

# Natural Language Processing

Lecture 6: Information Theory;  
Spelling, Edit Distance, and Noisy  
Channels

# Language Models

- Ngram models seem limited
  - Must be something better
- What about grammar/semantics?
  - But we care more about ranking good
  - Than ranking bad sentences
- Most LM are looking a “nearly” good examples
  - We care more about ranking near good
  - Than ranking very bad examples

# Neural Language Models

- Not just **previous** local context
  - What about future context
- Not just **local** context
  - What about words nearby
- Neural models aren't just about **N**-grams
  - They care about more context if its helpful
  - But you need lots of data to train from

# Neural Language Models

- BERT (ELMO)
  - Contextualized word embedding
  - Also a language model
- GPT-2/GPT-3
  - A more general language model
- Both using transformer neural models
  - Trained on lots and lots of data
- Give best LMs
  - if their training model matches yours (ish)

# A Taste of Information Theory

- Shannon Entropy,  $H(p)$
- Cross-entropy,  $H(p; q)$
- Perplexity

# Codebook

Horse	Code		
Clinton	000		
Edwards	001		
Kucinich	010		
Obama	011		
Huckabee	100		
McCain	101		
Paul	110		
Romney	111		

# Codebook

Horse	Code	Probability		
Clinton	000	1/4		
Edwards	001	1/16		
Kucinich	010	1/64		
Obama	011	1/2		
Huckabee	100	1/64		
McCain	101	1/8		
Paul	110	1/64		
Romney	111	1/64		

# Codebook

Horse	Probability	New Code
Clinton	1/4	10
Edwards	1/16	1110
Kucinich	1/64	111100
Obama	1/2	0
Huckabee	1/64	111101
McCain	1/8	110
Paul	1/64	111110
Romney	1/64	111111



# Three Spelling Problems

1. Detecting isolated non-words

“graffe” “exampel”

2. Fixing isolated non-words

“graffe” ⇨ “giraffe” “exampel” ⇨ “example”

3. Fixing errors in context

“I ate desert” ⇨ “I ate dessert”

“It was written be me” ⇨ “It was written by me”

# String edit distance

- How many letter changes to map A to B
- Substitutions
  - EXAMP EL
  - EXAMPLE 2 substitutions
- Insertions
  - EXA PLE
  - EXAMPLE 1 insertion
- Deletions
  - EXAMMPLE
  - EXA \_MPLE 1 deletion

# Levenshtein Distance

$$D_{0,0} = 0$$

$$D_{i,j} = \min \begin{cases} D_{i-1,j} + \text{inscost}(t_i) \\ D_{i,j-1} + \text{delcost}(s_j) \\ D_{i-1,j-1} + \text{substcost}(t_i, s_j) \end{cases}$$

# String Edit Distance

**function** MIN-EDIT-DISTANCE(*target*, *source*) **returns** *min-distance*

$n \leftarrow \text{LENGTH}(\textit{target})$

$m \leftarrow \text{LENGTH}(\textit{source})$

Create a distance matrix  $\textit{distance}[n+1, m+1]$

$\textit{distance}[0, 0] \leftarrow 0$

**for** each column  $i$  **from** 0 **to**  $n$  **do**

**for** each row  $j$  **from** 0 **to**  $m$  **do**

$\textit{distance}[i, j] \leftarrow \text{MIN}(\textit{distance}[i-1, j] + \textit{ins-cost}(\textit{target}_j),$   
   $\textit{distance}[i-1, j-1] + \textit{subst-cost}(\textit{source}_j, \textit{target}_i),$   
   $\textit{distance}[i, j-1] + \textit{ins-cost}(\textit{source}_j))$

---

**Figure 5.5** The minimum edit distance algorithm, an example of the class of dynamic programming algorithms.

# String edit distance

#	9	8	7	6	5	4	4	6	5	
L	8	7	6	5	4	3	3	5	7	
E	7	6	5	4	3	2	3	2	3	
P	6	5	4	3	2	1	2	3	4	
M	5	4	3	2	1	2	3	4	5	
M	4	3	2	1	0	1	2	3	4	
A	3	2	1	0	1	2	3	4	5	
X	2	1	0	1	2	3	4	5	6	
E	1	0	1	2	3	4	5	6	7	
#	0	1	2	3	4	5	6	7	8	
	#	E	X	A	M	P	L	E	#	

