# NLP2 project 1

**Tim van Elsloo**
10590315

**Fije van Overeem**
10373535

**Daan van Stigt**
10255141

## Abstract

This paper reports on a series of experiments conducted with the IBM models 1 and 2. For both models the parameters are estimated from a parallel corpus. These parameters are used to predict alignments on a test set. The alignments are evaluated against gold standard alignments and the alignment error rate (AER) is recorded. We experiment with two different methods of parameter estimation for IBM1, and three different methods of initialising the parameters of IBM2.

## 1 Introduction

The IBM models (Brown et al., 1993) lay the groundwork for much current research in statistical machine translation. Although the models were originally designed for the task of full translation, they are nowadays used mostly for the prediction of word alignments in translated sentence pairs. IBM1 and IBM2 are statistical models.

In this paper we will report on the results of a series of experiments performed with the IBM1 and IBM2 models. For IBM1 we experiment with two different methods for parameter-estimation: expectation maximisation (EM); and variational inference, for a Bayesian formulation of the model. For IBM2 we experiment with three different methods of initialising the translation parameters: uniformly, like in IBM1; randomly; and initialisation using pre-trained parameters obtained from a trained IBM1 model. Because of the non-convexity of IBM2, each of these initialisations yield different results.

## 2 Models

The IBM models 1 and 2 are both alignment models: they predict how each word in a sentence aligns to a word in the translated sentence. The two models are identical and differ only in their alignment assumptions. IBM1 assumes that alignments between any two sentences positions are equally likely; IBM2 additionally trains these alignments. We discuss how the two models are defined below, starting with IBM1.

### 2.1 IBM 1

The simplest model that we have implemented is IBM Model 1. Following Schulz (2017a)'s tutorial, we start by defining a joint probability over a parallel corpus $C$ consisting of translated sentence pairs $(e_1^l, f_1^m)$ as follows:

$$P(C) = \prod_{s=1}^{|C|} P(e_1^{s_l}, f_1^{s_m}).$$

This assumes all sentence pairs to be independent[1]. This is the first simplifying assumption used with IBM models that derogates natural languages, whereas sentences are in fact not independent of one another. This assumption is simplifying in the sense that we from now on only have to consider individual sentence pairs, as statements about them are easily extended to a whole corpus because of the independence assumption. We are interested in modelling the probability $P(e_1^{s_l}, f_1^{s_m})$. Using the chain rule of probability this can be decomposed:

$$P(e_1^l, f_1^m) = P(e_1^l)P(f_1^m|e_1^l)$$

in which we can identify a language model $P(e_1^l)$ and the translation model $P(f_1^m|e_1^l)$. The the language model models the probability of the English sentence, and for this any language model could be used. The translation model is the part that IBM1 and IBM2 are designed to model. This is the

[1] Using the multiplication rule for independent events

probability of a French sentence, given an English sentence. To model this probability, IBM1 introduces an alignment variable $a_j$ for each word $f_j$ in the French sentence. For each $j \in \{1, \dots, m\}$ the alignment variable $a_j$ can take any value in $\{0, ...., l\}$ indicating which English word $e_{a_j}$ the French word $f_j$ is aligned with. Thus this alignment variable will align each French word with one word in the English sentence.

The probability $P(f_1^m, a_1^m | e_1^l)$—which includes the alignment variables—is then modelled as:

$$\begin{aligned} P(f_1^m, a_1^m | e_0^l) &= \prod_{j=1}^{m} P(a_j | m, l) P(f_j | e_{a_j}) \\ &= \prod_{j=1}^{m} \frac{1}{l+1} P(f_j | e_{a_j}) \\ &= \frac{1}{(l+1)^m} \prod_{j=1}^{m} P(f_j | e_{a_j}) \end{aligned}$$

where $P(a_j|m,l) = \frac{1}{l+1}$ expresses the fact that in IBM1 the probability of French word $j$ aligning to English word $a_j$ given sentence lengths $m$ and $l$ is assumed to be uniform. Marginalizing the $a_j$ gives us then that

$$P(f_1^m | e_0^l) = \prod_{j=1}^{m} \frac{1}{(l+1)^m} \sum_{a_j=0}^{l} \dot{P}(f_j | e_{a_j}).$$

