CS 224S / LINGUIST 285
Spoken Language Processing

Andrew Maas
Stanford University

Lecture 1: Introduction, ARPAbet, Articulatory Phonetics

Original slides by Dan Jurafsky

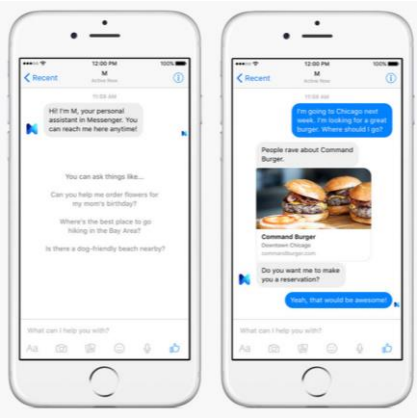
April 3, Week 1

- Course introduction
- Course topics overview
  - Speech recognition
  - Dialog / conversational agents
  - Speech synthesis (Text to speech)
  - Affect extraction
- Very brief history
- Articulatory Phonetics
- Course Logistics
- ARPAbet transcription

An exciting time for spoken language processing

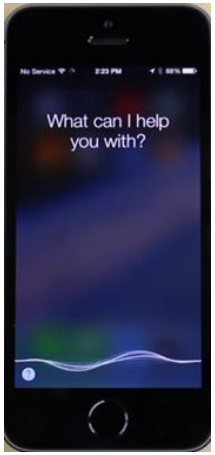
Amazon Echo
2015

Google Home
2016

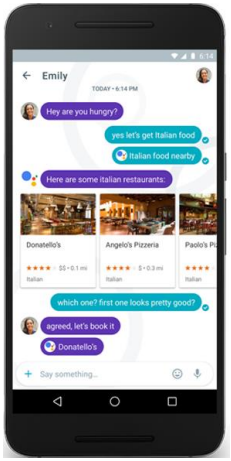


Facebook M
2015

Anki Cozmo
2016

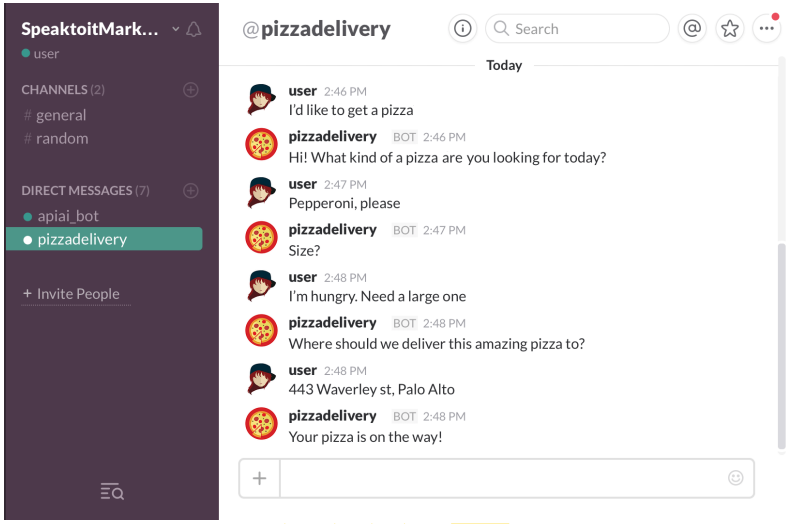


Apple
Siri
2011



Google
Assistant
2016

Microsoft
Cortana
2014



Slack Bot API
2015

LVCSR

- Large Vocabulary Continuous Speech Recognition
  - ~64,000 words
  - Speaker independent (vs. speaker-dependent)
  - Continuous speech (vs isolated-word)

Current error rates

Ballpark numbers; exact numbers depend very much on the specific corpus

Task	Vocabulary	Word Error Rate %
Digits	11	0.5
WSJ read speech	5K	1
WSJ read speech	20K	1
Broadcast news	64,000+	4
Conversational Telephone	64,000+	6

Why is conversational speech harder?

- A piece of an utterance without context

- The same utterance with more context

HSR versus ASR

Task	Vocab	ASR	Hum SR
Continuous digits	11	.5	.009
WSJ 1995 clean	5K	3	0.9
WSJ 1995 w/noise	5K	9	1.1
SWBD 2004	65K	~6	3-4?

- Conclusions:
  - Gap increases with noisy speech
  - These numbers are rough, take with grain of salt
  - We are overfitting to the benchmark datasets

HSR versus ASR

Deletions				Insertions			
SWB		CH		SWB		CH	
ASR	Human	ASR	Human	ASR	Human	ASR	Human
30: it	19: i	46: i	20: i	13: i	16: is	23: a	17: is
20: i	17: it	46: it	18: and	10: a	14: %hes	14: is	17: it
17: that	16: and	39: and	15: it	7: and	12: i	11: i	16: and
16: a	14: that	32: is	15: the	7: of	11: and	10: are	14: have
14: and	14: you	26: oh	14: is	6: you	9: it	10: you	13: a
14: oh	12: is	25: a	13: not	5: do	6: do	9: the	13: that
14: you	12: the	20: to	10: a	5: the	5: have	8: have	12: i
12: %bcack	11: a	19: that	10: in	5: yeah	5: yeah	8: that	11: %hes
12: the	10: of	19: the	10: that	4: air	5: you	7: and	10: not
11: to	9: have	18: %bcack	10: to	4: in	4: are	7: it	9: oh

Table 3: Most frequent deletion and insertion errors for humans and ASR system on SWB and CH.

