Speech recognition and synthesis

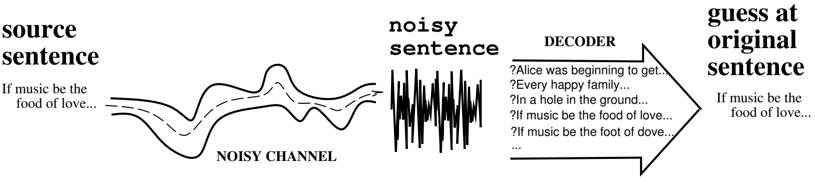
Introduction

Speech recognition in Human Machine interaction

- A full interaction requires human input
- Input with speech is often faster and easier than with text or pointers
  - Over the phone
  - With large or unlimited choice, eg, person and place names
  - Free text, eg, dictation messages
  - With hands occupied, eg, while driving
- Sometimes speech input is ineffective
  - In a noisy surrounding, eg, a train station
  - With small menu based selections
  - Large variation in speakers, eg, second language speakers
  - Tasks that are difficult to describe verbally, eg, routing on a map

Many pictures (and their copyrights) are from [Jurafsky and Martin(2000)]

Automatic Speech Recognition



ASR is a database retrieval problem

- A speech recognizer is a clever example database
- The problem is: How to retrieve the most likely words from the acoustic signal
- Break down into two problems: Get the most likely
  - word candidates given the sound
  - word sequence given the available word candidates
- Currently both problems are solved stochastically

Speech Input: How to partition the ASR problem

What is the most likely word sequence given the observed sound:

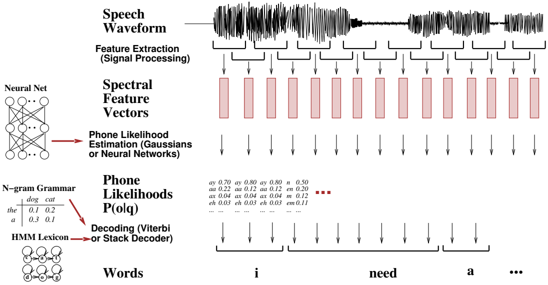
$$\underset{Words}{argmax}\ P\left(Words|Observation\right) =$$

$$\underset{Words}{argmax} \frac{P\left(Observation|Words\right) \cdot P\left(Words\right)}{P\left(Observation\right)}$$

Split this into two separate tasks

- $P\left(Observation\right)$ is a normalization constant, independent of word recognition (ignore it)
- $P\left(Observation|Words\right)$ is the acoustic *likelihood* of the words
- $P\left(Words\right)$ is the *prior* of the word sequence (i.e. the language model)

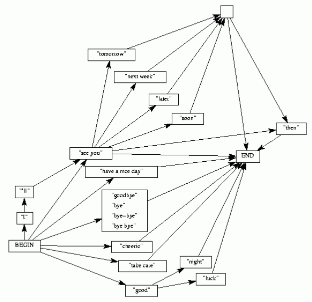
Speech Input: An overview of ASR



Sound waveform to word sequence

- Encode the waveform into Spectral Features
- Determine word likelyhoods $P(Sound|Words)$ for each word
- Determine word sequence probability $P(Words)$ for each sequence

Language Prior: $P(Words)$



Farewell Finite State example

every arrow has a probability

- The probability of observing an utterance
- Example from `http://www.geocities.com/SoHo/Square/3472/nounphrase.html`

Language Prior: Word sequences

Estimate $P\left(Words\right) =$

$$P\left(w_1, \ldots, w_n\right) = \prod_{i=1}^{n} P\left(w_i | w_1 \ldots w_{i-1}\right)$$

Approximate $P\left(Words\right)$ by modelling $P\left(w_i | w_1 \ldots w_{i-1}\right) \approx$

- $P\left(w_i | State_\alpha\right)$: Finite State Grammar
- $P\left(w_i | w_{i-n+1} \ldots w_{i-1}\right)$: N-gram
- $\sum_\alpha P\left(w_i | Tree_\alpha\left(w_1 \ldots w_{i-1}\right)\right) \cdot P\left(Tree_\alpha\left(w_1 \ldots w_{i-1}\right)\right)$: Context Free Grammar with (lexicalized) tree structures build from $\left(w_1 \ldots w_{i-1}\right)$

Language Prior: N-grams

Collect *word*, *word-pair* and *word-triplet* frequencies [Goodman(2001)]

- Impossible to get frequencies of all possible bi/trigrams
- Construct smoothed probability distributions
- Special "states" for sentence start and "end"
- $P\,(Words) \approx P\,(w_i | w_{i-2}, w_{i-1})$
- Use interpolation or backoff, eg, $P\,(w_i | w_{i-2}, w_{i-1}) \approx \alpha \cdot P\,(w_i | w_{i-1})$ if the tri-gram (w_{i-2}, w_{i-1}, w_i) was not observed

Language Prior: Data Oriented Parsing (CFG Example) [?]

