An Empirical Evaluation of IBM 1 and IBM 2 Translation Models

First Author
Maartje de Jonge
0194107
maartjedejonge@gmail.com

Second Author
Lina Murady
10776389
lina.murady@gmail.com

Abstract

In this report we perform an empirical evaluation of the IBM 1 and IBM 2 alignment models on a French-English parallel corpus. We trained both models with expectation maximization. Our results show that the IBM 2 model gave better performance than the IBM 1 model on our corpus. We also experimented with variational Bayes on IBM 1, but the results were inconclusive.

1 Introduction

The objective in statistical machine translation is to find the most probable sentence in a target language e, given a sentence in a source language f. Using Bayes rule we can describe the model mathematically as $\hat{e}_1^I = argmax_{e_1^I}\ p(e_1^I|f_1^J) = argmax_{e_1^I}\ p(e_1^I) \cdot p(f_1^J|e_1^I)$. The resulting equation has the advantage that it includes a language model $p(e_1^I)$ that expresses the probability of a translated sentence in the target language. In this report we focus on the translation model $p(f_1^J|e_1^I)$.

A key issue in modeling the translation probability of a sentence pair is to define the probability of a word pair (f_j, e_i), assuming independence of other word pairs. Typically the additional assumption is made that each foreign word f_j is aligned with exactly one English word e_i. Furthermore, the NULL word is added to each English sentence to align words in the foreign sentence that do not have an equivalent word in the English sentence.

In this report we compare two well known and widely used alignment models: IBM 1 and IBM 2 [1]. We train both models based on maximum likelihood estimation, using expectation maximization as the optimization method. Our empirical evaluation on a French-English corpus shows that the IBM 2 model gives better performance than the IBM 1 model. For IBM 1 we also experimented with a a Bayesian approach using variational inference, however, these results were inconclusive.

2 Models

2.1 IBM 1

We are interested in modeling the probability of a French sentence given an English sentence. First we decompose the probability of the French sentence into the probability of the sentence length multiplied by the product over the probabilities of each French word: $p(f_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} p(f_j|e_1^I)$. We then further decompose this equation assuming a pairwise dependency between French and English words; which we describe by a mixture model. This leads to the equation: $p(f_1^J|e_1^I) = p(J|I) \cdot \prod_{j=1}^{J} \sum_{i=1}^{I} [p(i|j, I, J)] \cdot p(f_j|e_i)]$. The equation can be understood as a decomposition into a sentence length probability, alignment probabilities between positions and lexical translation probabilities.

The IBM1 model [1] assumes that the alignment probabilities are uniformly distributed, i.e. $p(i|j, I, J) = \frac{1}{I}$. Furthermore, we can ignore the sentence length probability since we are only interested in finding the most probable word alignments. To learn the lexical probabilities we apply the maximum likelihood criterion to a bilingual corpus. Since this criterion results in an iterative function that can not be optimized analytically, we apply expectation maximization (EM) to estimate the lexical probabilities. In the case of uniform alignment probabilities the function to optimize is convex [1], therefore the EM algorithm always converges to the global optimum.

2.2 IBM 1 with Variational Inference

A limitation of the maximum likelihood estimation is that the probability distribution only takes into account the most likely parameter estimation, ignoring contributions from other possible parameter settings. This limitation can be addressed by taking a Bayesian approach, weighting the parameters by some prior belief.

In the IBM 1 model the Dirichlet distribution is chosen as the prior for the lexical translation probabilities since it is a conjugate prior to a categorical distribution. Furthermore, it can express the prior belief of how likely it is that many translation candidates exist for a given word in the source language. The hyperparameter α controls exactly this belief by controlling the sparsity of the distribution. A small α corresponds to a sparse distribution which is preferable in our case.

However, putting a Dirichlet distribution as a prior to the translation parameters leads to an intractable posterior. A possible approach is to draw samples from the posterior using an approximation technique such as MCMC. Unfortunately, drawing enough samples can be slow in practice. We therefore use variational inference instead.

Variational inference introduces an approximating distribution with the objective to minimize the Kullback-Leiber(KL) divergence. However, minimizing the KL divergence directly is not possible due to the intractable posterior. Fortunately, it can be shown that minimizing the KL divergence is equivalent to maximizing a simpler objective known as the evidence lower bound (ELBO) which can be computed. A mathematical derivation can be found in [2].

2.3 IBM 2

The IBM1 model assumes that all alignment probabilities are uniformly likely, which is a simplification that leads to sub-optimal results in practice. The IBM2 model [1] therefore does not make this assumption but instead tries to learn the alignment probabilities during training. A disadvantage of the IBM2 model is given by the large number of parameters to learn, which is especially problematic for small corpora.

For this reason we assume a specific model for the alignment probabilities, namely the jump probability model described in [3]. This model assumes that the distance relative to the diagonal is the most important factor. The alignment probabilities are then described by $p(i|j, I, J) = \frac{r(i-j\lfloor \frac{I}{J} \rfloor)}{\sum_{i'=1}^{I} r(i'-j\lfloor \frac{I}{J} \rfloor)}$. Alignment of French words to the artificial English NULL word can result in large jumps that are not actually meaningful and therefore cause noise in the jump probability distribution. For this reason we introduce a separate probability for jumps that include the engish NULL word.

