Engineering Tripos Part IIB FOURTH YEAR

MODULE 4F11: SPEECH AND LANGUAGE PROCESSING

Solutions to Examples Paper 2

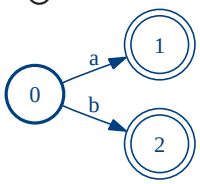
1. Discuss how *alignment* is used to indicate *translation equivalence*. Explain the hierarchical nature of translation equivalence.

   Parallel text collections consist of translations in two (or more) languages. In certain tasks, e.g. translation model parameter estimation or the evaluation of machine translation systems, it useful to know which portions of the collections are mutual translations, and which are not. An alignment is an annotation of the parallel text collection which provides this information. An alignment is typically done in two steps. (1) A scheme is defined which identifies regions of text in each language. This might be done by introducing boundary markers (start markers and end markers) into the text. For example, (s_1, e_1) could be pointers into the text which might indicate start and end positions in one language, while (s_2, e_2) might indicate similar information in the other language text. (2) Individual regions of text are linked across languages, with the implication that the linked regions are translation equivalent (see Lecture 11, handout page 19 for an example). Continuing the example in which regions are set off by boundary markers, an alignment may consist of a set of four-tuples (s_1, s_2, e_1, e_2).

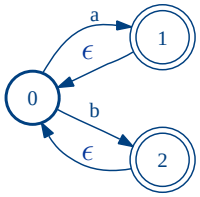
   Text collections often have a hierarchical structure. A news archive may contain collections of news articles, which themselves have a structure, such as headlines followed by paragraphs, which are further refined to sentences, phrases, and words. Translation equivalence can be defined to reflect the structure of the text collections. For example, parallel text collections can be aligned at the document level. In the scheme discussed above, the start and end markers would only be allowed to be placed at document boundaries. Once alignment is established at the document level, sentences within documents can be aligned. After this, words and phrases can be aligned between sentence translations. Note that the sense of 'translation equivalence' changes to reflect the scope of the alignment task. Asserting that 'the green man' is a translation of 'el hombre verde' is very different from judging whether a book in one language is a translation of a book in another language. The first case is typically much more literal and limited in scope.

2. The *closure* operators indicate repetition, e.g. $a^* = \{\epsilon, a, aa, aaa, \ldots\}$ and $a^+ = \{a, aa, aaa, \ldots\}$. Draw acceptors for the following:

