UNIVERSITY OF SAN FRANCISCO

Master of Science in Analytics

Edit Distance

Natural Language Processing

Edit Distance

- Application: spelling correction
  - Given: Lexicon (dictionary of words)
  - Problem: A typist has misspelled a word
  - Task: propose a list of words, in order of most likely to least
- Motivating example:
  - Lexicon has (among other entries): access, acres, across, actress, caress, cress
  - Typist has produced: acress
- Generally:
  - How similar are two strings?
  - Used for machine translation, information extraction, speech recognition
  - Strings need not be limited to language; also used for alignment of nucleotides
- Minimum edit distance:
  - Ranks the candidates by number of changes
  - Produces alignments

Alignment — Examples (1)

- Evaluating Machine Translation and speech recognition

```
R Spokesman confirms    senior government adviser was shot
H Spokesman said    the senior            adviser was shot dead
              S      I           D                        I
```

- Named Entity Extraction and Entity Coreference
  - IBM Inc. announced today
  - IBM profits
  - Stanford President John Hennessy announced yesterday
  - for Stanford University President John Hennessy

Alignment — Examples (2)

- Base sequence (computational biology)
  - Example:

```
AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTTGCCCGAC
```

  - Alignment:

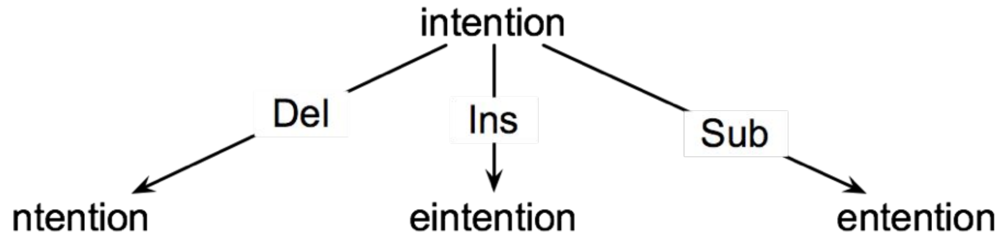
```
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC--GGTCGATTTGCCCGAC
```

- Old-fashioned spelling

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s   i s
```

Operations, Costs

- Given:
  - X (initial string — input by user?) with length of n
  - Y (target string — hypothesised by system?) with length of m
- Operations (cost / Levenshtein cost):
  - Delete (1 / 1) — remove a character when moving from initial to goal
  - Insert (1 / 1) — add a character when moving from initial to goal
  - Substitute (1 / 2) — change a character when moving from initial to goal
- Minimum cost / minimum memory
  - Minimum Edit Distance as search from initial to goal
  - There are several possible search paths, some redundant, each with a unique cost



  - We only care about search path with least cost (shortest path):
    - D(i,j): the edit distance between X[1..*i*] and Y[1..*j*]
    - Edit distance between the first *i* characters of X and *j* characters of Y
  - Keep track of only N possible changes

Computing Minimum Edit Distance

- Dynamic programming
  - First appeared in the 1950s as the marketing name of a US Defense research project
  - Start with some known alignment — eg. empty strings = D(0,0)
  - Bottom-up solution — compute D(i,j) for small i,j; compute larger values based on that
  - Combines subproblems; avoids redundancy
- Algorithm:
  - Initialisation:
    - D(i,0) = i
    - D(0,j) = j
  - Recurrence relation (pseudocode) for Levenshtein distance:

```
for each i = 1..m
     for each j = 1..n
          D(i,j) = min( D(i-1, j)+1  |  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 minimum edit distance
  - Substitute (1 / 2) — Remove one character and add another

Example Computation (1)

- What is the Levenshtein distance between “sets” and “seat”?

