NLP2 - Assignment 1 Report

Minh Ngo 10897402[1], Peter Dekker 10820973[1]

April 24, 2015

[1] University of Amsterdam
{minh.ngole, peter.dekker}@student.uva.nl

Abstract

IBM Model 1 and IBM Model 2 have been implemented for exploring their problems and possible improvements for boosting the speed of convergence or the quality of sentence alignments. The result of this investigation is introduced below.

1 Introduction

IBM models are word-based models for statistical machine translation. In this report, IBM Model 1 (IBM-M1) and 2 (IBM-M2) have been implemented and evaluated. Furthermore, word alignment from IBM-M1 has been improved by methods proposed by Moore (2004).

IBM Models are translating the **target** language sentence **f** with a length **m** to the **source** language sentence **e** with a length **l** and describe a translation model that assigns a conditional probability $p(e|f)$ (called the translation model) to the pair of target/source sentences that is defined via the Bayes rule (Eqn. 1).

$$p(e|f) = \frac{p(e)p(f|e)}{\sum_f p(e)p(f|e)} \tag{1}$$

where **e** is a sentence of the **source** language, and **f** is a sentence of the **target** language. The model is inverted to be able to use the language model (LM) **p(e)** of the source language.

In the IBM Models the word alignment **a** has been introduced (Eqn. 2). Each word from the target language is aligned to the word from the source one or to the NULL anchor. Each source word and NULL can be linked with several target words but not vice versa.

$$p(f|e) = p(f, a|e, m) = \prod_{i=1}^{m} q(a_i|i, l, m)t(f_i|e_{a_i}) \tag{2}$$

where **q** describes a probability of **i-th** position to be aligned to $\mathbf{a_i}$ for the pair of sentences with lengths of **l** and **m**, **t** is a conditional probability of the word $\mathbf{f_i}$ being a translation of the word $\mathbf{e_a}$.

The best alignment can be determined as the Viterbi one (Eqn. 3).

$$\arg\max_{a_1..a_m} p(a_1..a_m|f_1..f_m, e_1..e_l, m) \tag{3}$$

2 IBM Model 2

The EM algorithm for partially observed data described by Collins (2011) has been implemented to estimate model parameters. In this case the following normalized value δ (Eqn. 4) is used for updating word counts that are needed to estimate new **t** and **q** parameters (Eqn. 5).

$$\delta(k,i,j) = \frac{q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i,l_k,m_k)t(f_i^{(k)}|e_j^{(k)})} \tag{4}$$

$$t(f|e) = \frac{c(e,f)}{c(e)}; \quad q(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)} \tag{5}$$

where **c**(...) are "counts" (or more scores) that are assigned to corresponding conditions.

2.1 Problems of IBM-M2

Unlike IBM-M1, IBM-M2 is **non-convex**. That means that the EM algorithm converges to the local optima instead of the global one. By the intuition that the global convergence of the IBM-M1 can lead to better estimation for the **t** parameters, IBM-M2 is initialized by parameters estimated by IBM-M1.

3 IBM Model 1

For the IBM M1 implementation the IBM-M2 has been simplified by assuming that a conditional probability **q** to be uniform (Eqn 6). The EM framework in this case remains the same.

$$q(j|i,l,m) = \frac{1}{l+1} \tag{6}$$

3.1 Problems of IBM-M1 and IBM-M2

Moore (2004) describes some structural problems of IBM model 1. Below they are described with examples that have been found from the provided Hansard corpus.

1. (Many-to-One) Target words can only map to a single source word. A many-to-many mapping, which is close to the linguistic intuition that a whole phrase is the translation of another phrase, is not possible.

   An example of this problem can be the target French sentence:

   pour cette raison , nous ne avons pas cru bon de assister à la réunion et en avons informé le COJO en conséquence .

   that should be aligned to the source English sentence:

   in view of this , we deemed it inadvisable to attend the meeting and so informed COJO .

   Probably, "*pour cette raison*" should be aligned some way to four English words "*in view of this*" that is not possible in IBM-M1 and IBM-M2 models.