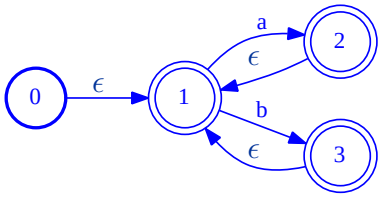
(a) $a \bigcup b$



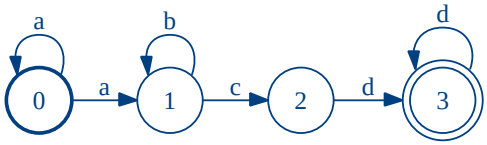
(b) $(a \bigcup b)^+$



(c) $(a \bigcup b)^*$



(d) $a^+b^*cd^+$

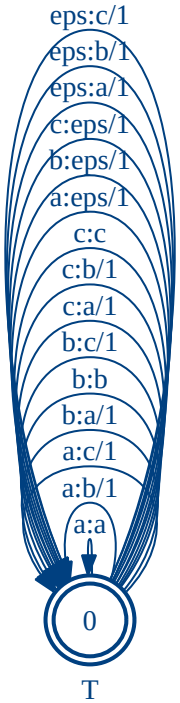


3. For the vocabulary $V = \{a, b, c\}$, a per-symbol loss function is defined as following for $x, y \in V$

$$d(x, y) = \begin{cases} 0 & x == y \\ 1 & x \neq y \end{cases}$$

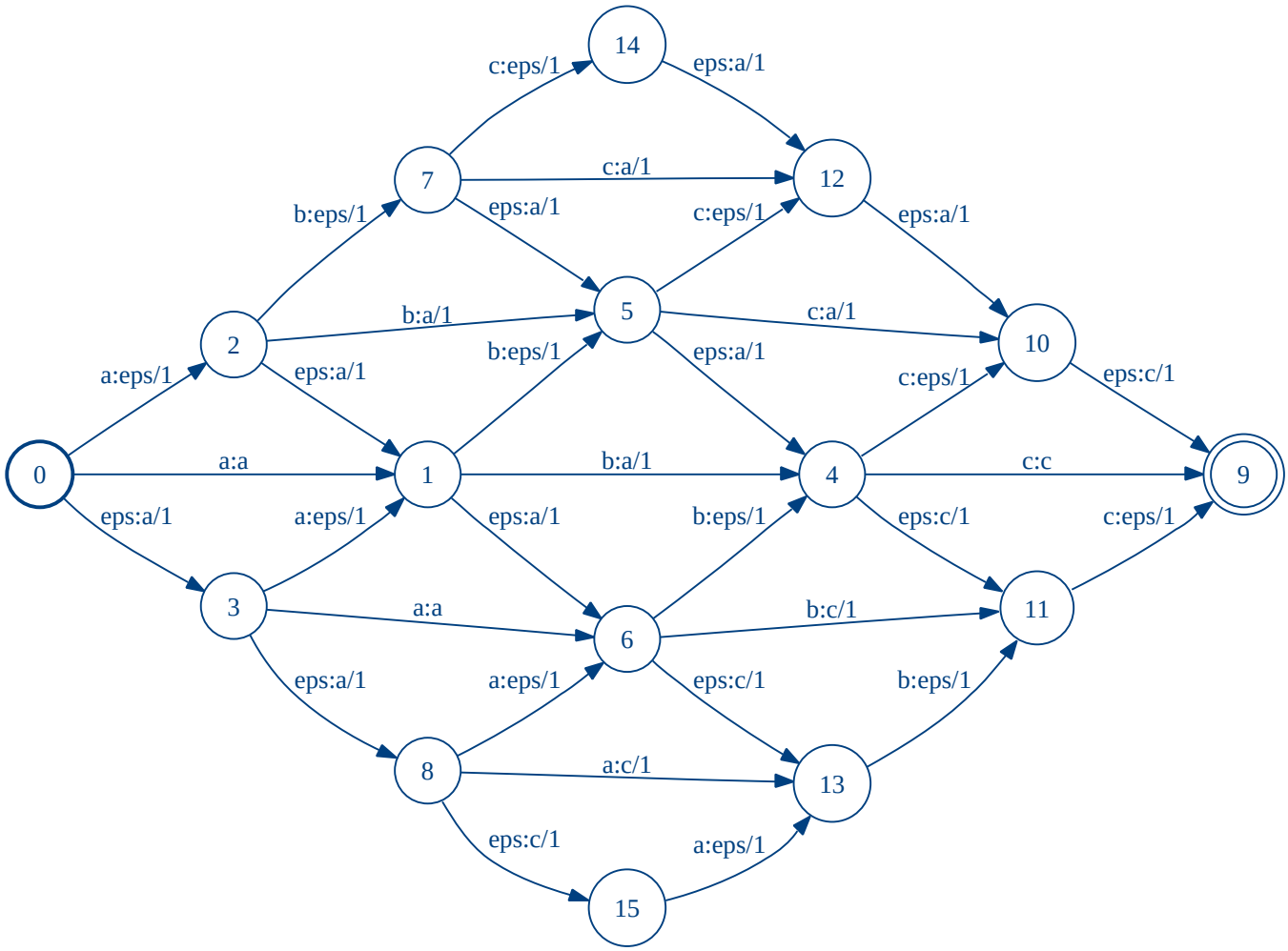
and $d(x, \epsilon) = d(\epsilon, x) = 1$. Draw a transducer which implements this distance and explain how this transducer can be used to compute the *edit distance* between two strings from V^*.

A transducer which maps symbols-to-symbols with the correct edit cost is:

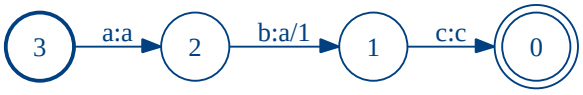


T

Suppose we consider an acceptor L for the string 'a b c' and another acceptor R for the string 'a a c'. These strings differ by one substitution operation, but they could also differ by *many* sequences of insertion and replacement operations. The transducer which contains *all* alignments of the two strings is found as $L \circ T \circ R$ is :



However, the *shortest path* through this transducer is



which gives the unique, best cost.

4. Suppose the Model-1 word translation probability distribution is to be estimated from the following two sentence pairs:

$E^{(1)} = a\ r\ p\ q\ r \quad F^{(1)} = \epsilon\ \gamma\ \rho\ \epsilon$
$E^{(2)} = o\ m\ r\ r\ a \quad F^{(2)} = \beta\ \alpha\ \epsilon\ \theta$

(a) Explain why the maximum likelihood estimate of $p_T(\gamma|m)$ is zero.

The Model-1 parameter estimation procedure only assigns probability to word translations associated with a valid word-to-word alignment. Since γ and m never occur in sentence translation pairs, they can't possibly be aligned, and thefore their word-to-word translation probability must be zero.

(b) Describe a parallel text alignment application which benefits from such harsh models.

If the goal is to perform alignment with very high precision, and to reject doubtful alignments, such probabilities can be very useful. For example, if a very large parallel text collection was available, and m and γ had been observed in the collection, but never in the same sentence pairs, a cautious approach to aligning new documents or sentences would be to discard potential alignments which would contain these two words. Such models can have good discriminative power, but poor generalization.

(c) Describe a parallel text alignment application for which models with word translation probability values of zero would be inappropriate.

If the goal is to generate word alignments for two sentences, as opposed to deciding whether or not two sentences are translations, such distributions can lead to zero probability assigned to alignments; this data would then not be used in building the translation system. In this task, where there is a need to use all available data, generalization is important.