The values for $r(x)$ are learned during training together with the lexical probabilities. As with the IBM 1 model, we can use expectation maximization to find the maximum likelihood estimate. An important difference however is that the function to optimize is non-convex. Therefore, the EM algorithm converges to a local optimum and the learned probabilities depend on the initialization. For this reason it is recommended to train the model with different initializations.

3 Experiments

3.1 Experimental Setup

All experiments conducted in this report use a French-English parallel corpus taken from the Canadian Hansards parliament proceedings. The data is split into a training, validation and test set containing respectively 231.164, 37 and 447 sentences. We take English as the source language and French as the target language; that is, we model the probability $p(f_1^J|e_1^I)$.

The IBM models use an iterative optimization process to estimate the model parameters. We implemented a convergence criterion based on the value to optimize, i.e. the training log likelihood for MLE and the training ELBO for variational Bayes. Since the log likelihood of the data in general is lower for larger data sets and since we want our criterion to work for data sets of different sizes, we choose a relative criterion: $\frac{LL_i - LL_{i-1}}{LL_{i-1}} < \epsilon$ with i the iteration number and LL the log likelihood (or ELBO). We set ϵ to be 0.001.

As an alternative convergence criterion we use the AER on the validation data. In contrast to the log likelihood, the AER is not guaranteed to monotonely increase. We therefore decided to select a model as soon as the AER increases for the next iteration, i.e. $AER_i < AER_{i+1}$.

Using these convergence criteria we select two models for every IBM variant that we investigate in this report. To evaluate the performance of the selected models we first obtain the viterbi align-

ments for every sentence pair in the test set and then compute the AER using human annotated alignments as the gold standard.

The sentences in the test and validation set may contain words or word pairs that do not occur in the training data. To handle this situation we map all words that occur a single time in the training data to the symbol 'LOW' and replace all unknown words in the validation and test set by the 'LOW' symbol.

3.2 IBM 1 with Expectation Maximization

We trained the IBM1 model for 15 iterations using EM with uniform initialization of the lexical probabilities. The results are shown in figure 1. We see that the log likelihood monotonely increases, while the AER decreases fast in the beginning and then varies between a small bandwidth.

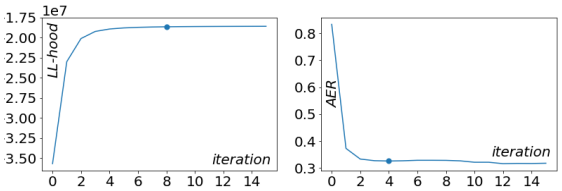

Figure 1: Evolution of training log likelihood (left) and validation AER (right) for the IBM 1 model. The selected models are indicated with a dot.

We used our model selection criteria to select a model based on log-likelihood and a model based on AER. The selected models are indicated with a dot in figure 1. For these models we calculated the AER score on the test data. We got the following results. The log likelihood model was selected after 8 iterations, which resulted in an AER score of 0.2926. The AER model was selected after 4 iterations, which resulted in an AER score of 0.2964. While the log-likelihood model performed slightly better on the test set, the AER criterion at the other hand required less iterations to obtain a comparable result.

3.3 IBM 1 with Variational Inference

The IBM 1 model with Variational Inference was uniformly initialized. Furthermore, we tuned the hyperparameter α by choosing the α that gave the lowest AER score on the validation set. As figure 2 shows, $\alpha = 0.1$ gave the lowest score. For choosing α the whole training set was considered. However, when experimenting with the evolution of the ELBO and AER values as a function of the iteration, we trained the model on a random subset of 20.000 sentences. The reason is that computing the ELBO while training the model turned out to be computationally expensive.

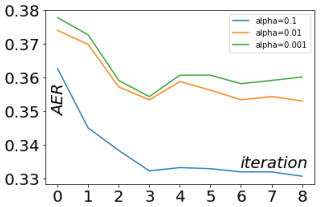

Figure 2: AER scores on the validation set for different values for the hyperparameter α

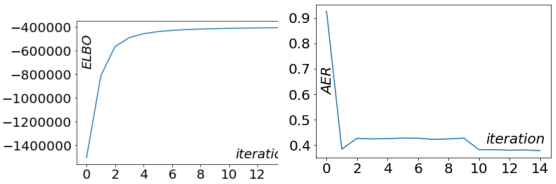

Figure 3: Evolution of the training ELBO for IBM1 with variational inference (left). Evolution of the validation AER for IBM1 with variational inference (right).

We see that the behaviour of the ELBO in figure 3 is similar to the behaviour of the log likelihood in the IBM1 model trained with EM. However, the behaviour of the AER scores in figure 3 shows some differences. The AER scores of the variational inference seems to have a small increase after the fast decrease in the beginning, and a small decrease later. We assume that this behavior occurs due to statistical change.