2. (Distortion) Every source and target position in the word have the same probability of aligning with each other.

   An example of this problem can be any French phrase "*Minister of Transport*" that should aligned to the English phrase "*ministre chargé de les transports*" in pairs of sentences:

   monsieur le Orateur , ma question se adresse à le ministre chargé de les transports .

   Mr. Speaker , my question is directed to the Minister of Transport .

3. (Fertility) The probability that a source word generates a target words does not change depending on the number of target words it already generates.

Next to these structural problems, there are also non-structural problems.

1. Firstly, rare source words align with too many target words, serving as "garbage collectors".

   monsieur le Orateur , comme je le ai signalé à le chef de le opposition , je espère faire une déclaration à ce sujet le1er novembre .

   vs

   NULL ({ 27 }) Mr. ({ 3 5 6 8 9 10 12 15 17 20 22 23 25 }) Speaker ({ }) , ({ }) as ({ }) I ({ }) indicated ({ 26 }) to ({ }) the ({ 1 }) Leader ({ 13 19 21 }) of ({ }) the ({ }) Opposition ({ }) , ({ }) I ({ }) would ({ }) hope ({ 2 7 11 14 18 }) to ({ }) make ({ }) an ({ }) announcement ({ }) on ({ }) this ({ }) question ({ 24 }) on ({ }) November ({ }) 1 ({ }) . ({ 4 16 })

   In this case rare source words "Mr." has become a "garbage collector".

2. Furthermore, the null source word gets assigned too few target words. This happens because there is only one null word, while other source words can occur more than once (The same example above).

4 IBM Model 1 improvements

Moore (2004) proposes three improvements to IBM model 1, to address the issues described in section 3.1.

4.1 Add-n smoothing

The translation probabilities of a source word e by a target word f are smoothed by *add-n smoothing*, to give more probability to rare words.

$$t(t|s) = \frac{C(t,s) + n}{C(s) + n \cdot |V|} \tag{7}$$

n is the smoothing parameter, $|V|$ is the estimated vocabulary size. We use the real vocabulary size (number of different words) and try different fractions of it using the multiplier **v**, because a number of the words stay unobserved.

In order to find the best parameters for add-n smoothing, AER scores for different values of n and V were evaluated on a smaller subset of the Hansard corpus, see Table 1. Possible value of **n** has been chosen as they were in Moore (2004) [1] with the expectation that they can provide best results.

The lowest AER value is achieved for n=0.001, $|V|$ = 0.1 * vocsize. These settings will be used in the combined improvements model.

4.2 More weight to null words

To align more words to the null word, the translation probabilities of the null word can be multiplied by a certain weight. Weights $w = 2, 3, 5, 10$ were evaluated on a smaller subset of the Hansard corpus, see Table 2. Although the differences are small $w = 10$ is the best score.

[1] Moore (2004), Table 1, Column "Add n"

v	n = 0.0005	n = 0.005	n = 0.008	n = 0.035
0.1	**0.826**	0.828	0.827	0.824
0.5	0.829	0.837	0.842	0.839
0.7	0.830	0.844	0.847	0.836
1.0	0.830	0.845	0.849	0.845

Table 1: AER scores for running IBM model 1 with add-n smoothing for different values of **n** and **v**. The model was evaluated on the first 1000 sentences of the Hansard corpus, concatenated with the test set. The algorithm was run for 20 iterations.

Weight	AER
2	0.814441130
3	0.812510967
5	0.812702869
10	**0.810229865**

Table 2: AER scores for running IBM model 1 with higher null weights, for different values of the null weight w. The model was evaluated on the first 5000 sentences of the Hansard corpus, concatenated with the test set. The algorithm was run for 40 iterations.

4.3 Heuristic initialization

Instead of using random initialization of translation probabilities, one could use more educated starting values. Heuristic initialization was implemented by running a pre-model, which supplies the actual model with initialized translation probabilities.

