Natural Language Processing HW5

Hong Pengfei 1002949

August 2020

1 P1

1.1 (a)

(2)$\rightarrow$ (4) $\rightarrow$ (7) $\rightarrow$ (9) $\rightarrow$ (6)

1.2 (b)

$score_{DM} = \sum_{k=1}^{l-1} \eta \mid pl(p_k){+}1{-}pf(p_{k+1}) \mid \; = \eta{\times}0{+}\eta{\times}|3{+}1{-}5|{+}\eta{\times}|5{+}1{-}6|{+}\eta{\times}|6{+}1{-}4| = \eta{\times}4$

1.3 (c)

$$score_{LM} = \sum_{k=1}^{l-1} log(q(e))$$

$= log(q(Fairprice|*)) + log(q(is|Fairprice)) + log(q(the|is))$

$+log(q(biggest|the)){+}log(q(supermarket|biggest)){+}log(q(in|supermarket)){+}log(q(Singapore|in))$

1.4 (d)

Possible sequences: (2) $\rightarrow$ (4) $\rightarrow$(7) $\rightarrow$ (9) $\rightarrow$ (6); (3) $\rightarrow$ (8) $\rightarrow$ (9) $\rightarrow$ (6)

$score_{LM}$ for all sequences are the same.

For sequence (2) $\rightarrow$ (4) $\rightarrow$ (7) $\rightarrow$ (9) $\rightarrow$ (6):

$$score_{DM} = (-8) \times 4 = -32$$

$$score_{TM} = -0.5 + 2 + 2 + 1 + 0.5 = 5$$

$$score = score_{DM} + score_{TM} = -32 + 5 = 27$$

For sequence (3) → (8) → (9) → (6):

$$score_{DM} = (-8) \times (1 + 0 + 3) = -32$$

$$score_{TM} = -1 + 1.5 + 1 + 0.5 = 2$$

$$score = score_{DM} + score_{TM} = -32 + 2 = -30$$

Optimal sequence is: (2)→ (4) →(7) →(9) →(6)

1.5 (e)

intermediate score minus by 1 after each step when the transition rule is used.

2 P2

2.1 a

Shown in figure 1 are the two cases that ITG cannot handle, which corresponds to BDAC and CADB in our case.

2.2 b

ABC, ACB, BAC, BCA, CAB, CBA

2.3 c

We can have 24 alignments in total, out of which 2 are invalid, 10 are repeatitive. So in total we have 12 possible unique translations:
ABCA, ABAC, ACBA, AABC, ACAB, AACB,
BACA, BAAC, CABA, AABC (repeat), CAAB (invalid), AACB (repeat),
BCAA, BAAC (invalid), CBAA, ABAC (repeat), CAAB, ACAB (repeat),
BCAA (repeat), BACA (repeat), CBAA (repeat), ABCA (repeat), CABA (repeat), ACBA (repeat).

