Machine Translation
Assignment 2

Handed out: February 2nd, due: February 16th

IBM Model 1

Implement the EM algorithm for IBM Model 1 (see class six) in your favorite programming language. Test it on the corpora available on the Wiki `http://www.statmt.org/mtm2/?n=Main.SummerSchool` for a summer school we organised one year.

- a toy corpus
- a small segment of the Europarl corpus `http://www.statmt.org/europarl.tgz` (French-English or German-English)

Your program should output two different things:

- A table containing the word translation probabilities that were learned (note: think of an efficient data structure for such a sparse matrix)
- The most likely alignment for each sentence pair in the training data

Please return a report containing:

- 1 page description of your source code
- 1 page excerpt of the word translation table
- 1 page excerpt of Viterbi alignments
- your source code

Pseudo-code of IBM Model 1 as presented in the lecture:

Require: set of sentence pairs $(\mathbf{e}, \mathbf{f})$
Ensure: translation prob. $t(e|f)$
initialize $t(e|f)$ uniformly
while not converged **do**
  {initialize}
  count$(e|f) = 0$ **for all** e, f
  total$(f) = 0$ **for all** f
  for all sentence pairs (**e**,**f**) **do**
    {compute normalization}
    for all words e in **e** **do**
      s-total$(e) = 0$
      for all words f in **f** **do**
        s-total(e) += $t(e|f)$
      end for
    end for
    {collect counts}
    for all words e in **e** **do**
      for all words f in **f** **do**
        count$(e|f)$ += $\frac{t(e|f)}{\text{s-total}(e)}$
        total(f) += $\frac{t(e|f)}{\text{s-total}(e)}$
      end for
    end for
  end for
  {estimate probabilities}
  for all foreign words f **do**
    for all English words e **do**
      $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$
    end for
  end for
end while