The pre-model computes a log-likelihood ratio for every combation of e_w and f_w. It sums four terms, corresponding to all combinations of e_w and f_w occurring (+) and not-occuring (-):

$$LLR(e_w, f_w) = \sum_{f_? \in f_+, f_-} \sum_{e_? \in e_+, e_-} C(f_?, e_?) log \frac{p(f_?|e_?)}{p(f_?)} \tag{8}$$

The LLR are then pseudo-normalized. Each LLR score is divided by the sum of LLR's for a single source word, the source word that has the highest sum of LLR's.

$$nLLR(e_w, f_w) = \frac{LLR(e_w, f_w)}{argmax_e \sum_f LLR(e, f_w)} \tag{9}$$

5 Experiments

Firstly, estimation of 3 IBM-M1 improvements parameters has been done on the small subset of the Hansard corpus of 1000 sentences concatenated with 447 sentences from the test set.

Secondly, experiments have been performed on the larger subset of the Hansard corpus of 5000 sentences concatenated with the test set for 40 iterations.

Several runs of the IBM-M1 have been performed to make sure that the IBM-M1 implementation converges into the same global optimum.

Experiments on the same data set have been done for Add-N smoothing to investigate if it makes the IBM-M1 results better, then the combined model based on Add-N smoothing and "Heavy" NULL weight has been built with the later combination with the heuristic initialization to show gradual improvements.

IBM-M2 has been initialized randomly, uniformly or by parameters estimated after the 3-rd iteration of the IBM-M1 to show its non-convex property.

Thirdly, the subset of the Hansard corpus of 25000 sentences concatenated with the test set has been chosen as more "real" task to be performed in 30 iterations on the main IBM Model configurations.

For evaluation three performance statistics, Recall, Precision and Alignment Error Rate (AER), have been used.

6 Results

For improvements comparison and evaluation the small data set of 5447 sentences has been used. Results of these experiments are reported in plots 1, 2, 3, 9, 5.

AER improvements have not been noticed by performed experiments for Add-N smoothing. Unfortunately, it's not exactly clear from the Moore (2004) how **n** is determined. There were attempts to do an "**empirical optimization**" as the paper advised, but probably it's need to try for smaller values of **n** to obtain some improvements in results. There were multiple tries to do Add-N smoothing for **n** as an integer > 0, but all these tries have provided worse results. It's expected that by using smoothing technique (Riley and Gildea, 2012) we can achieve better results.

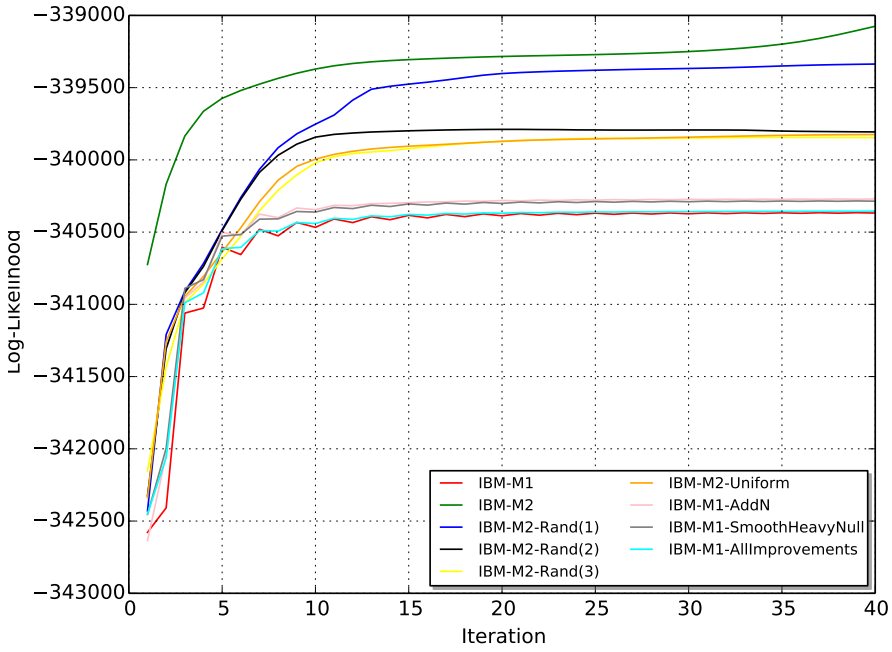

Figure 1: The evolution of **Log-likelihood** per iterations for experiments on the subset of the Hansard corpus of 5000 sentences concatenated with the test set of 447 sentences for 40 iterations.

