

CMPT 413

Computational Linguistics

Anoop Sarkar

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

Minimum Cost Edit Distance

- String edit distance: what is the minimum number of changes (char insertions or deletions) to transform the string *intention* into *execution* ?
- Assume cost of insertion is 1 and cost of deletion is 1

Levenshtein Distance

- Cost is fixed across characters
 - Insertion cost is 1
 - Deletion cost is 1
- Two different costs for substitutions
 - Substitution cost is 1 (transformation)
 - Substitution cost is 2 (one deletion + one insertion)

Edit Distance

- Think of it as an alignment between target and source

t_1, t_2, \dots, t_n

Find $D(n, m)$ recursively

s_1, s_2, \dots, s_m

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

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

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

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

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]

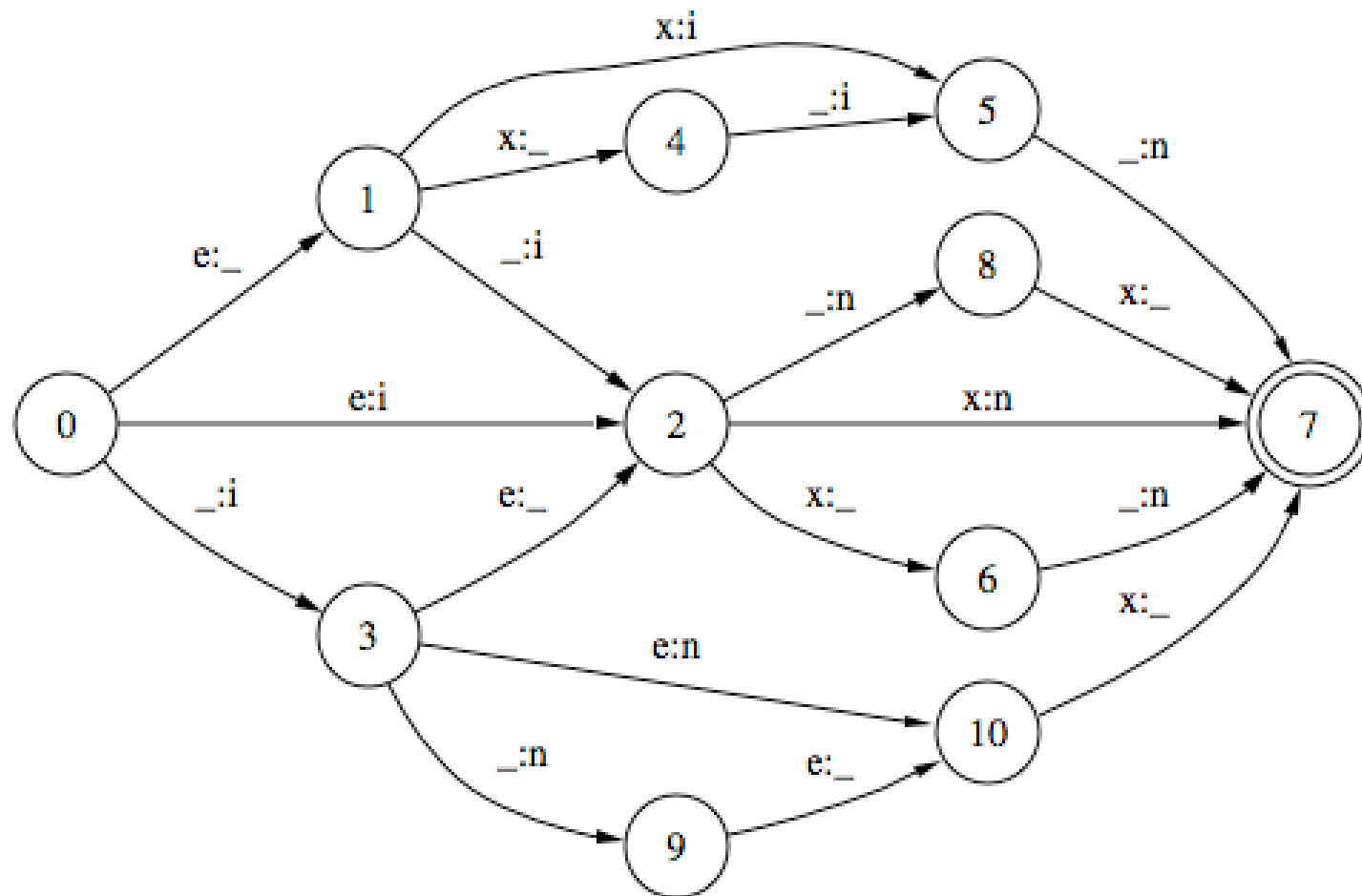
		target						
source		g	a	m	b	l	e	
		0	1	2	3	4	5	6
	g	1	0	1	2	3	4	5
	u	2	1	2	3	4	5	6
	m	3	2	3	2	3	4	5
	b	4	3	4	3	2	3	4
	o	5	4	5	4	3	4	5

