CS674 Natural Language Processing

- Last week
  - Word sense disambiguation
- Today
  - Noisy channel model
    - » Pronunciation variation in speech recognition

The pronunciation subproblem

[spooky music][music stops]

Head Knight of Ni: Ni!

Knights of Ni: Ni! Ni! Ni! Ni! Ni!

Arthur: Who are you?

Head Knight: We are the Knights Who Say…'Ni'! …

We are the keepers of the sacred words: 'Ni', 'Peng', and 'Neee-wom'!

The pronunciation subproblem

- Given a series of phones, compute the most probable word that generated them.
- Simplifications
  - Given the correct string of phones
    - » Speech recognizer relies on probabilistic estimators for each phone, so it's never entirely sure about the identification of any particular phone
  - Given word boundaries
- "I [ni]…"
  - [ni] → *neat, the, need, new, knee, to,* and *you*
  - Based on the (transcribed) Switchboard corpus
- Contextually-induced pronunciation variation

Probabilistic transduction

- surface representation → lexical representation
- string of symbols representing the pronunciation of a word in context → string of symbols representing the dictionary pronunciation
  - [er] → *her, were, are, their, your*
  - exacerbated by **pronunciation variation**
    - » *the* pronounced as THEE or THUH
    - » some aspects of this variation are systematic
- sequence of letters in a mis-spelled word → sequence of letters in correctly spelled words
  - *acress* → *actress, cress, acres*

Noisy channel model

- Channel introduces noise which makes it hard to recognize the true word.
- **Goal:** build a model of the channel so that we can figure out how it modified the true word…so that we can recover it.

Decoding algorithm

- Special case of **Bayesian inference**
  - Bayesian classification
    - » Given observation, determine which of a set of classes it belongs to.
    - » Observation
      - ◆string of phones
    - » Classify as a
      - ◆word in the language

Pronunciation subproblem

- Given a string of phones, e.g. [ni], determine which word corresponds to this string of phones, *O*
  - Consider all words in the vocabulary, *V*
  - Select the single word, *w,* such that *P (word* | *observation)* is highest

$$\hat{w} = \underset{w \in V}{\arg\max}\; P(w \mid O)$$

Bayesian approach

- Use Bayes' rule to transform into a product of two probabilities, each of which is easier to compute than *P(w|O)*

$$P(x \mid y) = \frac{P(y \mid x) \quad P(x)}{P(y)}$$

$$\hat{w} = \underset{w \in V}{\arg\max}\; \frac{\overbrace{P(O \mid w)}^{\text{likelihood}} \quad \overbrace{P(w)}^{\text{prior}}}{P(O)}$$

Pronunciation subproblem

- Compute

$$\hat{w} = \arg\max_{w \in W} \overbrace{P(y \mid w)}^{\text{likelihood}} \overbrace{P(w)}^{\text{prior}}$$

- where *y* represents the sequence of phones (e.g. [ni])
- and *w* represents the candidate word

Computing the prior

- Using the relative frequency of the word in a large corpus
  - Brown corpus and Switchboard Treebank

w	freq(w)	P(w)
knee	61	.000024
the	114,834	.046
neat	338	.00013
need	1417	.00056
new	2625	.001

Probabilistic rules for generating pronunciation likelihoods

- Take the rules of pronunciation (see chapter 4 of J&M) and associate them with probabilities
  - Nasal assimilation rule
- Compute the probabilities from a large labeled corpus (like the transcribed portion of Switchboard)
- Run the rules over the lexicon to generate different possible surface forms each with its own probability

Sample rules that account for [ni]

Word	Rule Name	Rule	P
the	nasal assimilation	$ð \Rightarrow n$ / [+*nasal*] # __	[.15]
neat	final t deletion	$t \Rightarrow \emptyset$ / V __ #	[.52]
need	final d deletion	$d \Rightarrow \emptyset$ / V __ #	[.11]
new	u fronting	$u \Rightarrow i$ / __ # [y]	[.36]

Final results

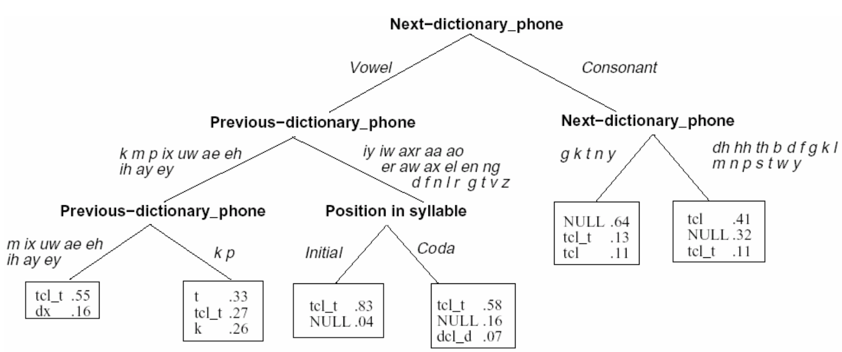
- *new* is the most likely
- Turns out to be wrong
  - *"I [ni]…"*

w	p(y|w)	p(w)	p(y|w)p(w)
new	.36	.001	.00036
neat	.52	.00013	.000068
need	.11	.00056	.000062
knee	1.00	.000024	.000024
the	0	.046	0

Decision trees for encoding lexical-to-surface pronunciation mappings

- Alternative to writing probabilistic pronunciation rules by hand is to learn the rules
- Decision tree approach
  - Riley (1991), Withgott and Chen (1993)
- Input to decision tree: a lexical phone described in terms of a set of features
- Output: classification and a probability

Example



Automatic induction of decision trees

- Riley / Withgott and Chen
  - Used CART (Breiman et al. 1984)
  - C4.5/C5.0 is an alternative
- How are decision trees induced automatically?
  - Training examples
  - Top-down induction

Training

- One tree for each lexical phone, p
  - One example for each occurrence lexical phone in corpus
  - Class value: surface realization of p
  - Features: previous-lexical-phone, next-lexical-phone, position-in-syllable