Statistical Machine Translation

Lab Exercise

5: IBM Model 1

Please use Java as your programming language for this lab
Refer to the lecture slides (Week 5, 6, 7) for extra information

1- Given two sentence pairs as follows:

Sentence ID	**Source**	**Target**
1	the house	la maison
2	house	maison

please **manually** calculate the **lexical translation probability** ***t***(e|f) of each word pair (e, f) within **two** iterations of the Expectation Maximisation (EM) algorithm (IBM 1) and show all the steps to arrive at these values:

t(la|the)

t(maison|the)

t(la|house)

t(maison|house)

Hint:

1, the tutorial in the slides (**Week7_Phrase-based SMT**, Page 6-28) can help you.

2**,** you can use either **the normal IBM model 1** or **the simplified IBM model 1**.

2- Using the two sentence pairs in Question 1 as input to implement the simplified IBM model 1 using Java. The **simplified IBM model 1** has the following steps:

- Initialise the **lexical translation probability** uniformly
- For each sentence pair (**e, f**), collect **counts** for word pair (e, f)

$$c(e|f;\boldsymbol{e},\boldsymbol{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e,e_j) \sum_{i=0}^{l_f} \delta(f,f_i)$$

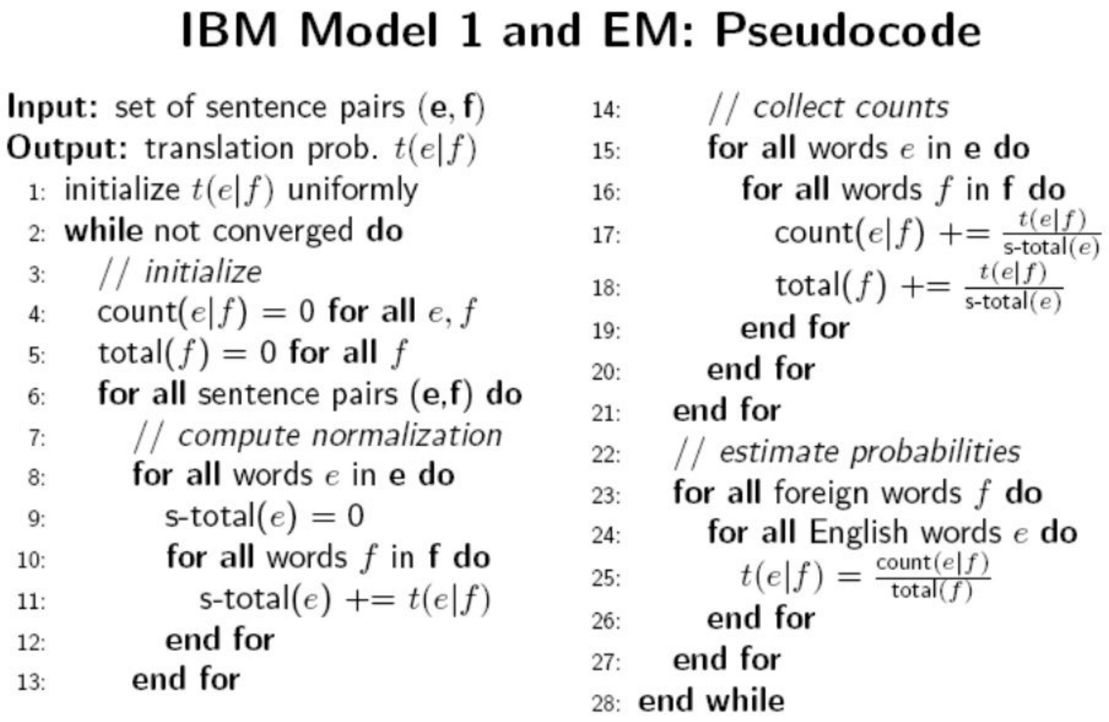
- Estimate **new lexical translation probabilities** on all sentences (corpus-level):

$$t(e|f;\boldsymbol{e},\boldsymbol{f}) = \frac{\sum_{(e,f)} c(e|f;\boldsymbol{e},\boldsymbol{f})}{\sum_e \sum_{(e,f)} c(e|f;\boldsymbol{e},\boldsymbol{f})}$$

- Iterate N times to stop.

Hint:

1, The pseudocode can help on implementation:



IBM Model 1 and EM: Pseudocode

```
Input: set of sentence pairs (e, f)
Output: 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
```

2, Example:

Input:

s1_src = “the house”

s1_tgt = “ la maison”

s2_src = “ house”

s2_tgt = “maison”

Iteration_number = 2

Output:

t(la|the) = 0.625

t(maison|the) = 0.375

t(la|house) = 0.172

t(maison|house) = 0.828

3- Following Question 1&2, please write a program to compute the lexical probabilities of any word pairs given a parallel corpus (train.en, train.de), where train.fr is the source data file and train.en is the target file. The output should be a file which contains word pairs with their translation probabilities.

Input:

train.en

train.de

Iteration_number = 2

Format of output file:

```
Mann small 0.0
groß man 0.262295081967
Mann a 0.262295081967
Haus small 0.267156339412
ist tall 0.213114754098
das my 0.0
groß is 0.0263964462687
klein my 0.204330927263
mein small 0.0580774650895
```