(d) Suggest a modeling strategy which avoids assigning zero probability to word translation probabilities.

One possibility is use a discounting and back-off strategy, similar to what is used in bigram language modeling. For example

$$p_T(\gamma|m) = \begin{cases} p(\gamma, m) & \text{if } f(m,\gamma) > C \\ \alpha(m) p_u(\gamma) & \text{else} \end{cases}$$

where $p_u(\gamma)$ is estimated over the parallel text or (more easily) set to a uniform distribution; the joint distribution for the seen word pairs would be discounted.

5. Give formulae which describe (a) BLEU and (b) Alignment Error.

Describe the resources needed for the calculation of both of these quantities. Explain their role in the development of statistical machine translation systems.

(a) The goal is to calculate the BLEU score of a set of automatic translations $\{E^i\}_{i=1}^R$ against a set of reference translations, e.g. $\{E^i_{(1)}, E^i_{(2)}, E^i_{(3)}, E^i_{(4)}\}_{i=1}^R$.

- Set N to be the order of the highest n-gram to be considered, e.g. N=5
- For each sentence i, and for $n = 1, \ldots, N$, gather the following n-gram counts:
  - c_n^i : the number of hypothesized n-grams

- $\bar{c}_n^i$: the number of correct n-grams, where the contribution of each distinct n-gram is *clipped* to the maximum number of occurrences in any one reference

- Compute the precision for each n-gram , $n = 1, \ldots, N \quad : \quad p_n = \left(\sum_i \bar{c}_n^i\right) / \left(\sum_i c_n^i\right)$
- Calculate the Brevity Penalty
  - Compute the shortest reference length : $r = \sum_i \min\{|E_{(1)}^i|, |E_{(2)}^i|, |E_{(3)}^i|, |E_{(4)}^i|\}$
  - Compute the hypothesis length : $c = \sum_i |E^i|$

$$BP = \begin{cases} 1 & c > r \\ \exp\left(1 - \frac{r}{c}\right) & c <= r \end{cases}$$

- The BLEU score is

$$\text{BLEU} = BP * \exp\{\sum_{n=1}^{N} \log \frac{p_n}{N}\}$$

(b) Alignment Error measures the number of non-NULL word alignments by which the automatic word alignment differs from the reference word alignment. Suppose there are two sets of alignments:
B : automatic word alignments $\leftarrow$ *produced by an alignment model*
B': reference word alignments $\leftarrow$ *created by humans*
AE is calculated as follows:
Step 1. Remove the NULL word links from B' and B to form $\bar{B}'$ and $\bar{B}$
Step 2. Compute $AE(B, B')$:

$$AE(B, B') = \frac{|\bar{B}| + |\bar{B}'| - 2\,|\bar{B} \cap \bar{B}'|}{|\bar{B}'| + |\bar{B}|}$$

Both AE and BLEU require human annotation of held out reference sets. AE requires reference alignments, in which known sentence-level translations are word-aligned by bilingual annotators. As alignment model parameter estimation proceeds, AE can be calculated to estimate the quality of automatic alignments generated by the models. BLEU requires a set of reference translations. After alignments are produced, and while the translation system is being built, BLEU can be used to measure the translation quality and determine the value of any refinements to the system.

6. Suppose a pair of sentences f_1^J and e_1^I are known to be translations.

   (a) Write the formula for the overall translation probability under Model-1.

   After introducing the alignment process a_1^J, the Model-1 probability is

$$P(f_1^J, a_1^J, J|e_0^I) = \frac{1}{(I+1)^J}\, p_L(J|I) \prod_{j=1}^{J} p_T(f_j|e_{a_j})$$

(b) Describe in detail how the Model-1 (i) Viterbi likelihood and (ii) marginal likelihood can be computed for this pair of sentences using Weighted Finite State transducers.

Suppose a WFSA F is constructed for the sequence f_1^J and another WFSA E is constructed for the sequence e_1^I.

(i) To find the Viterbi likelihood, construct a WFST T as follows:
1. For all words f in f_1^J and all words e in e_1^I, add an arc

$$0 \xrightarrow{f:e/-\log p_T(f|e)} 0$$

The reordering probability, which is constant under Model-1 for all alignments of a given length sequence, can be ignored.

2. Perform the composition $A = F \circ T \circ E$ under the tropical semiring. This will give the likelihood $-\log P(f_1^J, a_1^J = [1, 2, \ldots, I] \,|\, e_1^I)$; that is, the alignment of e_1^I and f_1^J without any reordering. Note that this may fail (i.e. give probability 0) if some of the word-to-word translations have zero probability.

As an alternative approach, the acceptor E can be a *permutation acceptor* for e_1^I. An acceptor of this form contains all possible reorderings of the sequeuence $e_1 \ldots e_I$. Since all reorderings have the same probability under Model-1, the permutation acceptor in this instance can be unweighted, and the alignment likelihood . Note that this approach is rarely taken: there are techniques to build permutation acceptors, but such machines are not practical.