Hence the full likelihood of the corpus $C$ under the model is given by

$$\prod_{(f_1^m, e_0^l) \in C} \prod_{j=1}^{m} \frac{1}{(l+1)^m} \sum_{a_j=0}^{l} P(f_j | e_{a_j}). \qquad (1)$$

The training task is to estimate the parameters $P(f_j|e_i)$ from the training corpus. We have implemented two methods for this: Expectation Maximisation (EM); and Variational Inference (VI) for the Bayesian version of the above model that will be discussed below.

### 2.1.1 Expectation Maximisation

The EM algorithm is a method to find the maximum likelihood estimates of parameters in a statistical model with latent variables. The EM-algorithm performs two steps alternatingly: in the E-step it gathers expected counts under the approximate distribution obtained in the previous step; in the M-step it updates the parameters by choosing the parameters that maximizes the log-likelihood of the counts gathered in the E-step.

In the case of IBM1 the E-step consists of collecting expected counts for the events

$$\begin{aligned} c(e,f) &= \mathbb{E}[\#(e \rightarrow f)] \\ c(e) &= \sum_f c(e,f) \end{aligned}$$

and in the M-step $P(f|e)$ is re-estimated as:

$$P(f|e) = \frac{c(e,f)}{c(e)}$$

Hence, the expectation step consists of iterating through all sentence pairs in the dataset and counting the co-occurrence of words in for pair using expected counts. So we count $P(f|e)$, the number of times a French word $f$ occurs in a translation of a sentence that contains English word $e$.

### 2.1.2 Variational Inference

Our second experiment for IBM Model 1 is to extend it into a Bayesian model. We now assume a prior we now assume a Dir$(\alpha)$ prior on our translation parameters. The intractable posterior that we obtain this way is approximated using the method of variational inference:

$$\phi_j = \frac{\exp\left(\Psi\left(\lambda_{f_j|e_{a_j}}\right) - \Psi\left(\sum_f \lambda_{f|e_{a_j}}\right)\right)}{\sum_{i=0}^{m} \exp\left(\Psi\left(\lambda_{f_j|e_i}\right) - \Psi\left(\sum_f \lambda_{f|e_i}\right)\right)}$$

$$\lambda_{f|e} = \alpha_f + \sum_{(e_0^m, f_1^n)} \sum_{j=1}^{n} \mathbb{E}_{Q(A_j|\phi_j)}[\#(e \rightarrow f | A_j)]$$

After conducting a brief pilot study, we have selected a symmetric prior $\alpha = 10^{-1}$.

### 2.1.3 ELBO

For the Bayesian model, we use the evidence lower bound (ELBO) instead of the likelihood. We can rephrase our likelihood maximization problem into a KL-divergence minimalization problem. However, we can't minimize the KL-divergence directly. The ELBO provides a lower bound of the evidence and maximising the ELBO minimizes the KL-divergence, which in turn maximizes the likelihood function.

$$\sum_{j=1}^{m} \mathbb{E}_q[\log P(a_j|m) P(f_j | e_{a_j}, \theta) - \log Q(a_j|\phi_j)] \\ + \sum_e \mathbb{E}_q[\log p(\theta_e|\alpha) - \log q(\theta_e|\lambda_e)]$$

We have computed the ELBO using the equations published in Schulz (2017b).

