CMPT 413: Computational Linguistics

ED2: Computing the Edit Distance

Anoop Sarkar

http://www.cs.sfu.ca/~anoop

Consider two strings:

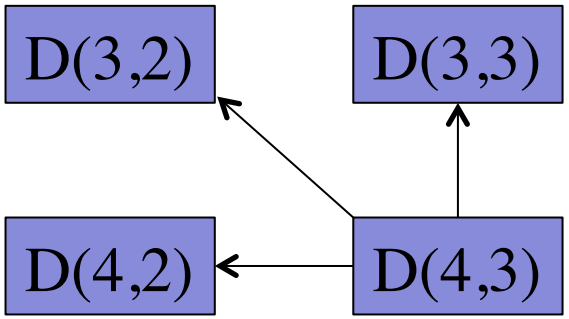
target = $g_1a_2m_3b_4l_5e_6$

source= $g_1u_2m_3b_4o_5$

- We want to find D(6,5)
- We find this recursively using values of D(i,j) where i≤6 j≤5
- For example, consider how to compute D(4,3)

target = $g_1a_2m_3b_4$

source= $g_1u_2m_3$



Take the minimum

- Case 1: SUBSTITUTE b_4 for m_3
- Use previously stored value for D(3,2)
- Cost($g_1a_2m_3b$ and g_1u_2m) = D(3,2) + cost(b≈m)
- For substitution: D(i,j) = D(i-1,j-1) + cost(subst)

- Case 2: INSERT b_4
- Use previously stored value for D(3,3)
- Cost($g_1a_2m_3b$ and $g_1u_2m_3$) = D(3,3) + cost(ins b)
- For substitution: D(i,j) = D(i-1,j) + cost(ins)

- Case 3: DELETE m_3
- Use previously stored value for D(4,2)
- Cost($g_1a_2m_3b_4$ and g_1u_2m) = D(4,2) + cost(del m)
- For substitution: D(i,j) = D(i,j-1) + cost(del)

Minimum Cost Edit Distance

- An alignment between target and source

Find $D(n,m)$ recursively

$$t_1, t_2, \ldots, \ldots, t_n$$
$$s_1, s_2, \ldots, s_m$$

$$D(i,j) = min \begin{cases} D(i-1,j) & +\text{cost}(t_i, \emptyset) \textit{ insertion into target} \\ D(i-1,j-1) & +\text{cost}(t_i, s_j) \textit{substitution/identity} \\ D(i,j-1) & +\text{cost}(\emptyset, s_j) \textit{deletion from source} \end{cases}$$

$$D(0,0) = 0$$

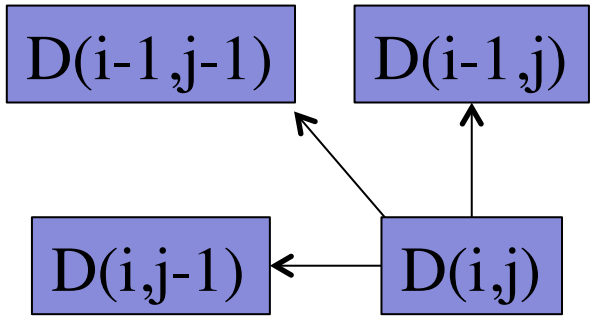
$$D(i,0) = D(i-1,0) + \text{cost}(t_i, \emptyset)$$

$$D(0,j) = D(0,j-1) + \text{cost}(\emptyset, s_j)$$

```
Function MinEditDistance (target, source)

 n = length(target)
 m = length(source)
 Create matrix D of size (n+1,m+1)
 D[0,0] = 0

 for i = 1 to n
   D[i,0] = D[i-1,0] + insert-cost

 for j = 1 to m
   D[0,j] = D[0,j-1] + delete-cost

 for i = 1 to n
   for j = 1 to m
     D[i,j] = MIN(D[i-1,j] + insert-cost,
                  D[i-1,j-1] + subst/eq-cost,
                  D[i,j-1] + delete-cost)
 return D[n,m]
```

D(i,j)

		g
	0	1
g	1	0



		g	u	m
	0	1	2	3
g	1	0	1	2
a	2	1	2	3
m	3	2	3	2
b	4	3	4	3

D(i,j)

		g	u	m	b	o
	0	1	2	3	4	5
g	1	0	1	2	3	4
a	2	1	2	3	4	5
m	3	2	3	2	3	4
b	4	3	4	3	2	3
l	5	4	5	4	3	4
e	6	5	6	5	4	5

Backtracing to find the alignments

D(i,j)

		g	u	m	b	o
	0	1	2	3	4	5
g	1	0 e	1	2	3	4
a	2	1	2 s	3	4	5
m	3	2	3	2 e	3	4
b	4	3	4	3	2 e	3
l	5	4	5	4	3 i	4
e	6	5	6	5	4	5 s

```
g a m b l e
|   | | |
g u m b _ o
```