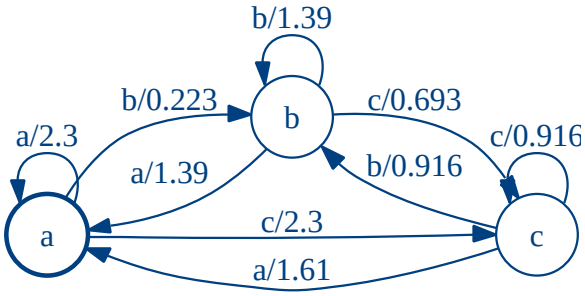
7. A Markov process X_t takes values in $\{a, b, c\}$. The transition probability associated with the process is

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
a	b	0.8
a	otherwise	0.1
b	c	0.5
b	otherwise	0.25
c	a	0.2
c	otherwise	0.4

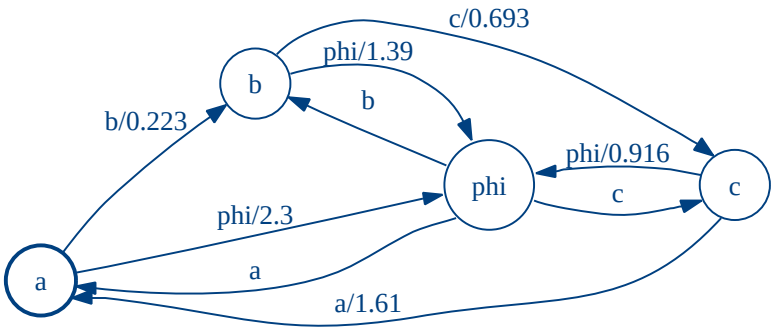
(a) A weighted acceptor is required which assigns weight to sequences consistent with the Markov transition probabilities. Draw this acceptor in such a way that all transition probabilities appear explicitly. Assume likelihood is to be computed with operations in the tropical semiring.

Computation under the tropical semiring requires replacing probabilities by their negative log likelihood value, as in this table

X_{t-1}	X_t	$-\log P(X_t \mid X_{t-1})$
a	b	0.223
a	otherwise	2.30
b	c	0.693
b	otherwise	1.39
c	a	1.61
c	otherwise	0.916



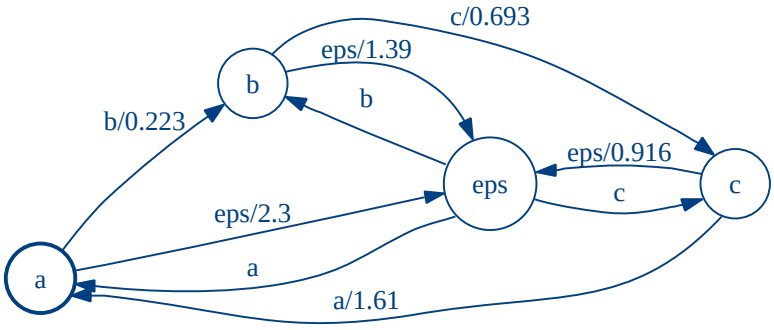
(b) Draw an equivalent machine using *failure transitions*. Explain how this leads to a simpler machine than that of part (a).



Note that taking a failure arc consumes an input symbol *after* reaching the failure state **(phi)**. For example, reading a symbol a or the symbol c while in state **(a)** leads through the failure state **(phi)** and either to state **(c)** or to state **(a)**, depending on which was read.

By use the failure transition to encode the 'otherwise' paths into the WFSA, it is not necessary to explicity draw all 9 (= 3 x 3) arcs needed to specify the model as in part (a). There is no size reduction in this particular case. However for problems with large vocabularies and with longer-histories the savings can be substantial, under the assumption that backing-off happens more often than not.

(c) Redraw the machine of (b) with the failure transitions replaced by epsilon transitions. Explain how this may lead to incorrect assignment of probability to some sequences. Illustrate this point by calculating the likelihood of the sequences 'a b b' and 'a c b' .



Ignoring the initial state probability,
$-\log P(\text{'b b'}|\text{'a'}) = -\log P(\text{'b'}|\text{'b'}) - \log P(\text{'b'}|\text{'a'}) = 1.39 + 0.223 = 1.61$
$-\log P(\text{'c b'}|\text{'a'}) = -\log P(\text{'b'}|\text{'c'}) - \log P(\text{'c'}|\text{'a'}) = 0.916 + 2.30 = 11.5$
Considering paths from the state a, the weights assigned by the WFSAs in parts (a) and (b) agree with these scores.