The sensible improvement has been achieved in the experiment where Add-N smoothing and heavy NULL weights are combined together (IBM-M1-SmoothHeavyNull experiment). It shows that the second approach really works in practice.

By combining the model IBM-M1-SmoothHeavyNull with an heuristic initialization the model IBM-M1-AllImprovements has been obtained that provide relatively the same result comparing with the original IBM-M1 model, except the fact that it converges faster (as it's expected for the heuristic initialization improvement) [Fig 1, 2]. Significant performance increase is supposed not to appear by the failure of the Add-N smoothing experiments.

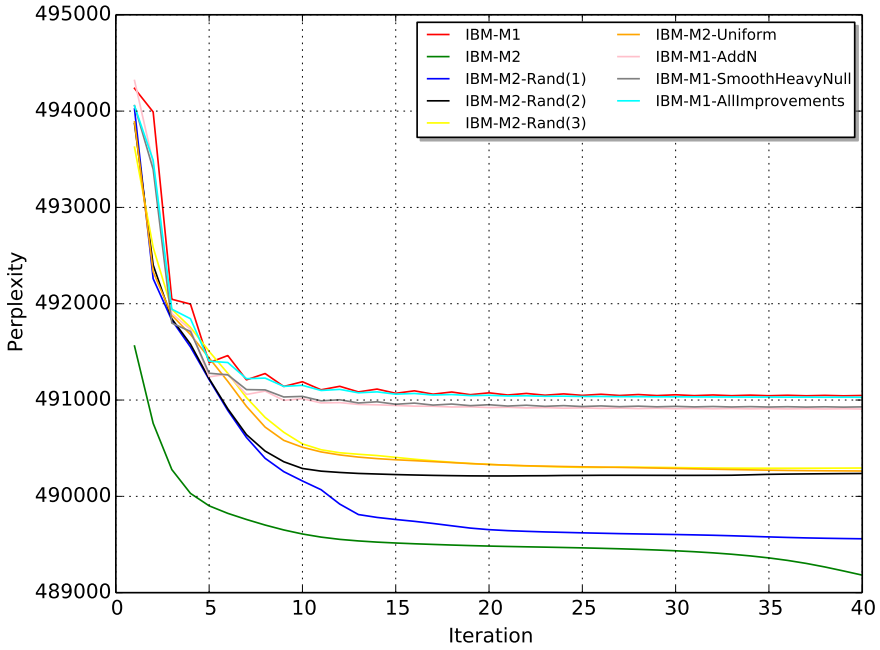

Figure 2: The evolution of **Perplexity** for the corpus of 5447 sentences.

Obtained results for the large corpus of 25447 sentences are reported in plots 6, 7, 8, 9, 10. Each EM experiment has converged [Fig 6, 7] that argues that the small fluctuation for the set of 5000 sentences reported previously is supposed to be the consequence of the small amount of data.

It can be seen that IBM Model 2 initialized by different random parameters converge to different local optimum. In comparison with IBM-M1, IBM-M2 produces more accurate result [Fig 8, 9, 10].

