Tagging

Most frequent tag baseline

b

F1 score on development set: **0.8**
(Entity level P/R/F1: 0.78/0.81/0.80)

HMM tagger

c Forward-Backward

$P(y = LOC \mid j = 7 \mid |x| = 15) = 0.022629134674979002$
(Computed using forward.py)

3.6.c in the next page

e Theoretical question

Sentences:

- Eat now (VB, RB)
- I fish fish (PR, VB, NN)

Test sentence:

- Fish now (VB, RB)

Using the following q and e:

$$q(v \mid w, u) = \frac{c(w, u, v)}{c(w, u)}, \quad e(w \mid t) = \frac{c(w\, is\, tagged\, t)}{c(t)}$$

Choosing $t_1 = NN$:

- t=NN:

$$e(fish \mid NN) \cdot q(NN \mid *, *) = \frac{c(fish \mid NN)}{c(NN)} \cdot \frac{c(*, *, NN)}{c(*, *)} = \frac{1}{1} \cdot \frac{1}{2} = \frac{1}{2}$$

- t=VB:

$$e(fish \mid VB) \cdot q(VB \mid *, *) = \frac{c(fish \mid VB)}{c(VB)} \cdot \frac{c(*, *, VB)}{c(*, *)} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

- for the other options $c(*, *, t) = 0$

Now for t_2:
$q(PR \mid *, NN) = 0$ and for any $t \neq RB$: $e(now \mid t) = 0$ so the greedy algorithm will choose some t_2 at random.
The probability for any choice is zero while there is some other sequence of tags with non zero probability:

$$P(fish\, now\, , VB, RB) = e(VB \mid fish) \cdot q(VB \mid *, *) \cdot e(RB \mid now) \cdot q(RB \mid *, VB) = \\ = \frac{1}{4} \cdot \frac{1}{2} \cdot 1 \cdot 1 = \frac{1}{8}$$

3. b+c. We used lambda1 = 0.3, lambda2 = 0.5 to get f1 score = 0.75 (maximal).

We perform to things in order to shorten the runtime, first, for (x_k,v) that were unseen in the training (meaning e(x_k,v) = 0) we didn't perform any calculation. Also, for each word we only considered tags that we learnt for it on the training.

4. d. Greedy F1 = 0.88, Viterbi F1 = 0.89

E. some examples for common mistakes:

MILWAUKEE - pred: ORG real: LOC

Washington - pred: LOC real: PER

U.S. - pred: LOC real: MISC

Texas - pred: ORG real: MISC

Tour - pred: MISC real: O

Test - pred: O real: ORG

Entity - pred: PER real: LOC

Stadium - pred: LOC real: O

Florida - pred: LOC real: ORG

We see many mistakes of LOC/ORG/PER since many of them usually appear in a phrase. Some of them also ambiguous, for example **Washington** is a LOC but also a person, **Florida** probably appears more times as a location name then as an organization name (or part of it). On the other hand, **Texas** and **Milwaukee** probably appeared more as part of organization name. **We see that the model makes more mistakes on ambiguous words.**

BiLSTM tagger

a)

i) If we didn't use masking, the computation of the gradient and the loss function would be affected by padded words (which should not impact learning), and consequently we would get unwanted results. We use masking to nullify these terms so they will not impact the loss function.