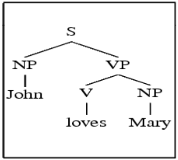


Fig. 1. A toy treebank

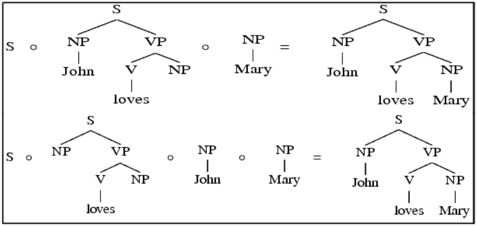


Fig. 2. Two different derivations of the same parse

Subtree have occurrence and insertion probabilities

- Requires a treebank with frequencies
- Correct normalization of probabilities
- Computationally expensive, like all probabilistic CF parsers

Language Prior: Grammar Perplexity

$$Perplexity\left(\mathfrak{G}\right) = 2^{H(\mathfrak{G})}$$

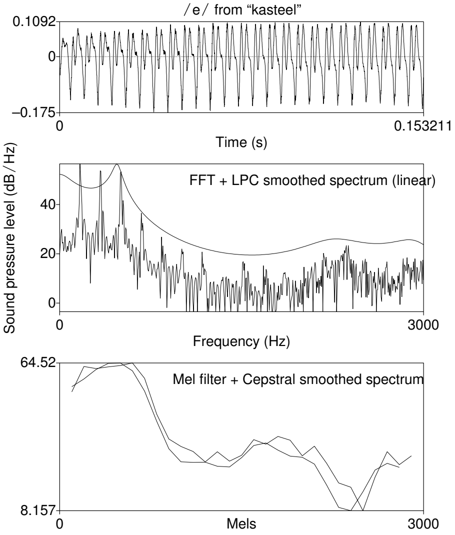
where

$$H\left(\mathfrak{G}\right) = \sum_{w_i} -P\left(w_i \middle| w_1 \ldots w_{i-1}\right) \cdot \log_2 P\left(w_i \middle| w_1 \ldots w_{i-1}\right)$$

For a tri-gram grammar this corresponds to:

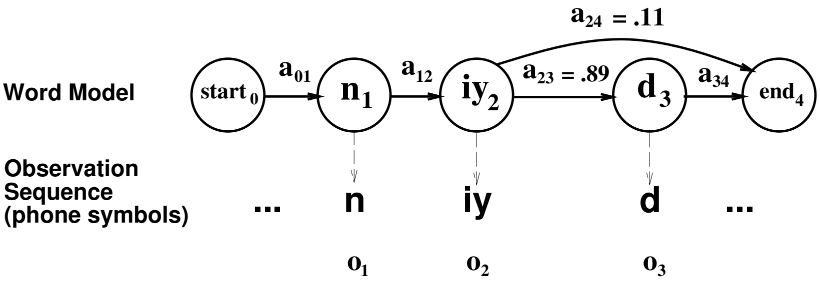
- $P\left(w_i \middle| w_{i-2}, w_{i-1}\right) = \dfrac{P\left(w_{i-2}, w_{i-1}, w_i\right)}{P\left(w_{i-2}, w_{i-1}\right)}$
- Perplexity corresponds to the difficulty of predicting the next word
- A lower perplexity is better for ASR

Spectral analysis: FFT, LPC, PLP, MFCC, filter-banks



- Need a spectral representation
- FFT: too noisy
- LPC: wrong sensitivity
- Resolution of the ear (Mel Freq, PLP, Filter banks)
- Sound level in dB (PLP, Filter banks)
- Spectral shape (MFCC)

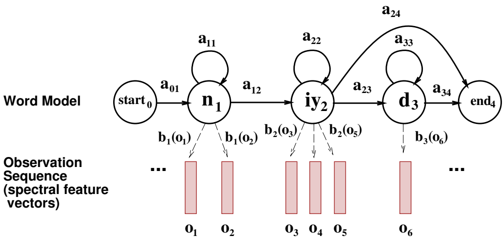
Hidden Markov Models: Markov chains



Word models: simple phone state model for *need*

- Each transition has a probability
- *start* and *end* are special states
- Each state *or* each transition has associated sound observations with a distinct probability density function (PDF)