The AER criterion selected the second iteration, the ELBO criterion however did not led to conversion within fifteen iterations; we therefore selected the model of the last iteration. For the selected AER and ELBO models the AER score obtained on the test set was respectively 0.4205 and 0.3582.

3.4 IBM 2 with Expectation Maximization

The IBM2 model is non-convex which means that the training result depends on the parameter initialization. We experimented with different initializations, namely: random (3 times), uniform and staged. For the staged initialization we initialized the lexical probabilities using the output of the IBM1 model selected by the relevant selection criterion. The jump probabilities are initialized uniformly for the staged initialization.

Figure 4 shows the training log-likelihood and the validation AER for IBM2 for all initializations plotted for 15 iterations. Since the model is non-convex, we expected different log-likelihood results after 15 iterations. However, the log likeli-

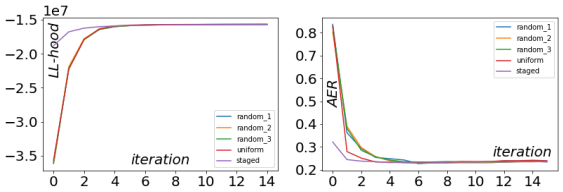

Figure 4: Evolution of training log likelihood (left) and validation AER (right) for the IBM 2 model with different initializations.

hood plot does not shows a clear difference in convergence for the different initializations. A possible explanation might be that real local maxima do not often occur in a high dimensional space. It would be interesting to repeat the experiment with a much smaller dataset. For the AER values we see that random initialization leads to worse AER scores in the first couple of iterations compared to uniform initialization. Interestingly we do not see this pattern in the log likelihood plot. Intuitively it makes sense that equal values are easier to tune in a small number of iterations.

Figure 5 shows the log-likelihood and AER values of the models selected based on the log-likelihood respectively AER criterion for the different initializations. Based on these values we select the uniform model as our optimal log-likelihood model and the first randomly initialized model as our best AER model.

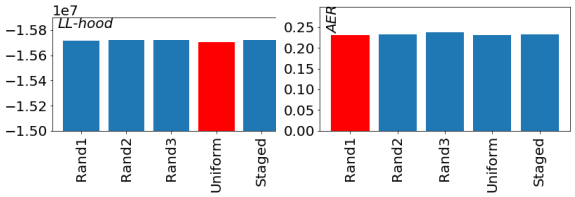

Figure 5: Comparing the selected models for IBM2 using different initializations. We select the model with the least negative log likelihood (left) and the model with the lowest AER score (right).

3.5 Comparison of Models

For our final evaluation we calculated the AER on the test set for the selected IBM1 and IBM2 models trained with EM. The results are shown in figure 6(left). We conclude that the IBM2-MLE model gives better performance on our corpus than the IBM1-MLE model.

In order to properly compare IBM1-MLE with IBM1-VB we also selected models for IBM1-MLE trained on the subset of 20.000 sentences that were used to train IBM1-VB. The results are shown in figure 6 (right). The figure shows that IBM1-VB does not improves the alignment quality compared to IBM1-MLE but instead performs slightly worse. We consider this result as preliminary since it was obtained using a small training set.

Figure 6 suggests that the AER and the log-likelihood convergence criteria give models that are comparable in performance. An advantage of the AER criterion is that it does not depend on an 'arbitrary' chosen parameter ϵ; an advantage of the log likelihood criterion at the other hand is that the log likelihood is guaranteed to monotonely increase.

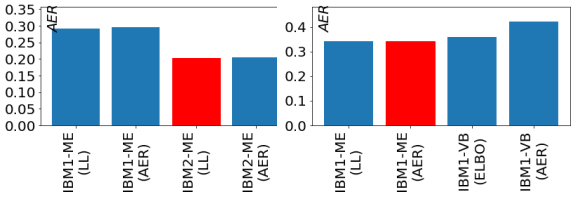

Figure 6: Comparison of the performance of the selected models on the test set. Left: IBM1-MLE and IBM2-MLE, both models trained on the full training set. Right: IBM1-MLE and IBM1-VB, both models trained on 20.000 sentences.

4 Conclusion and Future Work

In this report we compared the IBM 1 and IBM 2 alignment models, considering both maximum likelihood estimation and bayesian inference for IBM 1. For the IBM 2 model we assumed a specific alignment probability model that models jumps relative to the diagonal defined by the positions of a sentence pair. Our empirical evaluation showed that the IBM 2 model gave the best performance on the French-English parallel corpus that we used in our experiments.

An obvious limitation of our evaluation is that we only considered one language pair, namely French to English. Especially for the IBM 2 model it would be interesting to experiment with language pairs that are more different in word order than English and French; or with languages that have a mostly free word order. We recommend for future work to experiment with more divers languages.

Another idea that did not fit into the scope of this project is to take a more close look into the probability distributions that resulted from the models. Specifically, it would be interesting to plot the jump probability distribution for IBM 2; and to inspect lexical probabilities for both rare and common words, while comparing the MLE and VB approaches of IBM 1. We leave this as a recommendation for future work.

References

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

[2] Matthew James. Beal. Variational algorithms for approximate bayesian inference /. 01 2003.

[3] Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.