## 2.2 IBM 2

IBM2 revisits the independence assumption on the alignment probabilities $P(a_j|j,m,l)$ made by IBM1. In IBM2, these probabilities are additionally estimated from the data. This means that for IBM2 the joint probability of a French sentence $f_1^m$ and alignments $a_1^m$ given an English sentence $e_1^l$ is given by

$$P(f_1^m, a_1^m|e_1^l) = \prod_{j=1}^{m} P(a_j|j,m,l)P(f_j|e_{a_j}).$$

Note that with $P(a_j|j,m,l)$ we have now introduced a probability distribution over English sentence positions $a_j \in \{0,\dots,l\}$ for *each triple* $(j,m,l)$ of French word position $j$, French sentence length $m$, and English sentence length $l$.

This means that for IBM2 the complete likelihood of the corpus is given by:

$$\prod_{(f_1^m, e_0^l)\in C} \prod_{j=1}^{m} \sum_{a_j=0}^{l} P(a_j|j,m,l)P(f_j|e_{a_j}). \quad (2)$$

### 2.2.1 Jump parametrisation

The above parametrisation of the alignment probability $P(a_j = i|j,m,l)$ is very costly. In order to make estimation of this probability more efficient a different parametrisation based on a so-called jump-function is used, following (Vogel et al., 1996). First, we define a jump-function

$$\delta(i,j,m,l) = i - \left\lfloor l\frac{j}{m} \right\rfloor$$

for each $i,j,m,l$, where $i \in \{0,\dots,l\}$ and $j \in \{1,\dots,m\}$. The function $\delta$ measures how much of a jump is made when a word in the $j$th position of a sentence of length $m$ is aligned to a word in the $i$th position in sentence of length $l$, scaled to the respective sentence lengths.

The alignment distribution is then defined as

$$P(a_j = i|j,m,l) = \text{Cat}(\Delta|\theta_{\delta(i,j,l,m)})$$

where: $\Delta = (-k,\dots,0,\dots,k)$ is the range of possible jumps that an alignment can make, for some maximal value for an alignment-jump $k$; $\theta = (\theta_{-k},\dots,\theta_k)$ is the parameter-vector that holds the probabilities for each jump-event $-k$ to $k$; and $\delta(i,j,l,m)$ is used to determine the index in this vector and hence the probability of jumping a distance $\delta(i,j,l,m)$.

### 2.2.2 Parameter estimation

Parameter estimation in IBM2 is performed only with EM. The estimation of is identical to IBM1 with the exception that the expected counts in the E-step are now gathered with respect to the the likelihood in formula (2), which includes the jump probabilities.

## 3 Prediction of alignments

Given their estimated probabilities, IBM1 and IBM2 can predict alignments. For this we need the posterior probability of an alignment link, given by

$$\begin{aligned} P(a_j = i|f_j, e_0^l, m) &= \frac{P(i|j,m,l)P(f_j|e_i)}{\sum_{i=0}^{l} P(i|j,m,l)P(f_j|e_i)} \\ &= \frac{P(f_j|e_i)}{\sum_{i=0}^{l} P(f_j|e_i)} \text{ for IBM1.} \end{aligned}$$

When aligning two sentences we pick the alignment $a_1^m$ with the highest posterior probability. Since alignments are assumed to be independent we align each French word independently: we align French word $f_j$ to the English word $e_{a_j}$ for which $P(a_j = i|f_j, e_0^l, m)$ is maximal. This gives us our predicted sentence alignment.

## 4 Experiments

In order to compare IBM1-EM, IBM1-VI and IBM2 we propose an experimental setup that compares the alignment error rates (AERs) of each model that we have presented in this paper.

Our training data consists of 231k sentences selected from Canadian parliamentary proceedings[2]. These sentences were written in either English or French, and then translated to the alternative language by hand.

We run each model on all training data for 14 iterations. At each iteration, we use the model to make predictions on the validation data and compute the AER. For each model, we select the parameters of its best iteration with respect to the log-likelihood of the training data and the AER of the validation data.

