Spell correction & edit distance

What's the problem with this image?

Spelling Correction

Cambodian acress allegedly gets possessed while playing a ghost in a movie

Any spelling error in the above sentence?

Spelling Correction

Cambodian acress allegedly gets possessed while playing a ghost in a movie

Any spelling error in the above sentence?

Which are some close words to ‘acress’?

Spelling Correction

Cambodian acress allegedly gets possessed while playing a ghost in a movie

Any spelling error in the above sentence?

Which are some close words to ‘acress’?

- actress
- acres
- ....

Types of spelling errors

Non-word spelling error detection

- Any word not in a dictionary is an error
- Examples: **acress** → **actress**

Real word spelling errors

- The erroneous word is itself present in a dictionary.
- Typographical: **three** → **there**
- Cognitive Errors (homophones): **piece** → **peace**

Edit Distance

Isolated word error correction

- Pick the one that is closest to ‘acress’
- How to define ‘closest’?
- Need a **distance metric**
- The simplest metric: **edit distance**

The minimum edit distance between two strings

The minimum number of editing operations

- Insertion
- Deletion
- Substitution

Minimum Edit Distance

H		O	N	D	A	
H	Y	U	N	D	A	I

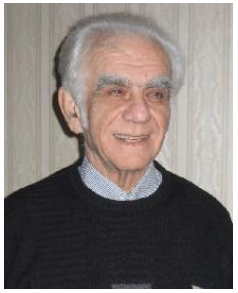
H	O		N	D	A	
H	Y	U	N	D	A	I

Minimum Edit Distance

H		O	N	D	A	
H	Y	U	N	D	A	I

H	O		N	D	A	
H	Y	U	N	D	A	I

- If each operation cost = 1, Distance = 3. (Levenshtein Distance)

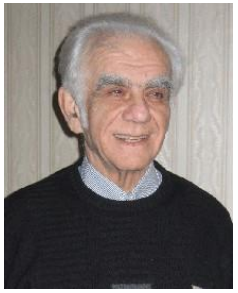


Minimum Edit Distance

H		O	N	D	A	
H	Y	U	N	D	A	I

H	O		N	D	A	
H	Y	U	N	D	A	I

- If each operation cost = 1, Distance = 3. (Levenshtein Distance)



- If substitution cost = 2, Distance = 4. (Alternate version)

Finding minimum edit distance: The simplest strategy

Process all characters one by one starting from either from left or right sides of both strings.

Algorithm

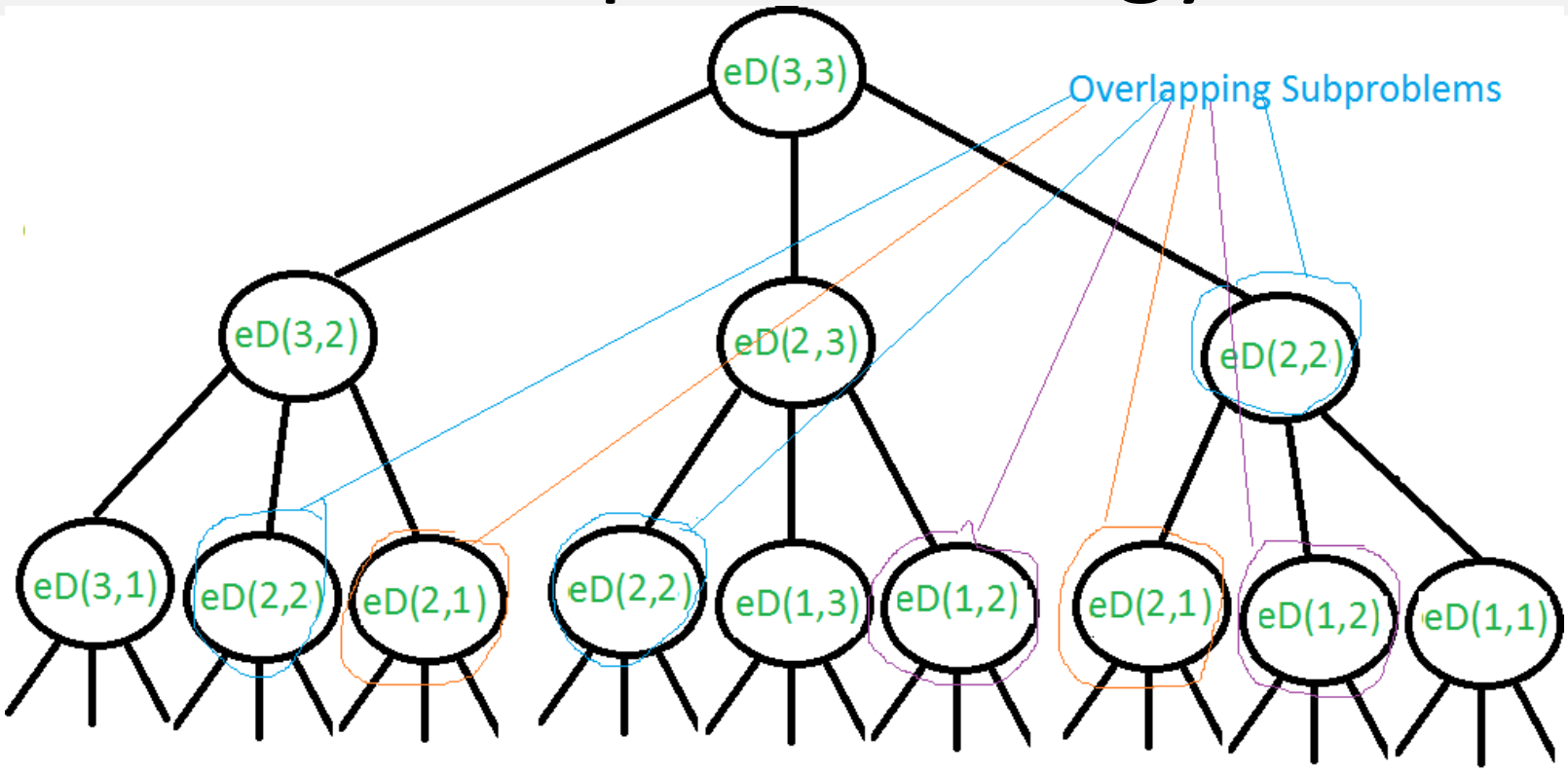
m: Length of X (first string)