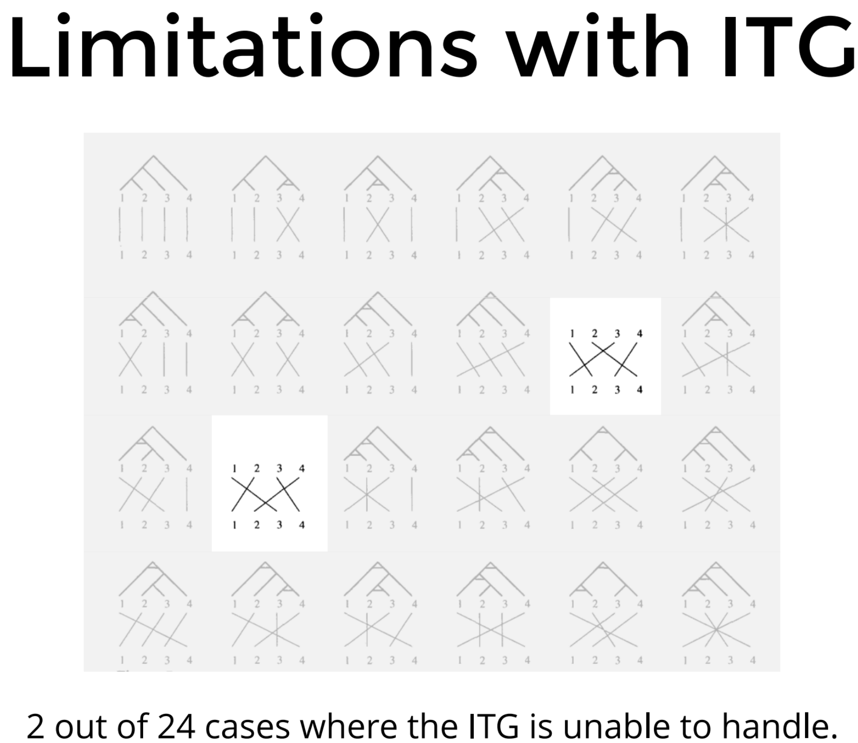

Figure 1: Limitation of ITG

2.4 d

No, no matter what alignment is invalid, it can always be mirrored into another direction to have another invalid alignment. So the number of invalid alignment will be even.

3 P3

3.1 a

Steps dont need change:

- Initialization: we can still randomly initialize transition probabilities.
- we can follow same steps for Expectation.

We can modify the step for Maximization step to the following.

1. Take assignments from E step Weight the counts by reliability score.
2. The weighted counted are used in the computation of transition probability:

$$count(e, f) = \sum_{f_i=f, e_j=e, k} count^{(k)}(i, j, k)$$

$$count(e) = \sum_{k, i, e_j=e} count^{(k)}(i, j, k)$$

here $count^{(k)}(i, j, k)$ is defined as:

$$count^{(k)}(i, j, k) = count(i, j)$$

in unreliable resources for instance k and

$$count^{(k)}(i, j, k) = count(i, j) * k$$

for reliable sources for instance k Use these weighted counts to do parameter estimation for model:

$$t(f|e) = \frac{count(e, f)}{count(e)}$$

3.2 b

From ideas in a, we can include the Chinese-English Dictionary into the word alignment learning process by assign a high weight for their reliability score in the instance.

$$count^{(k)}(i, j, k) = count(i, j) * high\_score$$

if w_i and w_j alignment is in the prior dictionary.
similarly,

$$count^{(k)}(i, j, k) = count(i, j) * low\_score$$

if w_i and w_j alignment is not in the prior dictionary.

3.3 c

Initialization method can be based on IBM Model 1. In IBM model 2:

- the $t(f|e)$ parameters estimated under IBM Model 1
- random values or use uniform distribution for initialization for the $q(j|i,l,m)$ parameters.

$q(j|i,l,m)$ is defined as the conditional probability of French word f_i being aligned to English word e_j , given the French length m and the English length l.

4 P4

4.1 a

To prove this, we need show that $W \cdot [e_j; d_i]^T$ is equivalent to $W_1 \cdot e_j + W_2 \cdot d_i$. we can rewrite W as $[W_1; W_2]$

$$W \cdot [e_j; d_i]^T = [W_1; W_2] \cdot [e_j; d_i]^T = W_1 \cdot e_j + W_2 \cdot d_i$$

4.2 b

we assume:

$$z = max(\alpha_i)$$

then

$$a_i = \frac{e^{\alpha_i - z}}{\sum_T^j e^{\alpha_j - z}} = \frac{e^{\alpha_i} * e^{-z}}{\sum_T^j e^{\alpha_j} * e^{-z}} = \frac{e^{\alpha_i}}{\sum_T^j e^{\alpha_j}}$$

Overflow problem By subtracting $z = max(\alpha_i)$, we can make sure $\forall \alpha_i, \alpha_i - z < 0$, therefore $\sum_T^j e^{\alpha_j - z}$ is bounded by j as upper bound and $e^{\alpha_i - z}$ is upper bounded by 1 and will not reach infinity.

Underflow problem By subtracting $z = max(\alpha_i)$, $\sum_T^j e^{\alpha_j - z}$ is lower bounded by 1 because for the maximum α_j it will be 1.