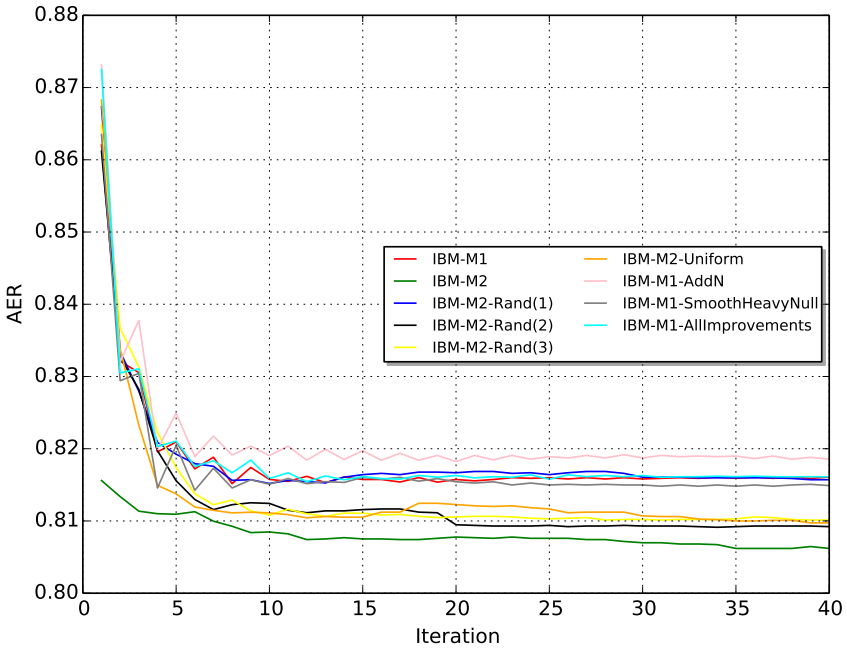

Figure 3: The evolution of **AER** for the corpus of 5447 sentences.

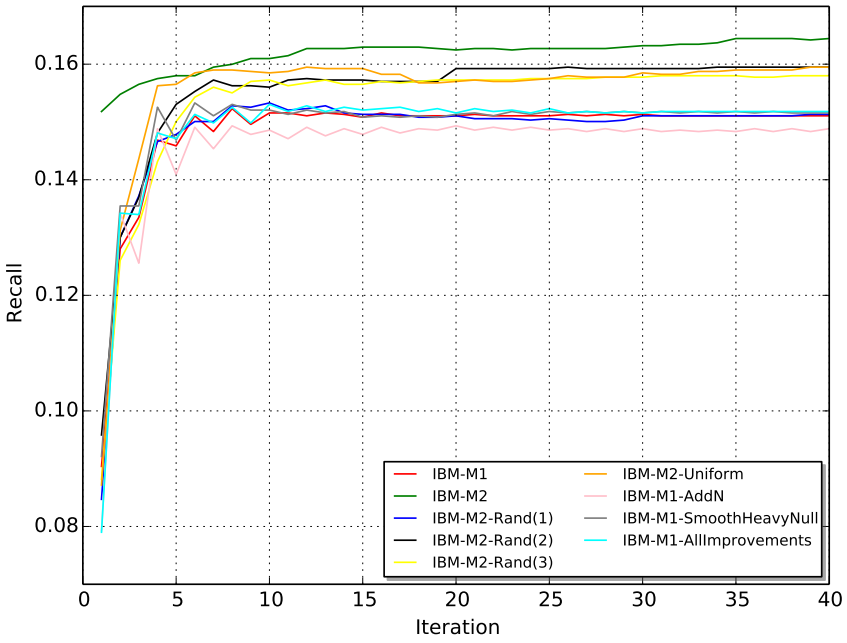

Figure 4: The evolution of **Recall** for the corpus of 5447 sentences.