The diagram illustrates the edit distance between source words and target words. The source words are g, u, m, b, o and the target words are g, a, m, b, l, e. The edit distance is calculated for each pair of words. The path of minimum edit distance is highlighted with arrows and labels: g to g (empty), g to u (insertion), u to m (deletion), m to b (deletion), b to o (deletion), and o to e (insertion).

Edit distance

- Useful in many NLP applications
- Comparing system output with human output, e.g.
input: *ibm* output: *IBM* vs. *Ibm*
- Error correction
- Defined over character edits or word edits, e.g. MT evaluation:
 - Foreign investment in Jiangsu ‘s agriculture on the increase
 - Foreign investment in Jiangsu agricultural investment increased

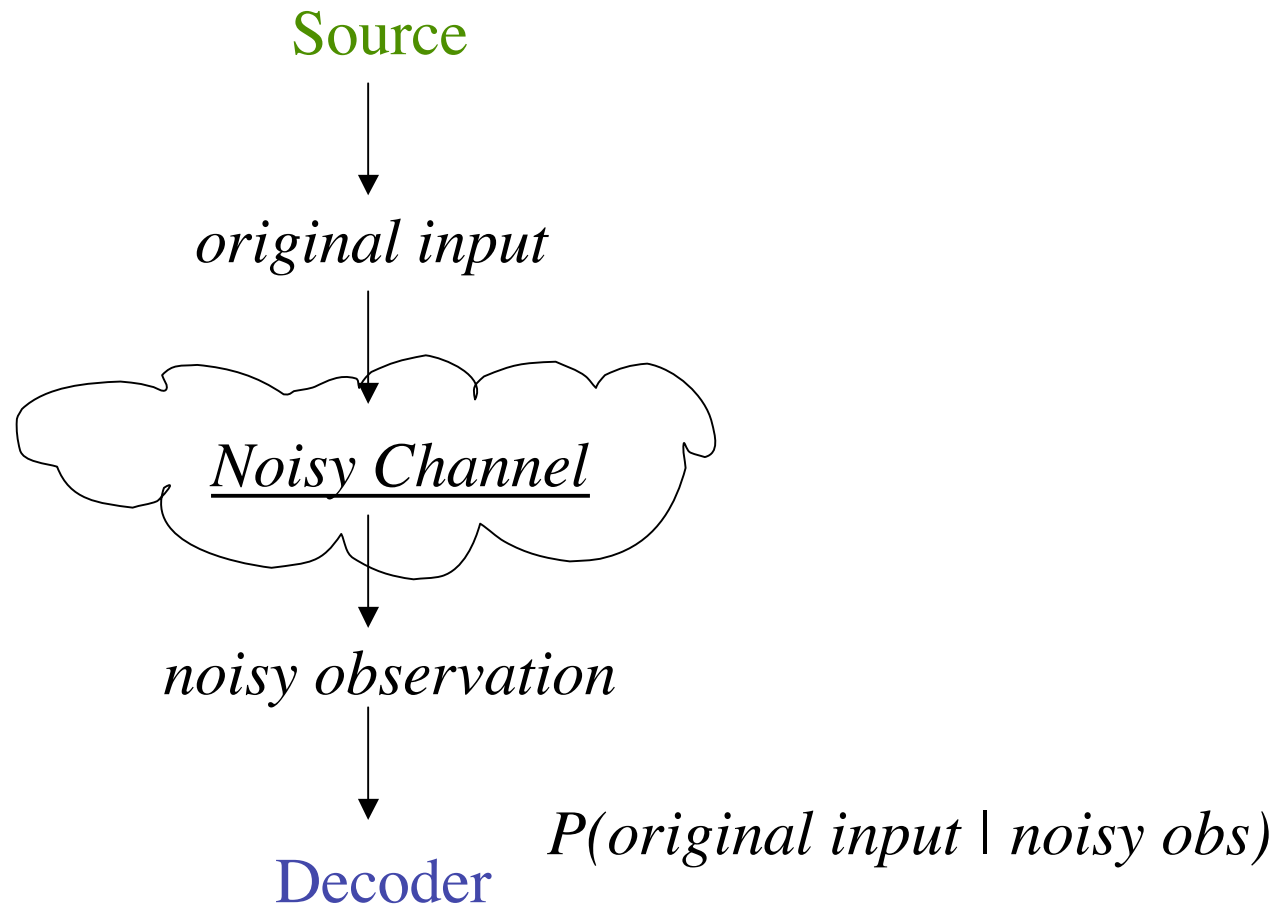
Edit distance and FSTs



Spelling Correction

- Types of spelling correction
 - non-word error detection
e.g. *hte* for *the*
 - isolated word error detection
e.g. *acres* vs. *access* (cannot decide if it is the right word for the context)
 - context-dependent error detection (real world errors)
e.g. *she is a talented acres* vs. *she is a talented actress*

Noisy Channel Model



Bayes Rule: *computing $P(\text{orig} \mid \text{noisy})$*

- let $x = \text{original input}$, $y = \text{noisy observation}$

$$p(x \mid y) = \frac{p(x, y)}{p(y)} \qquad p(y \mid x) = \frac{p(y, x)}{p(x)}$$

$$p(x, y) = p(y, x)$$

$$p(x \mid y) \times p(y) = p(y \mid x) \times p(x)$$

$$p(x \mid y) = \frac{p(y \mid x) \times p(x)}{\cancel{p(y)}} \qquad \underline{\text{Bayes Rule}}$$

Chain Rule

$$p(a, b, c \mid d) = p(a \mid b, c, d) \times \\ p(b \mid c, d) \times \\ p(c \mid d)$$

Approximations: Bias vs. Variance

