Natural Language Processing

Problem Set 3: Analytical

Linan Qiu (lq2137)

November 10, 2014

1.

1(a).

- **Input:** English string e, integer m
- **Algorithm:**

$$\begin{aligned} p(f, a|e, m) &= \prod_{j=1}^{m} t(f_j|e_{a_j}) q(a_j|j, l, m) \\ &= p(a|e, m) p(f|a, e, m) \end{aligned}$$

First, find a^* where

$$a^* = \underset{a \in \{\text{all possible alignments of length } m\}}{\arg\max} p(a|e, m)$$

All possible alignments can be found, though there are $(1 + l)^m$ possible a^*s.

Then find f^* where

$$f^* = \underset{f \in \{\text{all possible French sequences of length m}\}}{\arg\max} a^* * p(f|a, e, m)$$

- **Output:** f^* and a^*

1(b).

- **Input:** English string e, integer m
- **Algorithm:**

  We expand the IBM model to reduce run time from $O((l+1)^m)$.

$$\begin{aligned} p(f|e,m) &= \sum_{a:|a|=m} \prod_{j=1}^{m} t(f_j|e_{a_j}) q(a_j|j,l,m) \\ &= \sum_{a_1=0}^{l} \sum_{a_2=0}^{l} ... \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) q(a_j|j,l,m) \\ &= \left(\sum_{a_1=0}^{l} t(f_1|e_{a_1}) q(a_1|1,l,m) \right) ... \left(\sum_{a_m=0}^{l} t(f_m|e_{a_m}) q(a_m|m,l,m \right) \end{aligned}$$

  This expression cuts runtime to $O((l+1) * m)$.

  Next we iterate through all possible French sentences

$$f^* \in \{\text{all possible French sequences of length m}\}$$

- **Output:**

$$\underset{f^* \in \{f \text{ and } |f^*|=m\}}{\arg\max} \quad p(f|e,m)$$

1(c).

The model $\arg\max_e p(f|e) P_{LM}(e)$ comprises of two portions

- $p(f|e)$ which represents how likely a French sequence is given the English sequence
- $P_{LM}(e)$ which represents the linguistic correctness of the English sequence

Hence, both the likelihood of translation and the linguistic correctness of the resulting sentence are considered.

The model $\arg\max_e p(e|f)$ only considers the likelihood of e as a translation for f without checking for the linguistic (grammatical or otherwise) correctness of the resulting English sentence.

Without considering the linguistic correctness of the resulting English sentence, we may have worse results since we only considered the translation and not the correctness of the output sentence.

2.

- **Input:** English string e of length l, French string f of length m
- **Algorithm:**
  - **Initialize:**

$$a_0 = 0$$
$$\pi(0,0) = 1$$

  - For $j = 1...m$, for $b = 0...l$

$$\pi(j,b) = \max_{c \in \{0...l\}} [\pi(j-1,c) * t(f_j|e_b) * q(b|c,j,l,m)]$$
$$bp(j,b) = \arg\max_{c \in \{0...l\}} [\pi(j-1,c) * t(f_j|e_b) * q(b|c,j,l,m)]$$

  - Set

$$a_m = \arg\max_{c \in \{0...l\}} bp(m,c)$$

  - For $k = (m-1)...1$,

$$a_k = bp(k, a_k + 1)$$

- **Output:** Sequence $a = a_0...a_k$

3.

Since distortion limit is 0, we do not worry about rearranging phrases. This problem then becomes one of choosing split points. Hence, we should use a modified CKY algorithm.

- **Input:** Sentence $x = x_1 x_2 ... x_N$
- **Initialization:** for $i = 1...N$

$$\pi(ii) = g(p_i)$$

  where p_i is the single word phrase (i, i, x_i). Hence, we start with the scores of every word as a phrase and improve on it dynamically, building upwards.

- **Algorithm:**
  - For $l = 1...(N-1)$
    - for $i = 1...(N-l)$

* $j = i + l$

$$\text{splitMax} = \max_{s \in \{i \ldots (j-1)\}} (\pi(i, s) + \pi(s + 1, j))$$
$$\pi(i, j) = \max(\text{splitMax}, g(w_i ... w_j))$$

Here, we decide if we should split a sequence into two or more phrases or simply consider the entire sequence as one phrase. splitMax shows the maximum score for splitting the sequence in a certain way, while $g(w_i ... w_j)$ is the score for keeping it as a single phrase.

- **Output:** Return $\pi(1, N)$