Universität Stuttgart

Implementing IBM Model 1

Data and original pseudo-code from Philipp Koehn's SMT textbook.

(Posted here, with notes, by Alex Fraser)

Implement the EM algorithm for IBM Model 1 in your favorite programming language or a package like Matlab.

Start with the small toy set and then work on your choice of German/English or French/English:

Toy 6 short sentences
French--English: 1000 sentences, 292 short sentences
German--English: 1000 sentences, 292 short sentences

Your program should output two different things:

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

Pseudo-code of EM for IBM Model 1:

```
initialize t(e|f) uniformly
do until convergence
  set count(e|f) to 0 for all e,f
  set total(f) to 0 for all f
  for all sentence pairs (e_s,f_s)
    set total_s(e) = 0 for all e
    for all words e in e_s
      for all words f in f_s
        total_s(e) += t(e|f)
    for all words e in e_s
      for all words f in f_s
        count(e|f) += t(e|f) / total_s(e)
        total(f)   += t(e|f) / total_s(e)
  for all f
    for all e
      t(e|f) = count(e|f) / total(f)
```

The final implementation should include the NULL word as position 0 of f_s.

Final note: I have a graphical browser/editor available for word alignments, it comes with a small amount of gold standard data. The program is implemented in java (it is easy to test whether you have java: open a command shell and type "java", if it does something you are all set, otherwise you need to install from java.sun.com). If you are interested, please send me an email, get my email address from here.

Homepage of the Stuttgart SMT Reading Group