Example Computation (2)

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

Example Computation (2) Solved

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

Practice

- 1: Show the table of Levenshtein edit distance between the following:
  - acress vs. actress
  - acress vs. caress
- 2: Match usernames to names:
  - For each entry in `usernames.csv`, find the entry in `names.csv` which has the minimum distance
  - Check against `name-username.csv`, which has the correct alignment
  - What is the accuracy of this approach?
  - What went wrong? Why?

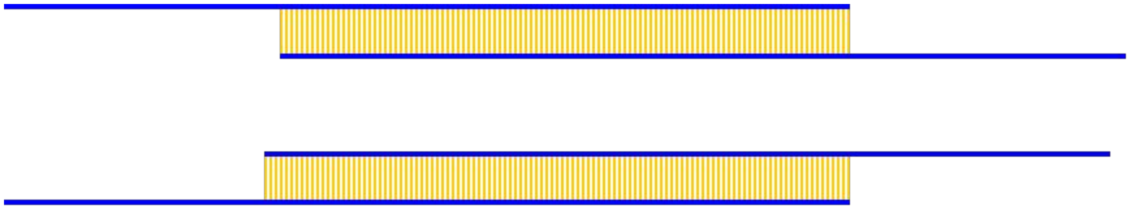
Backtrace

- How to produce alignments?
  - Need extra information in the table
  - Every time a cell is entered, place a trace of where it was filled from
  - Once table is filled, trace from D(n,m) back to D(0,0)
  - May prefer certain directions (eg. diagonal) over others
- Example:

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**

Minimum Edit Distance Variants

- Weighted Minimum Edit Distance
  - Spelling: certain substitutions (eg. a—>e) are more likely than others (eg. a—>g) — keyboard edit distance?
  - Biology: certain kinds of deletions / insertions are more likely than others
  - Adding cost function helps
- In Biology
  - Comparing genes / regions from different species finds mutations, important regions, evolutionary forces, etc.
  - Vocabulary: similarity (vs. distance) and scores (vs. weights)
  - Needleman-Wunsch algorithm: okay to have unlimited gaps at start/end of sequence



Example:
2 overlapping "*reads*" from a sequencing project

  - Smith-Waterman algorithm: ignore poorly-aligned regions; find areas of high local alignment

Autocorrect

- Problem:
  - Dynamic Programming solution is useful only after the word has been typed
  - Need a different framework if you want to predict what the user is typing

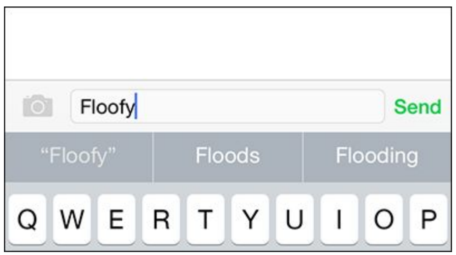


Image from How-To Geek

- One part of a solution: a **trie** data structure

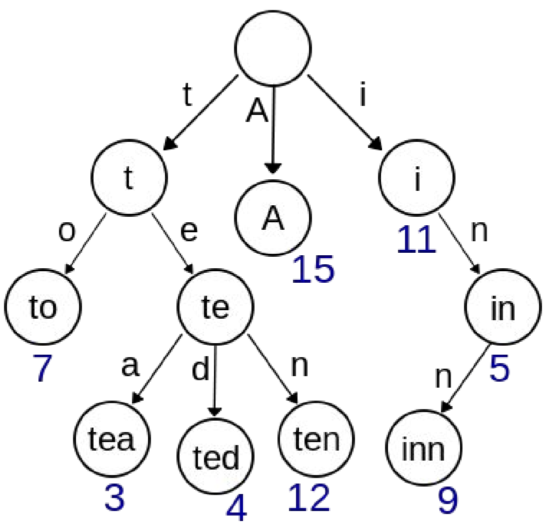


Image by Booyabazooka (based on PNG image by Deco). Modifications by Superm401. - own work (based on PNG image by Deco), Public Domain, https://commons.wikimedia.org/w/index.php?curid=1197221