Improving the IBM Alignment Models Using Variational Bayes

Darcey Riley and **Daniel Gildea**
Computer Science Dept.
University of Rochester
Rochester, NY 14627

Abstract

Bayesian approaches have been shown to reduce the amount of overfitting that occurs when running the EM algorithm, by placing prior probabilities on the model parameters. We apply one such Bayesian technique, variational Bayes, to the IBM models of word alignment for statistical machine translation. We show that using variational Bayes improves the performance of the widely used GIZA++ software, as well as improving the overall performance of the Moses machine translation system in terms of BLEU score.

1 Introduction

The IBM Models of word alignment (Brown et al., 1993), along with the Hidden Markov Model (HMM) (Vogel et al., 1996), serve as the starting point for most current state-of-the-art machine translation systems, both phrase-based and syntax-based (Koehn et al., 2007; Chiang, 2005; Galley et al., 2004).

Both the IBM Models and the HMM are trained using the EM algorithm (Dempster et al., 1977). Recently, Bayesian techniques have become widespread in applications of EM to natural language processing tasks, as a very general method of controlling overfitting. For instance, Johnson (2007) showed the benefits of such techniques when applied to HMMs for unsupervised part of speech tagging. In machine translation, Blunsom et al. (2008) and DeNero et al. (2008) use Bayesian techniques to learn bilingual phrase pairs. In this setting, which involves finding a segmentation of the input sentences into phrasal units, it is particularly important to control the tendency of EM to choose longer phrases, which explain the training data well but are unlikely to generalize.

However, most state-of-the-art machine translation systems today are built on the basis of word-level alignments of the type generated by GIZA++ from the IBM Models and the HMM. Overfitting is also a problem in this context, and improving these word alignment systems could be of broad utility in machine translation research.

Moore (2004) discusses details of how EM overfits the data when training IBM Model 1. He discovers that the EM algorithm is particularly susceptible to overfitting in the case of rare words, due to the "garbage collection" phenomenon. Suppose a sentence contains an English word e_1 that occurs nowhere else in the data, and its French translation f_1. Suppose that same sentence also contains a word e_2 which occurs frequently in the overall data but whose translation in this sentence, f_2, co-occurs with it infrequently. If the translation $t(f_2|e_2)$ occurs with probability 0.1, then the sentence will have a higher probability if EM assigns the rare word and its actual translation a probability of $t(f_1|e_1) = 0.5$, and assigns the rare word's translation to f_2 a probability of $t(f_2|e_1) = 0.5$, than if it assigns a probability of 1 to the correct translation $t(f_1|e_1)$. Moore suggests a number of solutions to this issue, including add-n smoothing and initializing the probabilities based on a heuristic rather than choosing uniform probabilities. When combined, his solutions cause a significant decrease in alignment error rate (AER). More recently, Mermer and Saraclar (2011) have added a Bayesian prior to IBM Model 1 using Gibbs sampling for inference, showing improvements in BLEU scores.

In this paper, we describe the results of incorpo-

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 306–310,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

rating variational Bayes (VB) into the widely used GIZA++ software for word alignment. We use VB both because it converges more quickly than Gibbs sampling, and because it can be applied in a fairly straightforward manner to all of the models implemented by GIZA++. In Section 2, we describe VB in more detail. In Section 3, we present results for VB for the various models, in terms of perplexity of held-out test data, alignment error rate (AER), and the BLEU scores which result from using our version of GIZA++ in the end-to-end phrase-based machine translation system Moses.

