CS314 - Edit Distance

Note Title 2/26/2010

Announcements

- HW is up

Edit Distance

The edit distance between two words is the minimum number of letter insertions, letter deletions, and letter substitutions required to transform one word into another.

Ex: Food to money : 4

FOOD → MOOD → MOND → MONY

(Just one possible way)

MONEY

edit distance ≤ 4 edit distance ≥ 3

Better display:

F	O	O		D
↓	↓	↓	↓	↓
M	O	N	E	Y

edit distance 4

Why can't you get 3?

at least 3 different letters, plus FOOD is shorter

Another: Algorithm to Altruistic

A	L	G	O	R		I		T	H	M
A	L		T	R	U	I	S	T	I	C
		1	+ 1		+1		+1		+1	+1 = $\boxed{6}$

So edit dist ≤ 6

Recursive idea:

Suppose we remove the last column.

What do we know about rest?

A L G O R I T H M

A L T R U I S T I C

d

$d-1$

spps could do better on substring

$d' < d-1$

$< d$

Lemma: If we remove last column, of optimal edit sequence the remaining columns must represent the shortest edit sequence for remaining substrings.

pf: by contradiction

If substring had a better edit sequence, then we could find a better edit sequence for the whole word.

So - recursive definition

my two words

Consider words $A[1..m]$ + $B[1..n]$

$A[1]$ $A[2]$ . . . $A[m]$

B →

$B[1]$ . . .

$B[n]$

- have substitution
- have insertion
- have deletion

What could happen in last column?

- if $A[m] = B[n]$, then free
- if $A[m] \neq B[n]$, rest +1

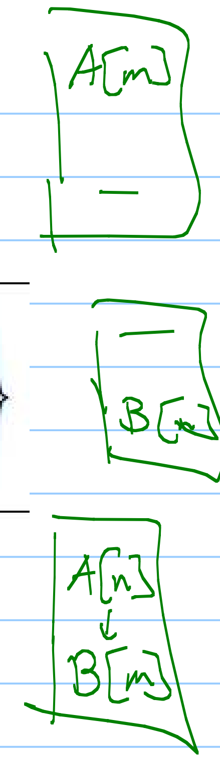
deletion +1

Formally:

deletion

$$Edit(A[1..m], B[1..n]) = \min \begin{cases} Edit(A[1..m-1], B[1..n]) + 1 \\ Edit(A[1..m], B[1..n-1]) + 1 \\ Edit(A[1..m-1], B[1..n-1]) + \underbrace{[A[m] \neq B[n]]}_{0 \text{ or } 1} \end{cases}$$

insertion



Base cases:

empty string

$$Edit(A[1..m], \varepsilon) = m, \qquad Edit(\varepsilon, B[1..n]) = n.$$

$$Edit(\varepsilon, \varepsilon) = 0$$

MOOD

↓

So (recap):

$$Edit(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \left\{ \begin{array}{l} Edit(i-1,j)+1, \\ Edit(i,j-1)+1, \\ Edit(i-1,j-1) + \underbrace{[A[i] \neq B[j]]}_{0 \text{ or } 1} \end{array} \right\} & \text{otherwise} \end{cases}$$

This gives a (nasty) recurrence.

$$T(0,n) = O(1)$$
$$T(m,0) = O(1)$$
$$T(m,n) = T(m-1,n) + T(m,n-1) + T(m-1,n-1) + O(1)$$

A trick - replace m + n with a single variable, $N = m+n$.

Then:

$$T(m,n) = \begin{cases} O(1) & \text{if } n = 0 \text{ or } m = 0, \\ \underbrace{T(m,n-1)}_{N-1} + \underbrace{T(m-1,n)}_{N-1} + \underbrace{T(n-1,m-1)}_{N-2} + O(1) & \text{otherwise.} \end{cases}$$

Becomes:

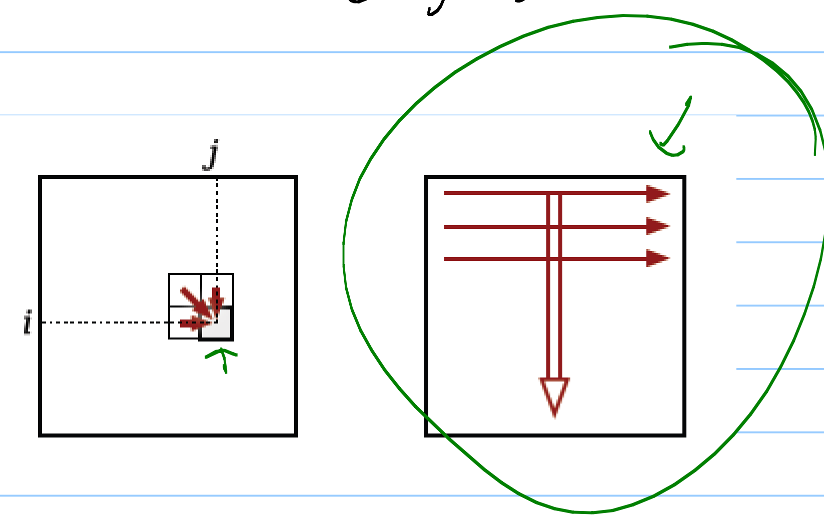
$$T'(N) = \max_{n+m=N} T(n,m) = \begin{cases} O(1) & \text{if } N = 0, \\ 2T(N-1) + T(N-2) + O(1) & \text{otherwise.} \end{cases}$$

(worse than Fibonacci!) way exponential

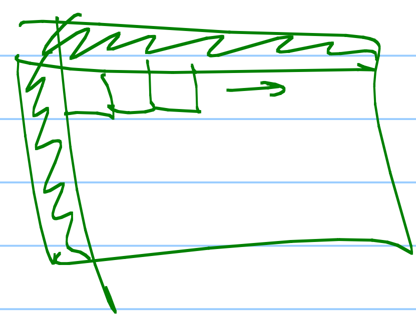
Smart recursion (aka memoization)

Keep 2 dimensional table Edit(m, n):

Edit(i, j) = the edit distance between A[1..i] and B[1..j]



Edit(i, j)



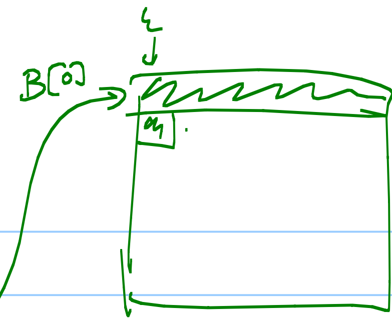
Space? How big is table? $n \times m$

How much time per entry?

need to check 3 other entries + maybe add 1, + take min

$\Rightarrow O(1)$

Total: $O(mn)$



Pseudocode:

EDITDISTANCE($A[1..m], B[1..n]$):

   for $j \leftarrow 1$ to n
      $Edit[0, j] \leftarrow j$

   for $i \leftarrow 1$ to m
      $Edit[i, 0] \leftarrow i$ ← know first entry of next row
      for $j \leftarrow 1$ to n
         if $A[i] = B[j]$
            $Edit[i, j] \leftarrow \min\{Edit[i-1, j] + 1,\ Edit[i, j-1] + 1,\ Edit[i-1, j-1]\}$
         else
            $Edit[i, j] \leftarrow \min\{Edit[i-1, j] + 1,\ Edit[i, j-1] + 1,\ Edit[i-1, j-1] + 1\}$

   return $Edit[m, n]$

fill in row i of the table

$O(nm)$

An example - Algorithm to Altruistic

horizontal = deletion

vertical = insertion

diagonal = substitution

	ε	A	L	G	O	R	I	T	H	M
ε	0	1	2	3	4	5	6	7	8	9
A	1	0	1	2	3	4	5	6	7	8
L	2	1	0	1	2	3	4	5	6	7
T	3	2	1	1	2	3	4	4	5	6
R	4	3	2	2	2	2	3	4	5	6
U	5	4	3	3	3	3	3	4	5	6
I	6	5	4	4	4	4	3	4	5	6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	4	5	6
I	9	8	7	7	7	7	6	5	5	6
C	10	9	8	8	8	8	7	6	6	6

Any path from top left to bottom right represents a valid edit sequence.

In our example, there are actually 3 optimal sequences:

```
A L G O R I   T H M
A L T R U I S T I C

A L G O R   I   T H M
A L   T R U I S T I C

A L G O R   I   T H M
A L T   R U I S T I C
```