In the WFSA of part (c), there are two possible paths from state a which can accept the sequence 'b b' :

$$a \overset{b/0.223}{\rightarrow} b \overset{eps/1.39}{\rightarrow} eps \overset{b/\bar{1}}{\rightarrow} b$$

$$a \overset{eps/2.3}{\rightarrow} eps \overset{b/\bar{1}}{\rightarrow} b \overset{eps/1.39}{\rightarrow} eps \overset{b/\bar{1}}{\rightarrow} b$$

The weight assigned to the sequence is

$$\begin{aligned} w(\text{'b b'}|\text{'a'}) &= 0.223 \otimes 1.39 \otimes \bar{1} \oplus 2.30 \otimes \bar{1} \otimes 1.39 \otimes \bar{1} \\ &= (0.223 + 1.39 + 0) \min(2.30 + 0 + 1.39 + 0) = 0.223 + 1.39 = 1.61 \end{aligned}$$

However in computing $P(\text{'c b'}|\text{'a'})$ there is only one possible path from state a which could accept the sequence 'c b' :

$$a \overset{eps/2.30}{\rightarrow} eps \overset{c/\bar{1}}{\rightarrow} c \overset{eps/0.916}{\rightarrow} eps \overset{b/\bar{1}}{\rightarrow} b$$

The weight assigned to this path is

$$w(\text{'c b'}|\text{'a'}) = 2.30 \otimes \bar{1} \otimes 0.916 \otimes \bar{1} = 2.30 + 0.916 = 3.22$$

As it happens, the lowest cost path which accepts 'b b' assigns the correct cost. However, this need not be the case, e.g. if $P(b|a)$ were 0.2 (rather than 0.8), and the 'otherwise' arcs were 0.1. In this case, the backoff path would win, and the WFSA would assign an incorrect weight.

8. The following sentences are to be used as a language model training set :

Sentence 1 : <s> a b b c </s>
Sentence 2 : <s> a c c a b </s>
Sentence 3 : <s> c a c c b </s>
Sentence 4 : <s> b b c a b </s>

(a) Tabulate the statistics needed to compute unigram, bigram, and trigram language models.

x	$f(x)$
<s>	4
</s>	4
a	5
b	7
c	7
total	27

x_1x_2	$f(x_1x_2)$
<s> a	2
<s> b	1
<s> c	1
a b	3
a c	2
b b	2
b c	2
b </s>	3
c a	3
c b	1
c c	2
c </s>	1

$x_1x_2x_3$	$f(x_1x_2x_3)$
<s> a b	1
<s> a c	1
<s> c a	1
<s> b b	1
a b b	1
a b </s>	2
a c c	2
b b c	2
b c a	1
b c </s>	1
c a b	2
c a c	1
c b </s>	1
c c a	1
c c b	1

Unigram, Bigram, and Trigram Counts Extracted from the Four Sentences

(b) Calculate maximum likelihood unigram, bigram, and trigram language models from these statistics.

x	$P(x)$
<s>	4/27
</s>	4/27
a	5/27
b	7/27
c	7/27

x_1x_2	$P(x_2\|x_1)$
<s> a	2/4
<s> b	1/4
<s> c	1/4
a b	3/5
a c	2/5
b b	2/7
b c	2/7
b </s>	3/7
c a	3/7
c b	1/7
c c	2/7
c </s>	1/7

$x_1x_2x_3$	$P(x_3\|x_2, x_1)$
<s> a b	1/2
<s> a c	1/2
<s> c a	1/1
<s> b b	1/1
a b b	1/3
a b </s>	2/3
a c c	2/2
b b c	2/2
b c a	1/2
b c </s>	1/2
c a b	2/3
c a c	1/3
c b </s>	1/1
c c a	1/2
c c b	1/2

Unigram, Bigram, and Trigram Probabilities Estimated from Counts

(c) Explain *discounting* and *backing-off* and illustrate your explanation using the statistics and models from (a) and (b).

Unigram probabilities are based on frequently observed counts, e.g. no unigram occurs less than five times. However, several bigrams – e.g. 'b a' , 'a a', and 'a </s>' – are not observed at all in the data set; the situation is significantly worse for trigram counts. Focusing on the bigram case, the maximum likelihood estimate of $P(b|a) = 0$ when estimated over this data. Backing off allows approximating $\hat{P}(b|a)$ as $\alpha(a)P(b)$, so that the probability of 'b' following 'a' is proportional to the unigram probability of 'b' , weighted by the back-off factor associated with the history 'a'. This approximation introduces probability into the distribution, therefore the distribution must be renormalized (so that the probability sums to 1.0). This is done through discounting, by which the probability of the frequently observed pairs is reduced so that the back-off bigram is a proper probability distribution.

9. A pair of sentences f_1^J and e_1^I are known to be translations. Under Model-2 their alignment is described by the process a_j, $j = 1 \ldots J$, such that

$$P(f_1^J, a_1^J, J|e_1^I) = \prod_{j=1}^{J} p_{M2}(a_j|j, I, J)\, p_T(f_j|e_{a_j})$$

Note: For simplicity, disregard the sentence length distribution $p_L(J|I)$.

(a) Derive the following expression for the efficient calculation of the translation posterior

$$P(f_1^J|e_1^I) = \prod_{j=1}^{J} \sum_{i=0}^{I} p_{M2}(i|j, I, J)\, p_T(f_j|e_i)$$

Hint: $P(f_1^J|e_1^I)$ can be written $\sum_{a_1^J} P(f_1^J, a_1^J|e_1^I) = \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} P(f_1^J, a_1^J|e_1^I)$.