n: Length of Y (second string)

$eD(i, j)$: edit distance between $X[1..i]$ and $Y[1..j]$, the first i characters of X and the first j characters of Y.

- If last characters of two strings are same, ignore last characters and recur for m-1 and n-1.
- Else:
  - Delete: Recur for m-1 and n
  - Insert: Recur for m and n-1
  - Substitute: Recur for m-1 and n-1

Finding minimum edit distance: The simplest strategy



Worst case recursion tree when m = 3, n = 3.
Worst case example str1="abc" str2="xyz"

https://www.geeksforgeeks.org/edit-distance-dp-5/

Finding minimum edit distance: Dynamic Programming

- A tabular computation of $D(n, m)$
- Solving problems by combining solutions to
- subproblems Bottom-up
  - Compute $D(i, j)$ for small i, j
  - Compute larger $D(i, j)$ based on previously computed smaller values
  - Compute $D(i, j)$ for all i and j till you get to $D(n, m)$

Dynamic Programming Algorithm

Initialization

```
D(i,0) = i
D(0,j) = j
```

Recurrence Relation:

```
For each  i = 1…M
      For each  j = 1…N
                          { D(i-1,j) + 1
          D(i,j)= min     { D(i,j-1) + 1
                          { D(i-1,j-1) +   2; { if X(i) ≠ Y(j)
                                          0; { if X(i) = Y(j)
```

Termination:

```
D(N,M) is distance
```

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4									
T	3									
N	2									
I	1									
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

The Edit Distance Table

N	9									
O	8									
I	7									
T	6									
N	5									
E	4	3	4							
T	3	4	5							
N	2	3	4							
I	1	2	3							
#	**0**	1	2	3	4	5	6	7	8	9
	#	E	X	E	C	U	T	I	O	N

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Minimum Edit with Backtrace

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ **8**	
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ **8**	← 9	
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	↙ **8**	← 9	← 10	
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ **8**	← 9	← 10	←↓ 11	
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ **8**	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10	
e	4	↙ 3	← 4	↙← **5**	← **6**	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	
t	3	↙←↓ 4	↙←↓ **5**	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8	
n	2	↙←↓ **3**	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7	
i	**1**	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8	
#	**0**	1	2	3	4	5	6	7	8	9	
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**	

- **Computing alignments:**

- For some applications, will have to find out which are the positions where edit took place.
- That which parts of string1 is aligned to string 2
- This can be done using backtracking. i.e. at each cell, will keep a pointer that we come from either D(i-1,j) or D(i,j-1) or D(i-1,j-1).
- And at the end, trace back the path from upper right corner to read off the alignment.
- i.e. back trace from final right corner cell value till bottom left corner null character.
- This is shown in fig by shaded elements.

