler effectively explores the posterior distribution (Sect. 3) and enables maximum probability and minimum risk decoding (Sect. 4). We present new results on three datasets showing that these techniques give competitive results with respect to the standard phrase-based MT pipeline (Sect. 5).

2 A Gibbs sampler for phrase based statistical machine translation

A phrase-based translation model (Koehn et al. 2003) segments input sentence f of length m into phrases, which are sequences of adjacent words. Each phrase is translated into a target phrase, producing an output sentence e and an alignment a representing the mapping from source to target positions. Phrases are also reordered during translation.

We use a log-linear model on features $\mathbf{h}$, parametrised by weights θ.

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

A parameter Λ limits the number of source language words that intervene between adjacent target phrases. In our experiments, $\Lambda = 6$.

Gibbs Sampling We use Markov chain Monte Carlo (MCMC) sampling for inference in this model (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. Eventually,