2 Variational Bayes and GIZA++

Beal (2003) gives a detailed derivation of a variational Bayesian algorithm for HMMs. The result is a very slight change to the M step of the original EM algorithm. During the M step of the original algorithm, the expected counts collected in the E step are normalized to give the new values of the parameters:

$$\theta_{x_i|y} = \frac{E[c(x_i|y)]}{\sum_j E[c(x_j|y)]} \qquad (1)$$

The variational Bayesian M step performs an inexact normalization, where the resulting parameters will add up to less than one. It does this by passing the expected counts collected in the E step through the function $f(v) = \exp(\psi(v))$, where ψ is the digamma function, and α is the hyperparameter of the Dirichlet prior (Johnson, 2007):

$$\theta_{x_i|y} = \frac{f(E[c(x_i|y)] + \alpha)}{f(\sum_j (E[c(x_j|y)] + \alpha))} \qquad (2)$$

This modified M step can be applied to any model which uses a multinomial distribution; for this reason, it works for the IBM Models as well as HMMs, and is thus what we use for GIZA++.

In practice, the digamma function has the effect of subtracting 0.5 from its argument. When α is set to a low value, this results in “anti-smoothing”. For the translation probabilities, because about 0.5 is subtracted from the expected counts, small counts corresponding to rare co-occurrences of words will be penalized heavily, while larger counts will not be affected very much. Thus, low values of α cause the algorithm to favor words which co-occur frequently and to distrust words that co-occur rarely. In this way, VB controls the overfitting that would otherwise occur with rare words. On the other hand, higher values of α can be chosen if smoothing is desired, for instance in the case of the alignment probabilities, which state how likely a word in position i of the English sentence is to align to a word in position j of the French sentence. For these probabilities, smoothing is important because we do not want to rule out any alignment altogether, no matter how infrequently it occurs in the data.

Sentence pair	count
e_2 f_3	9
e_2 f_2	2
e_1 e_2 f_1 f_2	1

Table 1: An example of data with rare words.

We implemented VB for the translation probabilities as well as for the position alignment probabilities of IBM Model 2. We discovered that adding VB for the translation probabilities improved the performance of the system. However, including VB for the alignment probabilities had relatively little effect, because the alignment table in its original form does some smoothing during normalization by interpolating the counts with a uniform distribution. Because VB can itself be a form of smoothing, the two versions of the code behave similarly. We did not experiment with VB for the distortion probabilities of the HMM or Models 3 and 4, as these distributions have fewer parameters and are likely to have reliable counts during EM. Thus, in Section 3, we present the results of using VB for the translation probabilities only.

3 Results

First, we ran our modified version of GIZA++ on a simple test case designed to be similar to the example from Moore (2004) discussed in Section 1. Our test case, shown in Table 1, had three different sentence pairs; we included nine instances of the first, two instances of the second, and one of the third.

Human intuition tells us that f_2 should translate to e_2 and f_1 should translate to e_1. However, the EM algorithm without VB prefers e_1 as the translation

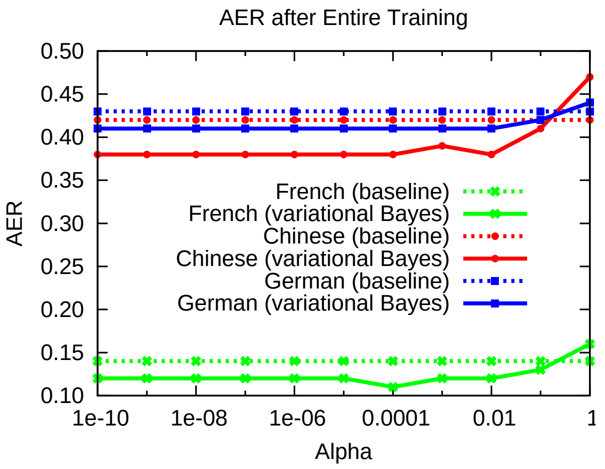

Figure 1: Determining the best value of α for the translation probabilities. Training data is 10,000 sentence pairs from each language pair. VB is used for Model 1 only. This table shows the AER for different values of α after training is complete (five iterations each of Models 1, HMM, 3, and 4).