Adding Backtrace to Minimum Edit

Base conditions:

$D(i,0) = i \qquad D(0,j) = j$

Termination:

$D(N,M)$ is distance

Recurrence Relation:

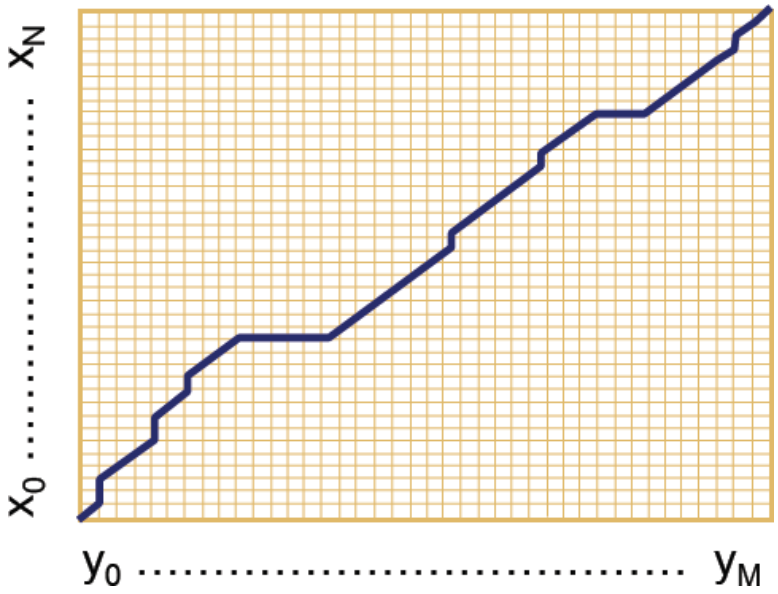
For each $i = 1 \ldots M$

For each $j = 1 \ldots N$

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 & \text{deletion} \\ D(i,j-1) + 1 & \text{insertion} \\ D(i-1,j-1) + \begin{cases} 2; \text{ if } X(i) \neq Y(j) & \text{substitution} \\ 0; \text{ if } X(i) = Y(j) \end{cases} \end{cases}$$

$$ptr(i,j) = \begin{cases} LEFT & \text{insertion} \\ DOWN & \text{deletion} \\ DIAG & \text{substitution} \end{cases}$$

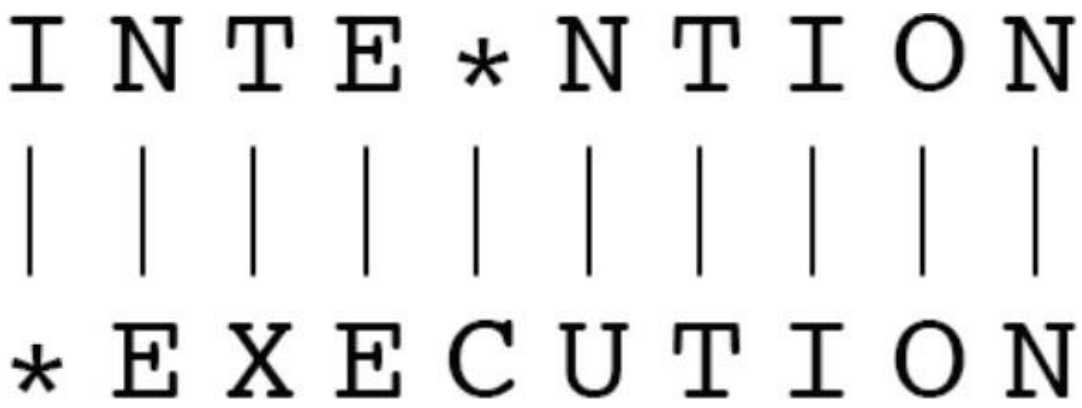
The distance matrix



Every non-decreasing path from (0,0) to (M,N) corresponds to an alignment of two sequences.

An optimal alignment is composed of optimal sub-alignments.

Result of Backtrace



Performance

Time

$O(nm)$

Space

$O(nm)$

Backtrace

Performance

Time

$O(nm)$

Space

$O(nm)$

Backtrace

$O(n+m)$

Weighted Edit Distance

Why to add weights to the computation?