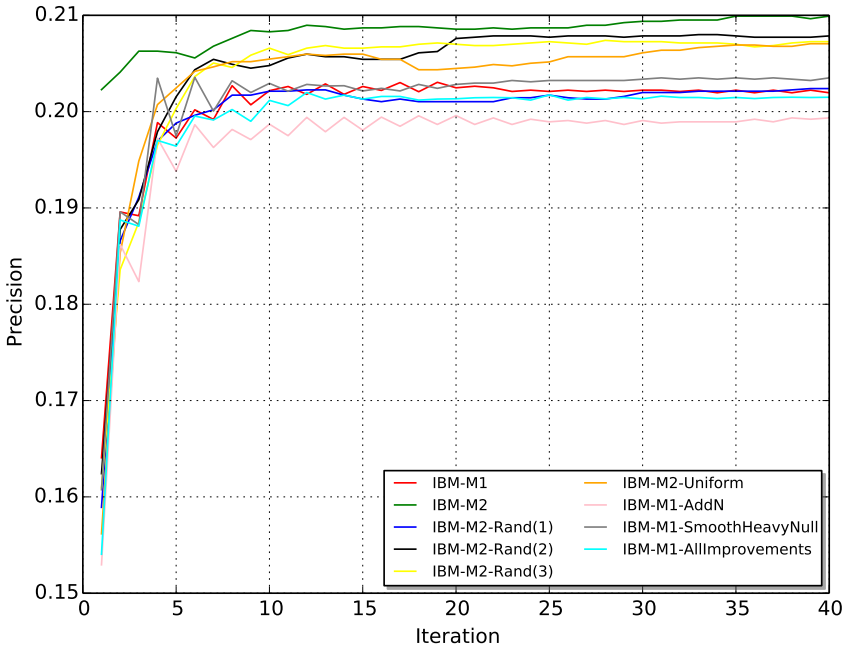

Figure 5: The evolution of **Precision** for the corpus of 5447 sentences.

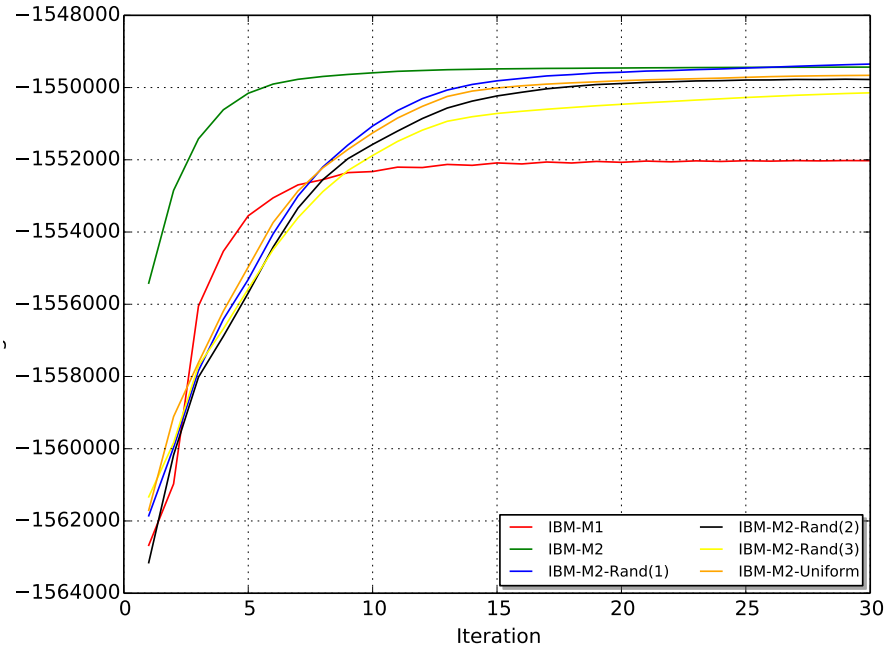

Figure 6: The evolution of **Log-likelihood** per iterations for experiments on the subset of the Hansard corpus of 25000 sentences concatenated with the test set of 447 sentences for 30 iterations. IBM-M2-Rand(i) are the IBM Model 2 initialized by different random parameters. IBM-M2-Uniform is the IBM Model 2 initialized uniformly. IBM-M2 is the IBM Model 2 initialized by estimated parameters from the 3-rd iteration of the IBM-M1 model.