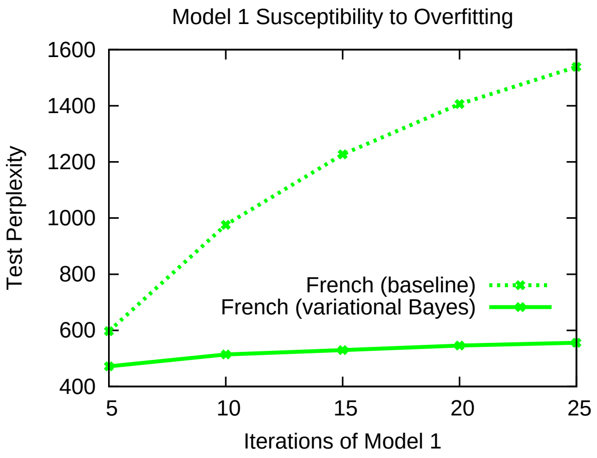

Figure 2: Effect of variational Bayes on overfitting for Model 1. Training data is 10,000 sentence pairs. This table contrasts the test perplexities of Model 1 with variational Bayes and Model 1 without variational Bayes after different numbers of training iterations. Variational Bayes successfully controls overfitting.

of f_2, due to the “garbage collection” phenomenon described above. The EM algorithm with VB does not overfit this data and prefers e_2 as f_2’s translation.

For our experiments with bilingual data, we used three language pairs: French and English, Chinese and English, and German and English. We used Canadian Hansard data for French-English, Europarl data for German-English, and newswire data for Chinese-English. For measuring alignment error rate, we used 447 French-English sentences provided by Hermann Ney and Franz Och containing both sure and possible alignments, while for German-English we used 220 sentences provided by Chris Callison-Burch with sure alignments only, and for Chinese-English we used the first 400 sentences of the data provided by Yang Liu, also with sure alignments only. For computing BLEU scores, we used single reference datasets for French-English and German-English, and four references for Chinese-English. For minimum error rate training, we used 1000 sentences for French-English, 2000 sentences for German-English, and 1274 sentences for Chinese-English. Our test sets contained 1000 sentences each for French-English and German-English, and 686 sentences for Chinese-English. For scoring the Viterbi alignments of each system against gold-standard annotated alignments, we use the alignment error rate (AER) of Och and Ney (2000), which measures agreement at the level of pairs of words.

We ran our code on ten thousand sentence pairs to determine the best value of α for the translation probabilities $t(f|e)$. For our training, we ran GIZA++ for five iterations each of Model 1, the HMM, Model 3, and Model 4. Variational Bayes was only used for Model 1. Figure 1 shows how VB, and different values of α in particular, affect the performance of GIZA++ in terms of AER. We discover that, after all training is complete, VB improves the performance of the overall system, lowering AER (Figure 1) for all three language pairs. We find that low values of α cause the most consistent improvements, and so we use $\alpha = 0$ for the translation probabilities in the remaining experiments. Note that, while a value of $\alpha = 0$ does not define a probabilistically valid Dirichlet prior, it does not cause any practical problems in the update equation for VB.

Figure 2 shows the test perplexity after GIZA++ has been run for twenty-five iterations of Model 1: without VB, the test perplexity increases as training continues, but it remains stable when VB is used. Thus, VB eliminates the need for the early stopping that is often employed with GIZA++.

After choosing 0 as the best value of α for the

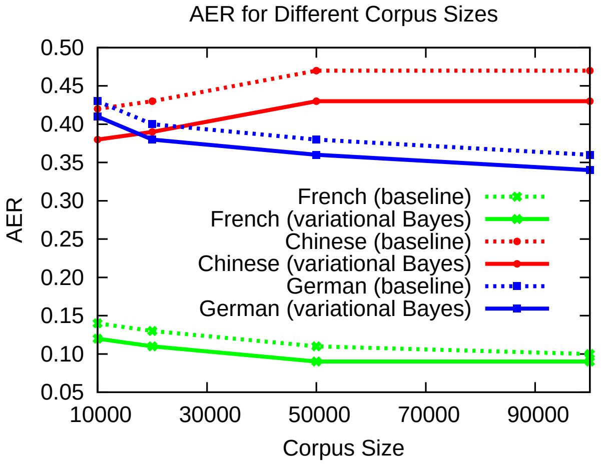

Figure 3: Performance of GIZA++ on different amounts of test data. Variational Bayes is used for Model 1 only. Table shows AER after all the training has completed (five iterations each of Models 1, HMM, 3, and 4).