- Some letters are more likely to be mistyped.

Confusion Matrix for Spelling Errors

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	Y (correct)																								
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Keyboard Design

Weighted Minimum Edit Distance

Initialization:

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

$$D(i,0) = D(i-1,0) + del[x(i)]; \quad 1 < i \leq N$$

$$D(0,j) = D(0,j-1) + ins[y(j)]; \quad 1 < j \leq M$$

Recurrence Relation:

$$D(i,j) = \min \begin{cases} D(i-1,j) + del[x(i)] \\ D(i,j-1) + ins[y(j)] \\ D(i-1,j-1) + sub[x(i),y(j)] \end{cases}$$

Termination:

$$D(N,M) \text{ is distance}$$

How to modify the algorithm with transpose?

Transpose

- $transpose(x, y) = (y, x)$
- Also known as metathesis

Marks Obtained

Transposition

* Transpose $(x, y) = (y, x)$
  ↳ Also known as metathesis

So how to modify the algorithm with transpose ?

→ Transpose is also common error /
for e.g. instead of xy, we write yx.

→ So now in current algorithm, after from insertion, deletion and substitution, how we take care of transposition

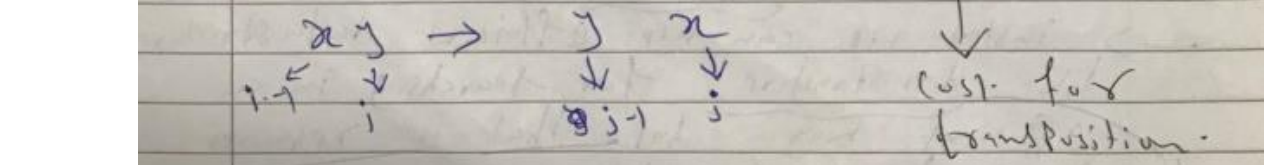
for e.g. $xy \rightarrow yx$

→ two positions, for two characters.

$D(i,j) = D(i-2, j-2) + 1$

↳ for transposition

if $x[i-1] = y[j]$
and $[xi] = y[j-1]$



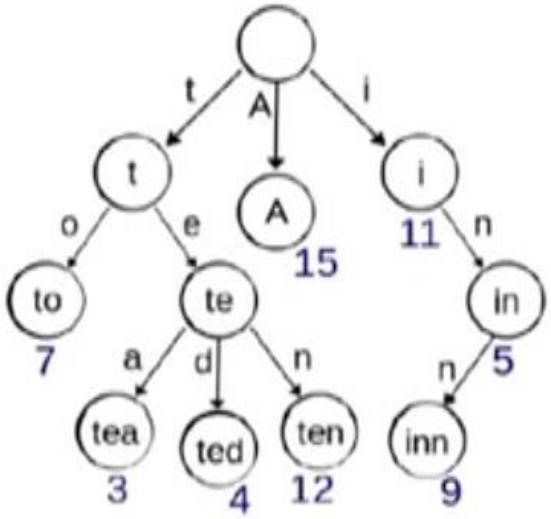
∴ in this way we can accomodate the cost for insert, deletion and sub..

How to find dictionary entries with smallest edit distance

Naïve Method

Compute edit ditance from the query term to each dictionary term – an exhaustive search

Can be made efficient if we do it over a trie structure



How to find dictionary entries with smallest edit distance

- Generate all possible terms with an edit distance <=2 (deletion + transpose + substitution + insertion) from the query term and search them in the dictionary.
- For a word of length 9, alphabet of size 36, this will lead to 114,324 terms to search for

How to find dictionary entries with smallest edit distance

Symmetric Delete Spelling Correction

- Generate terms with an edit distance ≤ 2 (deletes) from each dictionary term (offline)
- Generate terms with an edit distance ≤ 2 (deletes) from the input terms and search in dictionary

Spelling correction:

Types of spelling errors: Non-word Errors

- behaf → behalf

Non – word spelling errors

Non-word spelling error detection

- Any word not in a dictionary is an error
- The larger the dictionary the better

Non-word spelling error correction

- Generate candidates: real words that are similar to the error word
- Choose the best one:
  - Shortest weighted edit distance
  - Highest noisy channel probabliity

Real word spelling errors

For each word w, generate candidate set

- Find candidate words with similar pronunciations
- Find candidate words with similar spelling
- Include w in candidate set

Choosing best candidate

- Noisy Channel