$$p(a \mid b, c, d) \approx \frac{p(a \mid b, c)}{p(a \mid b)} \quad \text{less } \textit{bias}$$
$$p(a) \quad \text{more } \textit{variance}$$

Single Error Spelling Correction

- Insertion (addition)
 - acress vs. cress
- Deletion
 - acress vs. actress
- Substitution
 - acress vs. access
- Transposition (reversal)
 - acress vs. caress

Noisy Channel Model for Spelling Correction (Kernighan, Church and Gale, 1990)

- t is the typo and c is the correct word

$$P(c | t) = p(t | c) \times p(c)$$

- Find the best candidate for the correct word

$$\hat{c} = \arg \max_{c \in C} P(t | c) \times P(c)$$

$$P(t | c) = ?? \qquad P(c) = \frac{f(c)}{N}$$

Noisy Channel Model for Spelling Correction (Kernighan, Church and Gale, 1990)

single error, condition on previous letter

$$P(t \mid c) = \begin{cases} \frac{\text{del}[c_{p-1}, c_p]}{\text{chars}[c_{p-1}, c_p]} (xy)_c \text{ typed as } (x)_t \\ \frac{\text{ins}[c_{p-1}, t_p]}{\text{chars}[c_{p-1}]} (x)_c \text{ typed as } (xy)_t \\ \frac{\text{sub}[t_p, c_p]}{\text{chars}[c_p]} (y)_c \text{ typed as } (x)_t \\ \frac{\text{rev}[c_p, c_{p+1}]}{\text{chars}[c_p, c_{p+1}]} (xy)_c \text{ typed as } (yx)_t \end{cases}$$

Noisy Channel model for Spelling Correction

- The *del*, *ins*, *sub*, *rev* matrix values need data in which contain known errors
(**training data**)
- Accuracy on single errors on unseen data
(**test data**)

Noisy Channel model for Spelling Correction

- Experiments: 87% accuracy for machine vs. 98% average human accuracy
- What are the limitations of this model?
*... was called a “stellar and versatile **acress** whose combination of sass and glamour has defined her ...*

KCG model best guess is **acres**