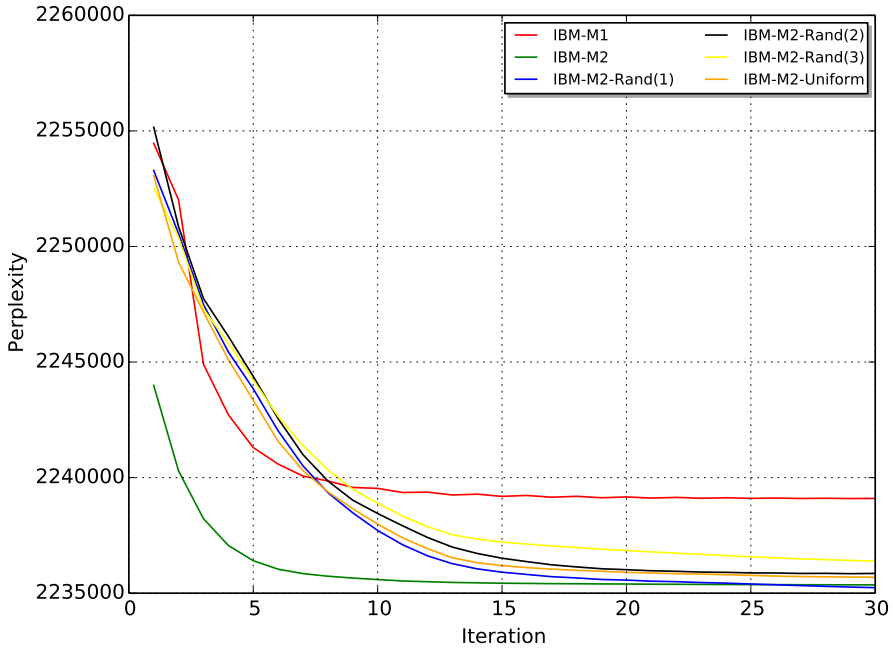

Figure 7: The evolution of **Perplexity** per iterations for experiments on the subset 25k sentences.

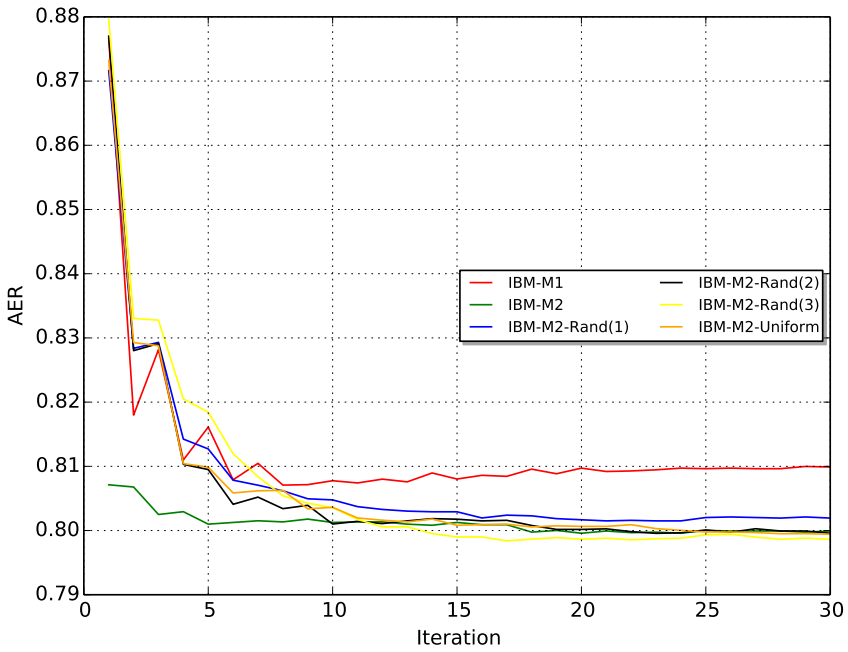

Figure 8: The evolution of **AER** per iterations for experiments on the subset 25k sentences.

