

(520|600).666

## Information Extraction from Speech and Text

Homework # 6

Due March 26, 2009.

Review Chapter 6 from *Statistical Methods for Speech Recognition* by Frederick Jelinek.

1. **The Multiple-stack Algorithm as A\* Search:** Show that the multiple-stack algorithm of Section 6.6.1 is an instance of the generic A\* algorithm.

Specifically, the A\* search in Section 6.3 is concerned with finding the best hypothesis  $\hat{\mathbf{w}}_1^n$  among all hypotheses  $\mathbf{w}_1^n$  of a given length  $n$ . The parameter  $k$  is associated with the length of a partial hypothesis  $\mathbf{w}_1^k$ , and a goodness function  $F(\mathbf{w}_1^k) = g(\mathbf{w}_1^k) + d(\mathbf{w}_1^k)$  is defined for each partial hypothesis. Starting with the empty hypothesis in a *stack*, the current best partial hypothesis  $\hat{\mathbf{w}}_1^k$  is extended at each iteration to  $\hat{\mathbf{w}}_1^{k+1} = \hat{\mathbf{w}}_1^k \parallel \hat{w}_{k+1}$ , until the best partial hypothesis attains a length  $n$ .

However, given the acoustics  $\mathbf{a}_1^m = a_1, \dots, a_m$ , the goal in Section 6.6.1 is to find

$$\hat{\mathbf{w}}_1^n = \arg \max_{\mathbf{w}} P(\mathbf{a}_1^m, \mathbf{w}) = \arg \max_{\mathbf{w}} \log P(\mathbf{a}_1^m, \mathbf{w}),$$

among all word sequences  $\mathbf{w}$  regardless of their length, and  $n$  is not provided a priori. Let us contemplate how to extended the notation of Section 6.3 to this problem.

Define a partial hypothesis to be a *pair*  $\langle l, \mathbf{w}_1^k \rangle$ , where  $0 \leq l \leq m$ , and let

$$g(\langle l, \mathbf{w}_1^k \rangle) = \log P(\mathbf{a}_1^l, \mathbf{w}_1^k) = \max_{l_1, \dots, l_{k-1}} \sum_{i=1}^k \log P(\mathbf{a}_{l_{i-1}+1}^{l_i}, w_i), \quad 0 = l_0 \leq l_1 \leq \dots \leq l_k = l.$$

Define the length of a partial hypothesis  $\langle l, \mathbf{w}_1^k \rangle$  to be  $l$ , not  $k$ , so that the length of a complete hypothesis is  $m$ . Extending an  $l$  length hypothesis to a  $l+1$  length hypothesis therefore does not necessarily entail extending  $\mathbf{w}_1^k$  to  $\mathbf{w}_1^{k+1}$ : we could have  $\langle l, \mathbf{w}_1^k \rangle \rightarrow \langle l+1, \mathbf{w}_1^k \rangle$  just as easily as  $\langle l, \mathbf{w}_1^k \rangle \rightarrow \langle l+1, \mathbf{w}_1^k \parallel w_{k+1} \rangle$ . To see that the multiple-stack algorithm is an A\* algorithm, let

$$d(\langle l, \mathbf{w}_1^k \rangle) = \max_{\mathbf{z}} \max_{l_1, \dots, l_{|\mathbf{z}|-1}} \sum_{i=1}^{|\mathbf{z}|} \log P(a_{l_{i-1}+1}^{l_i}, z_i), \quad l = l_0 \leq l_1 \leq \dots \leq l_{|\mathbf{z}|} = m,$$

where  $|\mathbf{z}|$  is the length of the word sequence  $\mathbf{z}$ . Note that  $d(\langle l, \mathbf{w}_1^k \rangle)$  does not depend on  $\mathbf{w}_1^k$ !

- (a) Show that if an A\* search is conducted using a *single* stack, with  $g(\langle l, \mathbf{w}_1^k \rangle) + d(\langle l, \mathbf{w}_1^k \rangle)$  as the goodness of a partial hypothesis  $\langle l, \mathbf{w}_1^k \rangle$ , and is stopped the first time an  $m$  length hypothesis  $\langle m, \hat{\mathbf{w}}_1^n \rangle$  percolates to the top, then  $\hat{\mathbf{w}}_1^n$  is indeed the most likely word sequence.
- (b) Derive a relationship between  $d(\langle l, \mathbf{w}_1^k \rangle)$  and  $g^*(l)$  as defined in Equation (11) on p100.
- (c) Show that sorting the stack entries during this A\* search according to

$$F^*(\langle l, \mathbf{w}_1^k \rangle) = g(\langle l, \mathbf{w}_1^k \rangle) - g^*(l),$$

will also lead to the discovery of the same  $\hat{\mathbf{w}}_1^n$ , i.e.  $\langle m, \hat{\mathbf{w}}_1^n \rangle$  will be the first  $m$  length hypothesis to percolate to the top of the (single) stack.

Argue based on these results that when the multiple-stack algorithm on p101 stops, the top entry in the  $m$ -th stack is the most likely word sequence  $\hat{\mathbf{w}}_1^n$ .

2. **N-best Paths Using the Multiple-stack Algorithm:** Modify the multiple-stack algorithm of Section 6.6 to obtain the  $N$  best hypotheses instead of only the best. In particular, assume that the conditions (a), (b) and (c) of Section 6.6.1 on p100 are satisfied.

- (a) Minimally modify the algorithm of Section 6.6.1 to obtain the *two* best paths.
- (b) Check if the extension of Section 6.6.2 will hold for your new algorithm.
- (c) Generalize your modification for any  $N \geq 2$  best paths.

Discuss whether the extension to the actual multiple-stack algorithm of Section 6.6.3, when the assumptions (a), (b) and (c) do not hold, will be possible for your new algorithm. What are the pitfalls for  $N \geq 2$  that were not there for  $N = 1$ ?

Finish Project #2 even as you work on these problems. It is due two days before this homework.