# String edit distance

#	9	8	7	6	5	4	4	6	5	
L	8	7	6	5	4	3	3	5	7	
E	7	6	5	4	3	2	3	2	3	
P	6	5	4	3	2	1	2	3	4	
M	5	4	3	2	1	2	3	4	5	
M	4	3	2	1	0	1	2	3	4	
A	3	2	1	0	1	2	3	4	5	
X	2	1	0	1	2	3	4	5	6	
E	1	0	1	2	3	4	5	6	7	
#	0	1	2	3	4	5	6	7	8	
	#	E	X	A	M	P	L	E	#	

# ~~Levenshtein~~ Hamming Distance

$$D_{0,0} = 0$$

$$D_{i,j} = \min \begin{cases} D_{i-1,j} + \infty \\ D_{i,j-1} + \infty \\ D_{i-1,j-1} + \text{substcost}(t_i, s_j) \end{cases}$$

# Levenshtein Distance with Transposition

$$D_{0,0} = 0$$

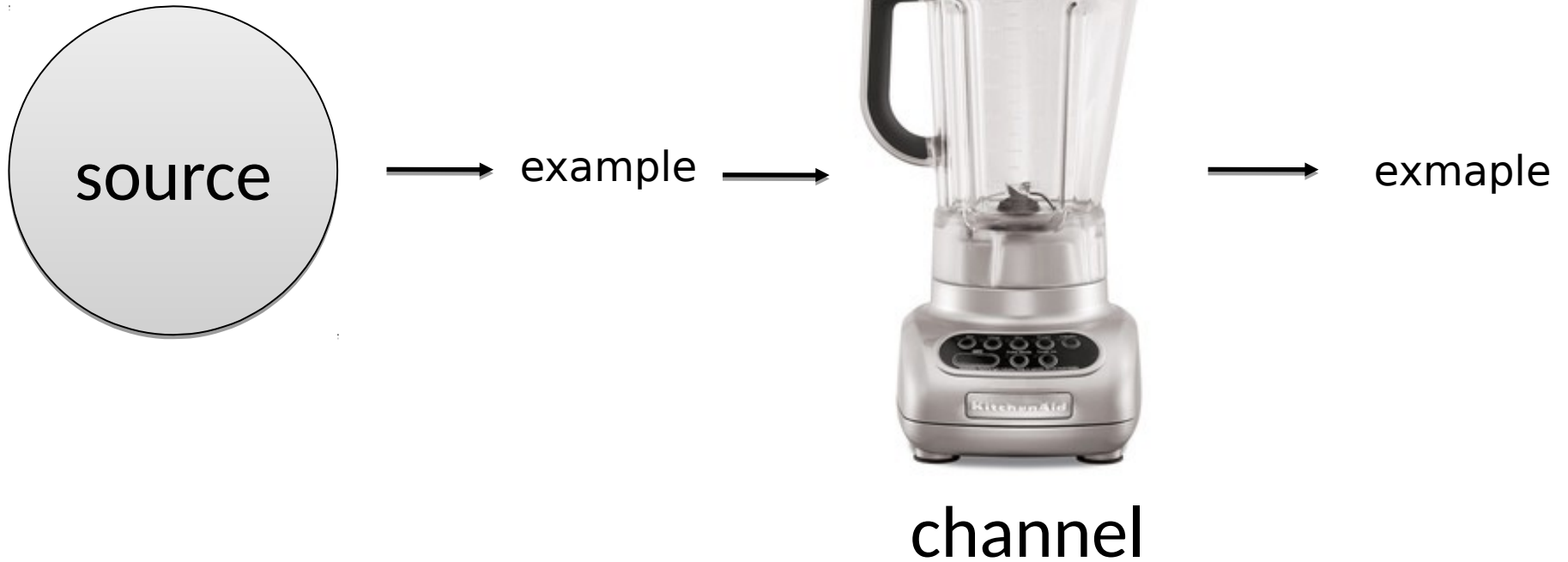
$$D_{i,j} = \min \begin{cases} D_{i-1,j} + \text{inscost}(t_i) \\ D_{i,j-1} + \text{delcost}(s_j) \\ D_{i-1,j-1} + \text{substcost}(t_i, s_j) \\ D_{i-2,j-2} + \text{transcost}(s_{j-1}, s_j) \text{ if } s_{j-1} = t_i \text{ and } s_j = t_{i-1} \end{cases}$$



# Three Spelling Problems

- ✓ Detecting isolated non-words
- ✓ Fixing isolated non-words
- 3. Fixing errors in context