[2] The Hansard French/English corpus. Source https://catalog.ldc.upenn.edu/LDC95T20

| | AER | Selected by |
|---|---|---|
| EM | 0.2867 | likelihood |
| VI | 0.4114[3] | likelihood |

Table 1: Results of best models for IBM1 on test set. How models were selected based on validation set is indicated.

| | AER | Selected by |
|---|---|---|
| uniform | 0.2643 | likelihood |
| random 1 | 0.2646 | likelihood |
| random 2 | 0.2741 | likelihood |
| random 3 | 0.2640 | likelihood |
| IBM1 init | 0.2637 | likelihood |

Table 2: Best results for IBM2 on test set. How models were selected based on validation set is indicated.

In addition, we have repeated the training of IBM2-EM with random initialisation three times, in order to determine the magnitude of its significance on the final AER.

Experiments performed on our IBM1-VI model are constrained to 10k sentences due to exhaustion of resources above that level. We will therefore only compare the performance of IBM1-EM and IBM1-VI on 10k sentences both.

The coded implementation can be found on https://github.com/elslooo/nlp2.

### 4.1 Results

In table 1 we demonstrate the AER of the best models for IBM1 on the test set for IBM1-EM and IBM1-VI. In table 2 we see the best AER scores for the IBM2 model with the five different initial parameter settings. The AER of the IBM1-initialised model is the lowest and thus has produced the best alignment links for IBM2. In figure 7 it is demonstrated how the jump distribution gets more and more skewed around 0 after each iteration.

## 5 Conclusion

We have implemented IBM Model 1 and 2 with Expectation Maximisation and Variational Inference. We have seen that IBM1-EM already performs surprisingly well with respect to its AER. In our experiments, IBM1-VI is outperformed by IBM1-EM, however we acknowledge that this is inconclusive due to the lack of training because of constraint resources. IBM Model 2 expands on 1 in the sense that it no longer assumes that the distribution over alignments is uniform, and instead estimates this probability distribution as well. It is interesting to see that the IBM2-runs with random initialisation differ a lot in the beginning but at the end, around epoch 8, converge to the same AER. We could conclude that the initial state for IBM2 does not influence the quality of the alignment after a certain amount of epochs but statistical testing is necessary to ascertain this.

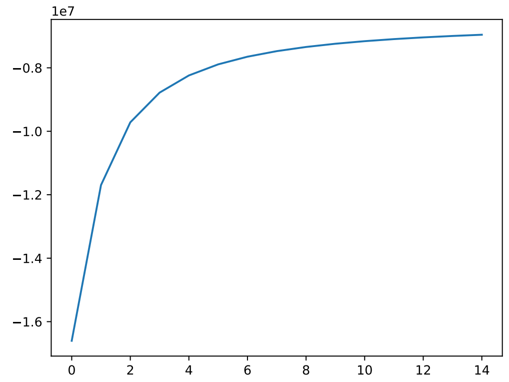

Figure 1: Log-likelihood of the training set for IBM1-EM plotted per epoch

## References

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

Philip Schulz. 2017a The IBM Mixture Models 1 and 2 for Word Alignment. https://uva-slpl.github.io/nlp2/resources/papers/Schulz-IBM12-Tutorial.pdf.

Philip Schulz. 2017b Computing the ELBO of a Dirichlet distribution. https://github.com/philschulz/PublicWriting.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '96, pages 836–841. https://doi.org/10.3115/993268.993313.

[3] Trained on 10k sentences instead of 231k.

