A: Theory

The algorithm used on this assignment was an enhanced version of the baseline algorithm. The enhancements are listed below:

- Considered both $p(f|e)$ and $p(e|f)$. The output is the intersection of both sets.
- Added NULL words to the end of each f and e sentences.
- The alignments considered are the nearest to the current index i. According to the last sentence on *pg. 837* of the HMM-Based Word Alignment in Statistical Translation paper (http://aclweb.org/anthology/C/C96/C96-2141.pdf), the difference in the position index is smaller than 3 (for Indoeuropean languages).
- Applied probability smoothing for both $p(f|e)$ and $p(e|f)$. Assuming we have a source vocabulary of size v_f and a target vocabulary of size v_e, the formula for calculating probabilities are smoothed as the followings:

$$P(f|e) = \frac{(count_{f,e} + \delta)}{(count_e + \delta * v_f)} \tag{1}$$

$$P(e|f) = \frac{(count_{e,f} + \delta)}{(count_f + \delta * v_e)} \tag{2}$$

Here, δ is a very small constant. We chose it to be 0.01

B: Algorithm

The algorithms for the both the word alignment and decoding are shown in the next page:

Algorithm 1 Espectation Maximization Training

Input: Set of sentences F from source language and Set of sentences E from target language
Output: Probability distributions $P(f|e)$ and $P(e|f)$ for any source word e and target word f

add a last word to represent $NULL$ on each f and e sentences.
$v_f \leftarrow$ number of distinct source words and $v_e \leftarrow$ number of distinct target words
Initialize all counts to 0
Initialize $t1_{f|e} \leftarrow \frac{1}{v_f}$ and $t2_{e|f} \leftarrow \frac{1}{v_e}$ for all possible pair (f, e)
Initialize $\delta \leftarrow 0.01$
while $Epoch \leq MaxEpoch$ **do**
    $Epoch \leftarrow Epoch + 1$
    for each pair of sentences (f, e) **do**
        for each f_i in f **do**
            $c \leftarrow 0$ and $z \leftarrow 0$
            for each e_j in e **do**
                $z \leftarrow z + t1_{f_i|e_j}$
            for each e_j in e **do**
                $count_{f_i,e_j} \leftarrow count_{f_i,e_j} + \frac{t1_{f_i|e_j}}{z}$ and $count_{e_j} \leftarrow count_{e_j} + \frac{t1_{f_i|e_j}}{z}$
        for each e_i in e **do**
            $c \leftarrow 0$ and $z \leftarrow 0$
            for each f_j in f **do**
                $z \leftarrow z + t2_{e_i|f_j}$
            for each f_j in f **do**
                $count_{e_i,f_j} \leftarrow count_{e_i,f_j} + \frac{t2_{e_i|f_j}}{z}$ and $count_{f_j} \leftarrow count_{f_j} + \frac{t2_{e_i|f_j}}{z}$
    for each pair (f, e) in $count_{f,e}$ **do**
        $t1_{f|e} \leftarrow \frac{(count_{f,e}+\delta)}{(count_e+\delta * v_f)}$
    for each pair (e, f) in $count_{e,f}$ **do**
        $t2_{e|f} \leftarrow \frac{(count_{e,f}+\delta)}{(count_f+\delta * v_e)}$
return $t1_{f|e}$ and $t2_{e|f}$ for all e and f

Algorithm 2 Decoding

Input: Probability distributions $t1_{f|e}$ and $t2_{e|f}$ for any source word e and target word f
Output: Set of aligned word pairs (f_i, e_j)

$AlignedWords \leftarrow \emptyset$
for each pair of sentences (f, e) **do**
    Ignore possible $NULL$ alignments
    $list_{FE} \leftarrow \emptyset$ and $list_{EF} \leftarrow \emptyset$
    for each f_i in f **do**
        $best_j \leftarrow argmax_{e_j} t1_{f_i|e_j}$ // *if* $p(j) = p(best_j)$ *the nearest distance* $d(f_a j, e_j)$ *is chosen*
        add pair (f_i, e_j) to $list_{FE}$
    for each e_i in e **do**
        $best_j \leftarrow argmax_{f_j} t2_{e_i|f_j}$ // *if* $p(j) = p(best_j)$ *the nearest distance* $d(e_a j, f_j)$ *is chosen*
        add pair (f_j, e_i) to $list_{EF}$
    add $list_{FE} \cup list_{EF}$ to $AlignedWords$
return $AlignedWords$