Finite-State and the Noisy Channel
Word Segmentation

theprophetsaidtothecity

- What does this say?
  - And what other words are substrings?
- Could segment with parsing (how?), but slow.

- Given L = a “lexicon” FSA that matches all English words.
- How to apply to this problem?
- What if Lexicon is weighted?
- From unigrams to bigrams?
- Smooth L to include unseen words?
Spelling correction

- Spelling correction also needs a lexicon $L$
- But there is distortion ...
  - Let $T$ be a transducer that models common typos and other spelling errors
    - $\text{ance} \rightarrow \text{ence}$ (deliverance, ...)
    - $e \rightarrow \varepsilon$ (deliverance, ...)
    - $\varepsilon \rightarrow e // \text{Cons} _ \text{Cons}$ (athlete, ...)
    - $rr \rightarrow r$ (embarrass occurrence, ...)
    - $ge \rightarrow dge$ (privilege, ...)
    - etc.
  - Now what can you do with $L \cdot o \cdot T$ ?
  - Should $T$ and $L$ have probabilities?
  - Want $T$ to include “all possible” errors ...
Noisy Channel Model

real language $X$

noisy channel $X \rightarrow Y$

yucky language $Y$

want to recover $X$ from $Y$
Noisy Channel Model

real language $X$

noisy channel $X \rightarrow Y$

yucky language $Y$

want to recover $X$ from $Y$
Noisy Channel Model

real language $X$

noisy channel $X \rightarrow Y$

yucky language $Y$

(lexicon space)*
delete spaces
text w/o spaces

want to recover $X$ from $Y$
Noisy Channel Model

real language \( X \)

language model

noisy channel \( X \rightarrow Y \)

acoustic model

yucky language \( Y \)

(lexicon space)*

pronunciation

speech

want to recover \( X \) from \( Y \)
Noisy Channel Model

real language $X$

probabilistic CFG

noisy channel $X \rightarrow Y$

delete everything but terminals

tree
text

yucky language $Y$

want to recover $X$ from $Y$
Noisy Channel Model

real language $X$

noisy channel $X \rightarrow Y$

yucky language $Y$

$p(X)$

$p(Y \mid X)$

$p(X,Y) = p(X,Y)$

want to recover $x \in X$ from $y \in Y$

choose $x$ that maximizes $p(x \mid y)$ or equivalently $p(x,y)$
Noisy Channel Model

\[ p(X) \]

\[ * \]

\[ p(Y \mid X) \]

\[ = \]

\[ p(X,Y) \]

Note \( p(x,y) \) sums to 1.

Suppose \( y=\text{“C”}; \) what is best “\( x \)?
Noisy Channel Model

Suppose y=“C”; what is best “x”?

\[
p(X) \ast p(Y | X) = p(X,Y)
\]
Noisy Channel Model

\[
p(X) 
\] * 

\[
p(Y \mid X) 
\] * 

\[(Y=y)\]?

\[
p(X, y) 
\]
Morpheme Segmentation

- Let *Lexicon* be a machine that matches all Turkish words
  - Same problem as word segmentation
  - Just at a lower level: morpheme segmentation
  - Turkish word: `uygarlaştıramadıklarımızdan mış sınsınizcasına`
    = `uygar+laş+tır+ma+dık+ları+mız+dan+mış+sınız+ca+si+na`
    (behaving) as if you are among those whom we could not cause to become civilized
  - Some constraints on morpheme sequence: bigram probs
  - Generative model – concatenate then fix up joints
    - stop + -ing = stopping, fly + -s = flies, vowel harmony
    - Use a cascade of transducers to handle all the fixups
  - But this is just morphology!
  - Can use probabilities here too (but people often don’t)
Edit Distance Transducer

- \(O(k)\) deletion arcs
- \(O(k^2)\) substitution arcs
- \(O(k)\) insertion arcs
- \(O(k)\) no-change arcs
Stochastic Edit Distance Transducer

Likely edits = high-probability arcs
Stochastic Edit Distance Transducer

clara
.0.
a:b
ε:a
ε:b
b:a
ε:b
b:b
0.
caca

Best path (by Dijkstra’s algorithm)
Speech Recognition by FST Composition  
(Pereira & Riley 1996)

trigram language model $p(\text{word seq})$

pronunciation model $p(\text{phone seq} | \text{word seq})$

acoustic model $p(\text{acoustics} | \text{phone seq})$

observed acoustics
Speech Recognition by FST Composition
(Pereira & Riley 1996)

trigram language model

\[ p(\text{word seq}) \]

\[ p(\text{phone seq} \mid \text{word seq}) \]

\[ p(\text{acoustics} \mid \text{phone seq}) \]