# Kernighan's Model: A Noisy Channel



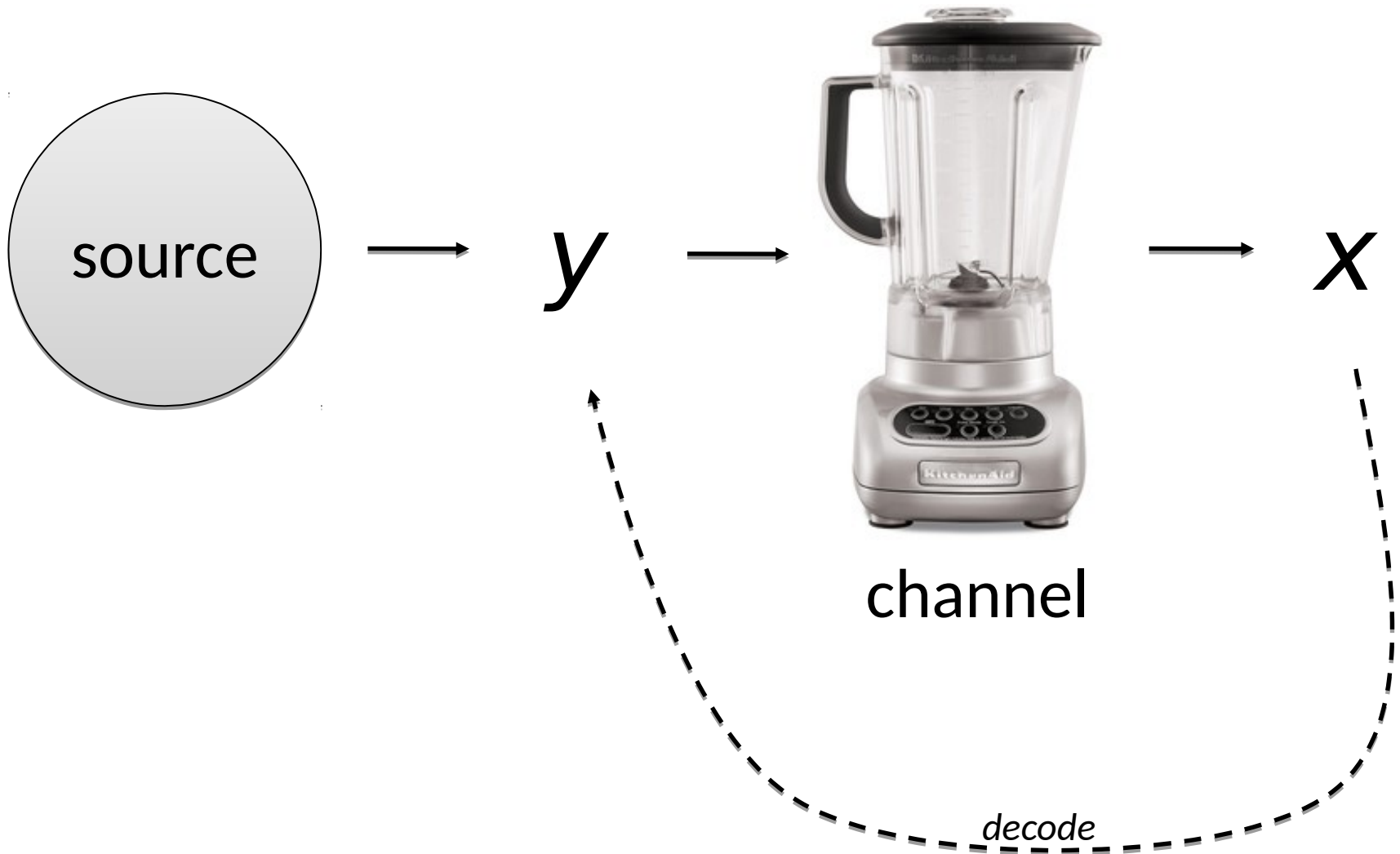
# acress

$c$	freq( $c$ )	$p(t   c)$	%
actress	1343	p(delete t)	37
cress	0	p(delete a)	0
caress	4	p(transpose a & c)	0
access	2280	p(substitute r for c)	0
across	8436	p(substitute e for o)	18
acres	2879	p(delete s)	21
...			

# How to choose between options

- Probabilities of edits
  - Insertions, deletions, substitutions,
  - Transpositions
- Probability of the new word

# Noisy Channel Model (General)



# Probability model

- Most likely word given observation
  - $\text{Argmax} ( P(W | O) )$
- By Bayes Rule is equivalent to
  - $\text{Argmax} ( \frac{P(W)P(O|W)}{P(O)} )$
- Which is equivalent to
  - $\text{Argmax} ( P(W) P(O|W) )$  (denom is constant)
- $P(O | W)$  calculated from edit distance
- $P(W)$  calculated from language model