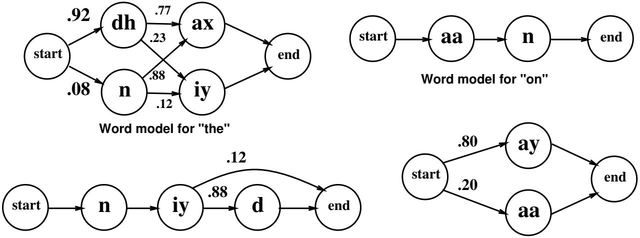
Hidden Markov Models: Observation probabilities



Observed are sound "spectra" for time "frames"

- Observation sequences have a probability
- Calculate this probability for each possible word
- Probabilities of O_i calculated from all possible underlying states
- Chose word *sequence* with the highest overall probability

Hidden Markov Models: Pronunciation networks



Word model for "the"

Word model for "on"

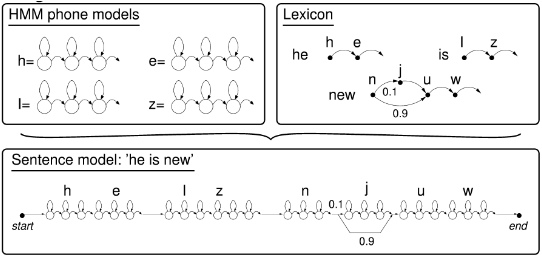
Word model for "need"

Word model for "I"

Construct phone state models for each word in the dictionary

- The possible pronunciations for each word have to be encoded in the dictionary
- The transition probabilities are "trained" from the frequency of occurrence of the pronunciation in the speech corpus

Hidden Markov Models: Phone networks



Phone models are concatenated into utterance networks

- Each word model is itself a Markov finite state network of phone models
- Phones and word are connected through the *start* and *end* states (not shown)

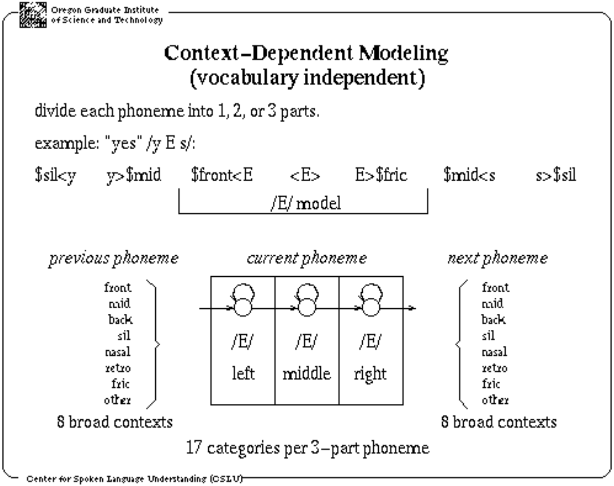
Hidden Markov Models: Context Sensitive Phone latices

Phone models are constructed of subphone states in context

- Each phone model is itself a Markov finite state network
- For each phoneme context separate phone models are constructed
- Each sub-phone context sensitive state can have it's own observation PDF
- For the sake of reducing training, the observation PDF's of different states are *tied* (i.e. made identical)

Hidden Markov Models: Context Sensitive Phone latices

[CSLU()]

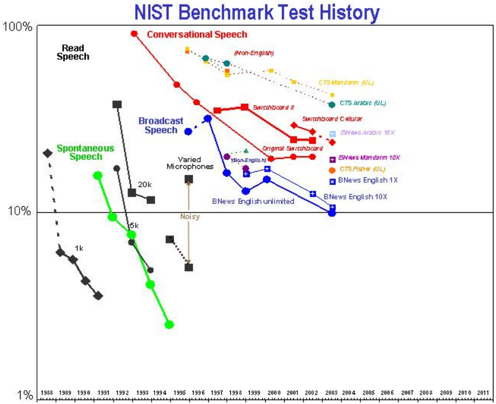


Evaluation: NIST, DARPA, hubs and spokes

The National Institute of Standards (NIST) and the DARPA program organize evaluation "contests" for ASR systems

- Tests contain mandatory core components *hubs*
- Tests contain optional specialized components *spokes*
- Tests evolve to include not only Speech-to-Text but also who spoke when, speaker localization etc.
- Includes varying speech material and conditions
- Contestants get training materials from the organization
- After time for training, contestants receive test speech and have to return the results