Starting with the hint

$$\begin{aligned}
P(f_1^J|e_1^I) &= \sum_{a_1^J} P(f_1^J, a_1^J|e_1^I) = \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} P(f_1^J, a_1^J|e_1^I) \\
&= \sum_{a_1=0}^{I} \cdots \sum_{a_J=0}^{I} \prod_{j=1}^{J} p_{M2}(a_j|j, I, J)\, p_T(f_j|e_{a_j}) \\
&= \sum_{a_1=0}^{I} p_{M2}(a_1|1, I, J)\, p_T(f_1|e_{a_1}) \times \\
&\qquad \sum_{a_2=0}^{I} p_{M2}(a_2|2, I, J)\, p_T(f_2|e_{a_2}) \times \\
&\qquad\qquad \cdots \times \sum_{a_J=0}^{I} p_{M2}(a_J|J, I, J)\, p_T(f_J|e_{a_J}) \\
&= \prod_{j=1}^{J} \sum_{i=0}^{I} p_{M2}(i|j, I, J) p_T(f_j|e_i)
\end{aligned}$$

(b) Using the result of (a), derive the following expression for the efficient calculation of the alignment link posterior probability

$$P(a_j = i|e_1^I, f_1^J) = \frac{p_{M2}(i|j, I, J)\, p_T(f_j|e_i)}{\sum_{i'=0}^{I} p_{M2}(i'|j, I, J) p_T(f_j|e_{i'})}$$

First, note that we can write $P(a_1^J|f_1^J, e_1^I) = P(a_1^J, f_1^J|e_1^I)/P(f_1^J|e_1^I)$. Using the definition of the Model-2 joint distribution for the numerator and the result of (a) for the denominator,

$$\frac{P(a_1^J, f_1^J|e_1^I)}{P(f_1^J|e_1^I)} = \frac{\prod_{j=1}^{J} p_{M2}(a_j|j, I, J)\, p_T(f_j|e_{a_j})}{\prod_{j=1}^{J} \sum_{i'=0}^{I} p_{M2}(i'|j, I, J) p_T(f_j|e_{i'})} = \prod_{j=1}^{J} \frac{p_{M2}(a_j|j, I, J)\, p_T(f_j|e_{a_j})}{\sum_{i'=0}^{I} p_{M2}(i'|j, I, J) p_T(f_j|e_{i'})}$$

For simplicity, write $A_j(a_j) = \frac{p_{M2}(a_j|j,I,J)\, p_T(f_j|e_{a_j})}{\sum_{i'=0}^{I} p_{M2}(i'|j,I,J) p_T(f_j|e_{i'})}$ and note that $\sum_{a_j} A_j(a_j) = 1$.

By summing over indices other than j, it follows that

$$\begin{aligned}
P(a_j = i|e_1^I, f_1^J) &= \sum_{a_1} \cdots \sum_{a_{j-1}} \sum_{a_{j+1}} \cdots \sum_{a_J} \prod_{j=1}^{J} A_j(a_j) \\
&= \sum_{a_1} A_1(a_1) \sum_{a_2} A_2(a_2) \cdots \sum_{a_{j-1}} A_{j-1}(a_{j-1}) A_j(i) \sum_{a_{j+1}} A_{j+1}(a_{j+1}) \cdots \sum_{a_J} A_J(a_J) \\
&= A_j(i) \sum_{a_1} A_1(a_1) \sum_{a_2} A_2(a_2) \cdots \sum_{a_{j-1}} A_{j-1}(a_{j-1}) \sum_{a_{j+1}} A_{j+1}(a_{j+1}) \cdots \sum_{a_J} A_J(a_J) \\
&= A_j(i) \\
&= \frac{p_{M2}(i|j, I, J)\, p_T(f_j|e_i)}{\sum_{i'=0}^{I} p_{M2}(i'|j, I, J) p_T(f_j|e_{i'})}
\end{aligned}$$

(c) Give parameter update equations for the Model-2 component distributions in terms of these posterior probabilities.

Compute $P^{(r)}(a_j = i)$ as $P(a_j = i|E^{(r)}, F^{(r)})$, where $E^{(r)}$ and $F^{(r)}$ are the r^{th} pair of sentences from the parallel text. This can be computed efficiently using procedures derived in part (b).

Compute the word translation distribution.

Step 1 : Accumulate the expected number of times a word f is aligned to an word e.

$$\#_T(f \leftrightarrow e) = \sum_{r=1}^{R} \sum_{j=1}^{J^{(r)}} \sum_{i=1}^{I^{(r)}} \underbrace{1(\, e = e_i^{(r)}\,)}_{\substack{e \text{ is the } i^{th} \\ \text{word of } E^{(r)}}} \underbrace{1(\, f = f_j^{(r)}\,)}_{\substack{f \text{ is the } j^{th} \\ \text{word of } F^{(r)}}} P^{(r)}(a_j = i)$$

Step 2 :

$$p_T(f|e) = \frac{\#_T(f \leftrightarrow e)}{\sum_{f'} \#_T(f' \leftrightarrow e)}$$

Compute the alignment distribution.