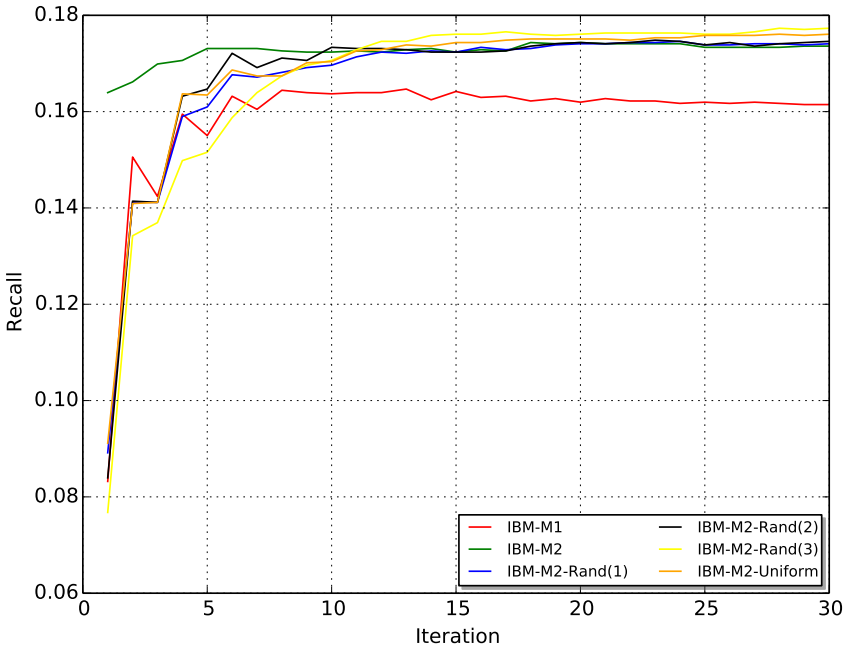

Figure 9: The evolution of **Recall** per iterations for experiments on the subset 25k sentences.

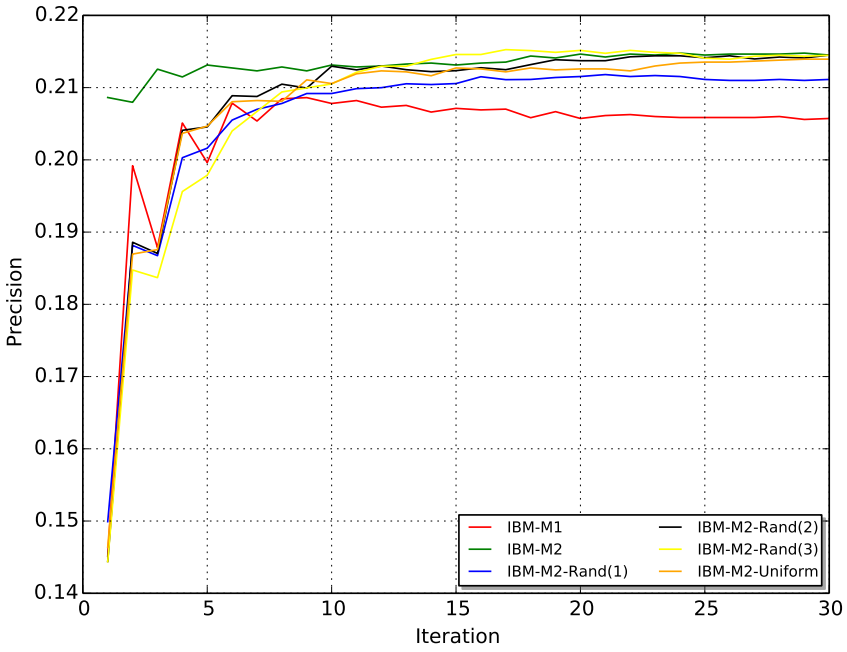

Figure 10: The evolution of **Precision** per iterations for experiments on the subset 25k sentences.

References

M. Collins. Statistical Machine Translation: IBM Models 1 and 2. 2011. URL http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/ibm12.pdf.

R. C. Moore. Improving ibm word-alignment model 1. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 518. Association for Computational Linguistics, 2004. URL http://research.microsoft.com/pubs/68958/model-one-final-rev.pdf.

D. Riley and D. Gildea. Improving the ibm alignment models using variation bayes. 2012. URL http://anthology.aclweb.org/P/P12/P12-2060.pdf.

Appendix I

Full source code is available in repositories https://github.com/pdekker12/nlp2 and https://github.com/Ignotus/nlp2 or in the attachment to this report.