Evaluation: NIST results [Pallett(2003)]



- WER (vertical) go down over time
- More complex tasks introduced over time

Assignment: Week 7 Tone recognition

Recognize level and rising tones

- New → *Create PitchTier... level 0 0.6*
- Modify → *Add point... 0.05 200 & Add point... 0.55 200*
- New → *Create PitchTier... rising 0 0.6*
- Modify → *Add point... 0.05 100 & Add point... 0.55 200*
- Add silences to both PitchTiers: *Add point... 0.049 0 & Add point... 0.551 0*
- Select PitchTier <level|rising> → *To Pitch... 0.02 60 40*
- Select Pitch <either one> Play → Hum
- Record your voice imitating the pitch → Periodicity → *To Pitch... <default settings>*
- Select Pitch <either one> AND Pitch <your voice>
  → *To DTW... 24 10 yes yes no restriction*
- Select DTW dtw_level_rising → Query - → *Get distance (weighted)*
- Compare distances. How do you think you could improver recognition?
- See Blackboard for complete description.

Further Reading I

See chapter 7.1, 7.2, 7.5 [Jurafsky and Martin(2000)]

P. Boersma.
Praat, a system for doing phonetics by computer.
Glot International, 5:341–345, 2001.
URL http://www.Praat.org/.

P. Boersma and D. Weenink.
Praat 4.2: doing phonetics by computer.
Computer program: http://www.Praat.org/, 2004.
URL http://www.Praat.org/.

CSLU.
CSLU Toolkit.
Web.
URL http://cslu.cse.ogi.edu/toolkit/index.html.

FSF.
GNU General Public License.
Web, June 1991.
URL http://www.gnu.org/licenses/gpl.html.

Joshua T. Goodman.
A bit of progress in language modeling.
Computer Speech and Language, 15:403–434, 2001.
URL http://arxiv.org/abs/cs.CL/0108005.
URL is extended preprint.

Further Reading II

E. Gouvêa.
The CMU Sphinx Group Open Source Speech Recognition Engines.
Web.
URL http://cmusphinx.sourceforge.net/html/cmusphinx.php.

ISIP.
The Mississippi State ISIP public domain speech recognizer.
Web, August 2004.
URL http://www.cavs.msstate.edu/hse/ies/projects/speech/index.html.

Daniel Jurafsky and James H. Martin.
Speech and Language Processing.
Prentice-Hall, 2000.
ISBN 0-13-095069-6.
URL http://www.cs.colorado.edu/~martin/slp.html.
Updates at http://www.cs.colorado.edu/

Kevin Lenzo.
The CMU Pronouncing Dictionary.
Web.
URL http://www.speech.cs.cmu.edu/SLM_info.html.

David S. Pallett.
A look at NISTs benchmark asr tests: Past, present, and future.
In *Proceedings of the 2003 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2003.
URL http://www.nist.gov/speech/history/pdf/NIST_benchmark_ASRtests_2003.pdf.

Further Reading III

Project Gutenberg.
Project gutenberg free ebook library.
Web, 2005.
URL http://www.gutenberg.org/.

Roni Rosenfeld.
The CMU Statistical Language Modeling (SLM) Toolkit.
Web.
URL http://www.speech.cs.cmu.edu/SLM_info.html.

Rita Singh.
Robust group's open source tutorial learning to use the cmu sphinx automatic speech recognition system.
Web, 2005.
URL http://www.cs.cmu.edu/~robust/Tutorial/opensource.html.

Manual for the Sphinx-III recognition system.
SPHINX-CMU.
URL http://fife.speech.cs.cmu.edu/sphinxman/.

Paul A. Taylor, S. King, S. D. Isard, and H. Wright.
Intonation and dialogue context as constraints for speech recognition.
Language and Speech, 41:493–512, 1998.
URL http://www.cstr.ed.ac.uk/downloads/publications/1998/Taylor_1998_b.pdf.

Jean-Marc Valin.
Open mind speech.
Web.
URL http://freespeech.sourceforge.net/.

Further Reading IV

Xue Wang.
incorporating knowledge on segmental duration in hmm-based continuous speech recognition.
PhD thesis, LOT Netherlands Graduate School of Linguistics, 04 1997.
URL http://www.fon.hum.uva.nl/wang/ThesisWangXue/TOCINDEX.html.

Appendix A

Copyright License