Step 1 : Accumulate position alignment statistics over the word-aligned sentences

$$\#_{M2}(i, j, J, I) = \sum_{r=1}^{R} 1(J = J^{(r)}, I = I^{(r)}\,) \sum_{j=1}^{J} P^{(r)}(a_j = i)$$

Step 2 : Compute the Position Alignment Distribution

$$p_{M2}(i\,|\,j, J, I) = \frac{\#_{M2}(i, j, J, I)}{\sum_{i''=1}^{I} \#_{M2}(i'', j, J, I)}$$

(d) Using the results of (a) and (b), give efficient expressions for the efficient calculation of $P(f_1^J, a_1^J|e_1^I)$ and $P(a_j = i|e_1^I, f_1^J)$ under Model-1.

Noting that Model-1 is a special case of Model-2 with $p_{M2}(a_j = i|j, I, J) = \frac{1}{I+1}$,[1] calculation of $P(f_1^J, a_1^J|e_1^I)$ is straightforward:

$$P(f_1^J, a_1^J|e_1^I) = \frac{1}{I+1} \prod_{j=1}^{J} p_T(f_j|e_{a_j})$$

and the calculation of the posterior simplifies to

$$P(a_j = i|e_1^I, f_1^J) = \frac{p_T(f_j|e_i)}{\sum_{i'=0}^{I} p_T(f_j|e_{i'})}$$

(e) Suppose it is necessary to choose between $f^{(1)}$ = 'a b c d' and $f^{(2)}$ = 'a c b d' as possible translations of an English sentence e_1^I. Is Model-1 suitable for this task ? Justify your answer.

Using the result of part (a) with the Model-1 alignment distribution, it follows that under Model-1

$$P(f_1^J|e_1^I) = \prod_{j=1}^{J} \sum_{i=0}^{I} \frac{1}{I+1} P_T(f_j|e_i)$$

Under this distribution, which ignores order in the foreign sentences, $P(f^{(1)}|e_1^I)$ and $P(f^{(2)}|e_1^I)$ will be identical. Model-1 cannot distinguish between the two sentences and is therefore not suitable.

10. A pair of sentences f_1^J and e_1^I are known to be translations. Under the word-to-word alignment HMM their alignment is described by the process a_j, $j = 1 \ldots J$, such that

$$P(f_1^J, a_1^J, J|e_1^I) = \prod_{j=1}^{J} p_T(f_j|e_{a_j})\, p_{HMM}(a_j|a_{j-1}, I)$$

Note: For simplicity, disregard the sentence length distribution $p_L(J|I)$.

(a) Derive the following relationship for the efficient computation of the forward probability $\alpha_j(i) = P(a_j = i, f_1^j|e_1^I)$:

$$\alpha_j(i) = \sum_{i'} p_T(f_j|e_i)\, p_{HMM}(a_j = i|a_{j-1} = i')\, \alpha_{j-1}(i')$$

[1] This length assumes the use of a NULL word.

The derivation is analogous to that of the forward algorithm for HMMs as used in ASR:

$$\begin{aligned}
\alpha_j(i) &= P(a_j = i, f_1^j | e_1^I) \\
&= \sum_{i'} P(a_j = i, a_{j-1} = i', f_j, f_1^{j-1} | e_1^I) \\
&= \sum_{i'} P(f_j | a_j = i, a_{j-1} = i', f_1^{j-1}, e_1^I)\, P(a_j = i | a_{j-1} = i', f_1^{j-1}, e_1^I)\, P(a_{j-1} = i', f_1^{j-1} | e_1^I) \\
&= \sum_{i'} P(f_j | a_j = i, e_1^I)\, P(a_j = i | a_{j-1} = i')\, P(a_{j-1} = i', f_1^{j-1} | e_1^I) \\
&= \sum_{i'} p_T(f_j | e_i)\, p_{HMM}(a_j = i | a_{j-1} = i')\, \alpha_{j-1}(i')
\end{aligned}$$

(b) Derive a similar recursion for the efficient computation of the backward probability $\beta_j(i) = P(f_{j+1}^J | a_j = i, e_1^I)$.

The derivation is analogous to that of the backward algorithm for HMMs as used in ASR:

$$\begin{aligned}
\beta_j(i) &= P(f_{j+1}^J | a_j = i, e_1^I) \\
&= \sum_{i'} P(f_{j+1}, f_{j+2}^J, a_{j+1} = i' | a_j = i, e_1^I) \\
&= \sum_{i'} P(f_{j+2}^J | f_{j+1}, a_{j+1} = i', a_j = i, e_1^I)\, P(f_{j+1}, a_{j+1} = i' | a_j = i, e_1^I) \\
&= \sum_{i'} P(f_{j+2}^J | a_{j+1} = i', e_1^I)\, P(f_{j+1}, a_{j+1} = i' | a_j = i, e_1^I) \\
&= \sum_{i'} \beta_{j+1}(i')\, P(f_{j+1} | a_{j+1} = i', a_j = i, e_1^I)\, P(a_{j+1} = i' | a_j = i, e_1^I) \\
&= \sum_{i'} \beta_{j+1}(i')\, P(f_{j+1} | a_{j+1} = i', e_1^I)\, p_{HMM}(a_{j+1} = i' | a_j = i) \\
&= \sum_{i'} \beta_{j+1}(i')\, p_T(f_{j+1} | e_{i'})\, p_{HMM}(a_{j+1} = i' | a_j = i)
\end{aligned}$$