	AER		
	French	Chinese	German
Baseline	0.14	0.42	0.43
M1 Only	0.12	0.39	0.41
HMM Only	0.14	0.42	0.42
M3 Only	0.14	0.42	0.43
M4 Only	0.14	0.42	0.43
All Models	0.19	0.44	0.45

Table 2: Effect of Adding Variational Bayes to Specific Models

translation probabilities, we reran the test above (five iterations each of Models 1, HMM, 3, and 4, with VB turned on for Model 1) on different amounts of data. We found that the results for larger data sizes were comparable to the results for ten thousand sentence pairs, both with and without VB (Figure 3).

We then tested whether VB should be used for the later models. In all of these experiments, we ran Models 1, HMM, 3, and 4 for five iterations each, training on the same ten thousand sentence pairs that we used in the previous experiments. In Table 2, we show the performance of the system when no VB is used, when it is used for each of the four models individually, and when it is used for all four models simultaneously. We saw the most overall improvement when VB was used only for Model 1; using VB for all four models simultaneously caused the most improvement to the test perplexity, but at the cost of the AER.

	BLEU Score		
	French	Chinese	German
Baseline	26.34	21.03	21.14
M1 Only	26.54	21.58	21.73
All Models	26.46	22.08	21.96

Table 3: BLEU Scores

For the MT experiments, we ran GIZA++ through Moses, training Model 1, the HMM, and Model 4 on 100,000 sentence pairs from each language pair. We ran three experiments, one with VB turned on for all models, one with VB turned on for Model 1 only, and one (the baseline) with VB turned off for all models. When VB was turned on, we ran GIZA++ for five iterations per model as in our earlier tests, but when VB was turned off, we ran GIZA++ for only four iterations per model, having determined that this was the optimal number of iterations for baseline system. VB was used for the translation probabilities only, with α set to 0.

As can be seen in Table 3, using VB increases the BLEU score for all three language pairs. For French, the best results were achieved when VB was used for Model 1 only; for Chinese and German, on the other hand, using VB for all models caused the most improvements. For French, the BLEU score increased by 0.20; for German, it increased by 0.82; for Chinese, it increased by 1.05. Overall, VB seems to have the greatest impact on the language pairs that are most difficult to align and translate to begin with.

4 Conclusion

We find that applying variational Bayes with a Dirichlet prior to the translation models implemented in GIZA++ improves alignments, both in terms of AER and the BLEU score of an end-to-end translation system. Variational Bayes is especially beneficial for IBM Model 1, because its lack of fertility and position information makes it particularly susceptible to the garbage collection phenomenon. Applying VB to Model 1 alone tends to improve the performance of later models in the training sequence. Model 1 is an essential stepping stone in avoiding local minima when training the following models, and improvements to Model 1 lead to improvements in the end-to-end system.

References

Matthew J. Beal. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, University College London.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. Bayesian synchronous grammar induction. In *Neural Information Processing Systems (NIPS)*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL-05*, pages 263–270, Ann Arbor, MI.

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*, 39(1):1–21.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 314–323, Honolulu, Hawaii, October.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of NAACL-04*, pages 273–280, Boston.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, Demonstration Session*, pages 177–180.

Coskun Mermer and Murat Saraclar. 2011. Bayesian word alignment for statistical machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-11)*, pages 182–187.

Robert C. Moore. 2004. Improving IBM word alignment Model 1. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 518–525, Barcelona, Spain, July.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL-00*, pages 440–447, Hong Kong, October.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING-96*, pages 836–841.