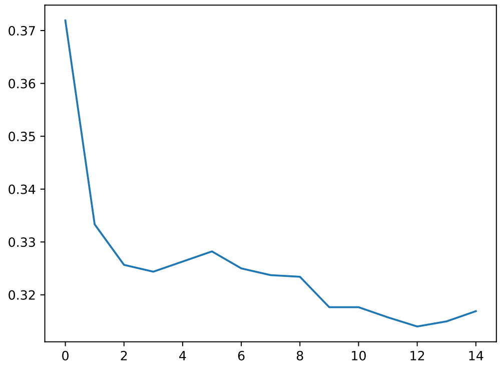

Figure 2: AER on the development set for IBM1-EM plotted per epoch

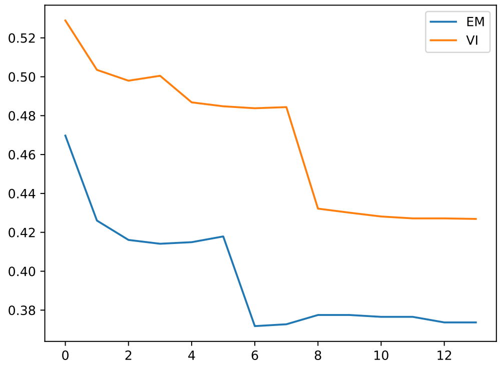

Figure 3: AER on a 10k subset of the development set for IBM1-VI and IBM-EM plotted per epoch

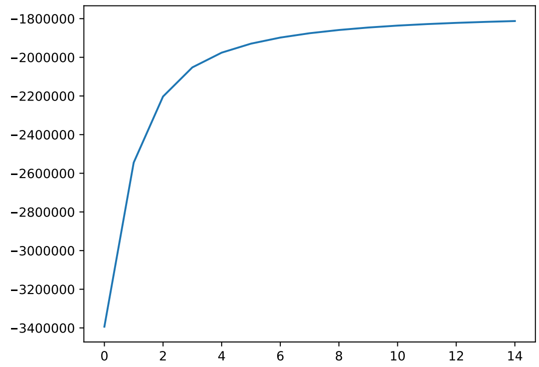

Figure 4: ELBO of the IBM-VI on a 10k subset of the development set plotted per epoch

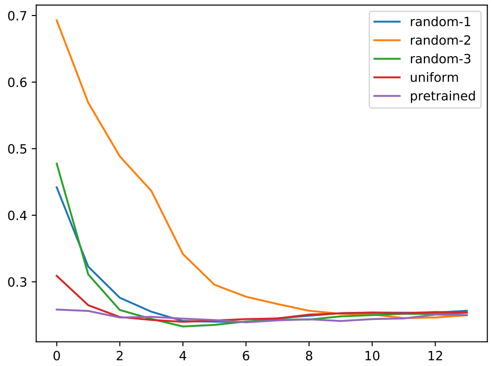

Figure 5: AER on the development set for various initialisations of IBM2 plotted per epoch

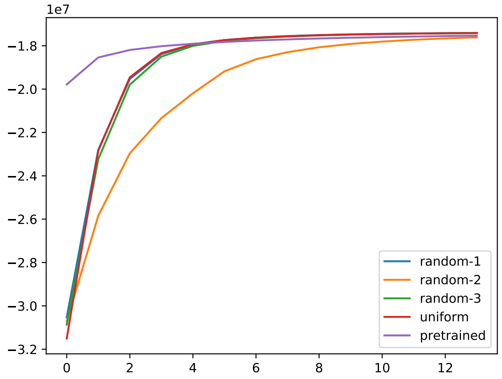

Figure 6: Log-likelihood of the training set for various initialisations of IBM2 plotted per epoch

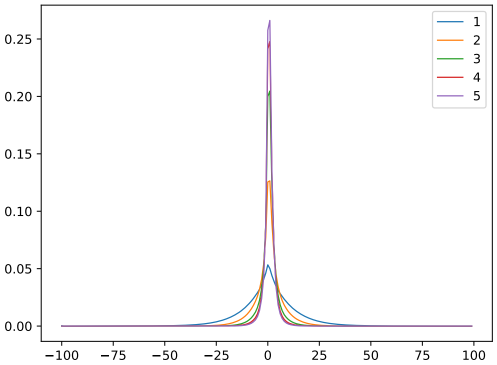

Figure 7: Evolution of the jump distribution over epochs 1-5. Change becomes minimal after 5 epochs.