(c) Show that the alignment link posterior probability can be computed as

$$P(a_j = i \,|\, f_1^J, e_1^I) = \frac{\alpha_j(i)\beta_j(i)}{\sum_i \alpha_J(i)}$$

Hint: First show that $P(a_j = i, f_1^J | e_1^I) = \alpha_j(i)\beta_j(i)$.

Following the hint,

$$\begin{aligned}
P(a_j = i, f_1^J | e_1^I) &= P(a_j = i, f_1^j | e_1^I)\, P(f_{j+1}^J | a_j = i, e_1^I) \\
&= \alpha_j(i)\beta_j(i)
\end{aligned}$$

Noting that $P(f_1^J | e_1^I) = \sum_i P(a_J = i, f_1^J | e_1^I) = \sum_i \alpha_J(i)$,

$$P(a_j = i | f_1^J, e_1^I) = \frac{P(a_j = i, f_1^J | e_1^I)}{P(f_1^J | e_1^I)} = \frac{\alpha_j(i)\beta_j(i)}{\sum_i \alpha_J(i)}$$

(d) Show that statistics needed for estimation of the HMM transition probabilities can be found as

$$P(a_j = i, a_{j-1} = i' | f_1^J, e_1^I) = \frac{\alpha_{j-1}(i')\, p_{HMM}(i|i')\, p_T(f_j|e_i)\, \beta_j(i)}{\sum_i \alpha_J(i)}$$

This derivation follows that of part (c):

$$\begin{aligned}
P(a_j = i, a_{j-1} = i', f_1^J | e_1^I) &= P(a_{j-1} = i', f_1^{j-1} | e_1^I)\, P(a_j = i, f_j^J | f_1^{j-1}, a_{j-1} = i', e_1^I) \\
&= \alpha_{j-1}(i')\, P(a_j = i, f_j^J | f_1^{j-1}, a_{j-1} = i', e_1^I) \\
&= \alpha_{j-1}(i')\, P(a_j = i | f_1^{j-1}, a_{j-1} = i', e_1^I)\, P(f_j^J | a_j = i, f_1^{j-1}, a_{j-1} = i', e_1^I) \\
&= \alpha_{j-1}(i')\, p_{HMM}(a_j = i | a_{j-1} = i')\, P(f_j^J | a_j = i, e_1^I) \\
&= \alpha_{j-1}(i')\, p_{HMM}(a_j = i | a_{j-1} = i')\, P(f_j | a_j = i, e_1^I)\, P(f_{j+1}^J | f_j, a_j = i, e_1^I) \\
&= \alpha_{j-1}(i')\, p_{HMM}(a_j = i | a_{j-1} = i')\, p_T(f_j|e_i)\, \beta_j(i)
\end{aligned}$$

From this it follows that

$$P(a_j = i, a_{j-1} = i' | f_1^J, e_1^I) = \frac{P(a_j = i, a_{j-1} = i', f_1^J, e_1^I)}{P(f_1^J | e_1^I)} = \frac{\alpha_{j-1}(i')\, p_{HMM}(i|i')\, p_T(f_j|e_i)\, \beta_j(i)}{\sum_i \alpha_J(i)}$$

(e) Give parameter update equations for the alignment HMM component distributions in terms of these posterior probabilities.

Compute $P^{(r)}(a_j = i, a_{j-1} = i')$ as $P(a_j = i, a_{j-1} = i' | F^{(r)}, E^{(r)})$ where $E^{(r)}$ and $F^{(r)}$ are the r^{th} pair of sentences from the parallel text. This is computed efficiently using the above recursions.

Step 1 : Accumulate position-to-position occurrence statistics over the alignments

$$\#_{HMM}(i, i', I) = \sum_{r=1}^{R} 1(I = I^{(r)}) \sum_{j=1}^{J^{(r)}} \underbrace{P^{(r)}(\, a_j = i, a_{j-1} = i'\,)}_{e_{i'}^{(r)} \leftrightarrow f_{j-1}^{(r)} \text{ and } e_i^{(r)} \leftrightarrow f_j^{(r)}}$$

Step 2 : Compute the Markov Position Alignment Distribution from these occurrence statistics

$$p_{HMM}(i \,|\, i', I) = \frac{\#_{HMM}(i, i', I)}{\sum_{i''=1}^{I} \#_{HMM}(i'', i', I)}$$

A.deGispert
P.C.Woodland
W.J.Byrne
April 2014