SWB		CH	
ASR	Human	ASR	Human
11: and / in	16: (%hes) / oh	21: was / is	28: (%hes) / oh
9: was / is	12: was / is	16: him / them	22: was / is
7: it / that	7: (i-) / %hes	15: in / and	11: (%hes) / %bcack
6: (%hes) / oh	5: (%hes) / a	8: a / the	10: bentsy / benji
6: him / them	5: (%hes) / hmm	8: and / in	10: yeah / yep
6: too / to	5: (a-) / %hes	8: is / was	9: a / the
5: (%hes) / i	5: could / can	8: two / to	8: is / was
5: then / and	5: that / it	7: the / a	7: (%hes) / a
4: (%hes) / %bcack	4: %bcack / oh	7: too / to	7: the / a
4: (%hes) / am	4: and / in	6: (%hes) / a	7: well / oh

Table 2: Most frequent substitution errors for humans and ASR system on SWB and CH.

(Saon et al, 2017)

Why accents are hard

- A word by itself

- The word in context

So is speech recognition solved? Why study it vs just use some API?

- In the last ~5 years
  - Dramatic reduction in LVCSR error rates (16% to 6%)
  - Human level LVCSR performance on Switchboard
  - New class of recognizers (end to end neural network)
- Understanding how ASR works enables better ASR-enabled systems
  - What types of errors are easy to correct?
  - How can a downstream system make use of uncertain outputs?
  - How much would building our own improve on an API?
- Next generation of ASR challenges as systems go live on phones and in homes

Speech Recognition Design Intuition

- Build a statistical model of the speech-to-words process
- Collect lots and lots of speech, and transcribe all the words.
- Train the model on the labeled speech
- Paradigm: Supervised Machine Learning + Search

Dialogue (= Conversational Agents)

- Personal Assistants
  - Apple SIRI
  - Microsoft Cortana
  - Google Assitant
- Design considerations
  - Synchronous or asynchronous tasks
  - Pure speech, pure text, UI hybrids
  - Functionality versus personality

Paradigms for Dialogue

- **POMDP**
  - Partially-Observed Markov Decision Processes
  - Reinforcement Learning to learn what action to take
  - Asking a question or answering one are just actions
    - "Speech acts"
- **Simple regular expressions and slot filling**
  - Pre-built frames
    - Calendar
      - Who
      - When
      - Where
  - Filled by hand-built rules
    - ("`on (Mon|Tue|Wed…)`")

Paradigms for Dialogue

- **POMDP**
  - Exciting Research
  - Implemented in no commercial systems
- **Simple regular expressions and slot filling**
  - State of the art used most systems
- **Reusing new search engine technology**
  - Intent recognition / semantic parsing
- **Neural network chatbots**
  - Recent research, not really dialog yet

Extraction of Social Meaning from Speech

- Detection of student uncertainty in tutoring
  - Forbes-Riley et al. (2008)
- Emotion detection (annoyance)
  - Ang et al. (2002)
- Detection of deception
  - Newman et al. (2003)
- Detection of charisma
  - Rosenberg and Hirschberg (2005)
- Speaker stress, trauma
  - Rude et al. (2004), Pennebaker and Lay (2002)

Conversational style

- Given speech and text from a conversation
- Can we tell if a speaker is
  - Awkward?
  - Flirtatious?
  - Friendly?

- Dataset:
  - 1000 4-minute “speed-dates”
  - Each subject rated their partner for these styles
  - The following segment has been lightly signal-processed:

Speaker Recognition tasks

- Speaker Recognition
  - Speaker Verification (Speaker Detection)
    - Is this speech sample from a particular speaker

Is that Jane?

  - Speaker Identification
    - Which of these speakers does this sample come from?

Who is that?

    - Related tasks: Gender ID, Language ID

Is this a woman or a man?

- Speaker Diarization
  - Segmenting a dialogue or multiparty conversation

Who spoke when?

Applications of Speaker Recognition

- Speaker Recognition:
  - Speaker verification (binary decision)
    - Voice password
    - Telephone assistant
  - Speaker identification (one of N)
    - Criminal investigation
- Diarization
  - Transcribing meetings

TTS (= Text-to-Speech) (= Speech Synthesis)

- Produce speech from a text input
- Applications:
  - Personal Assistants
    - Apple SIRI
    - Microsoft Cortana
    - Google Assistant
  - Games
  - Airport Announcements

TTS Overview

- Main Commercial Algorithm
  - Google TTS
- Collect lots of speech (5-50 hours) from one speaker, transcribe very carefully, all the syllables and phones and whatnot
- To synthesize a sentence, patch together syllables and phones from the training data.
- Parametric synthesis shows recent gains
- First end to end neural systems in 2016

History: foundational insights 1900s-1950s

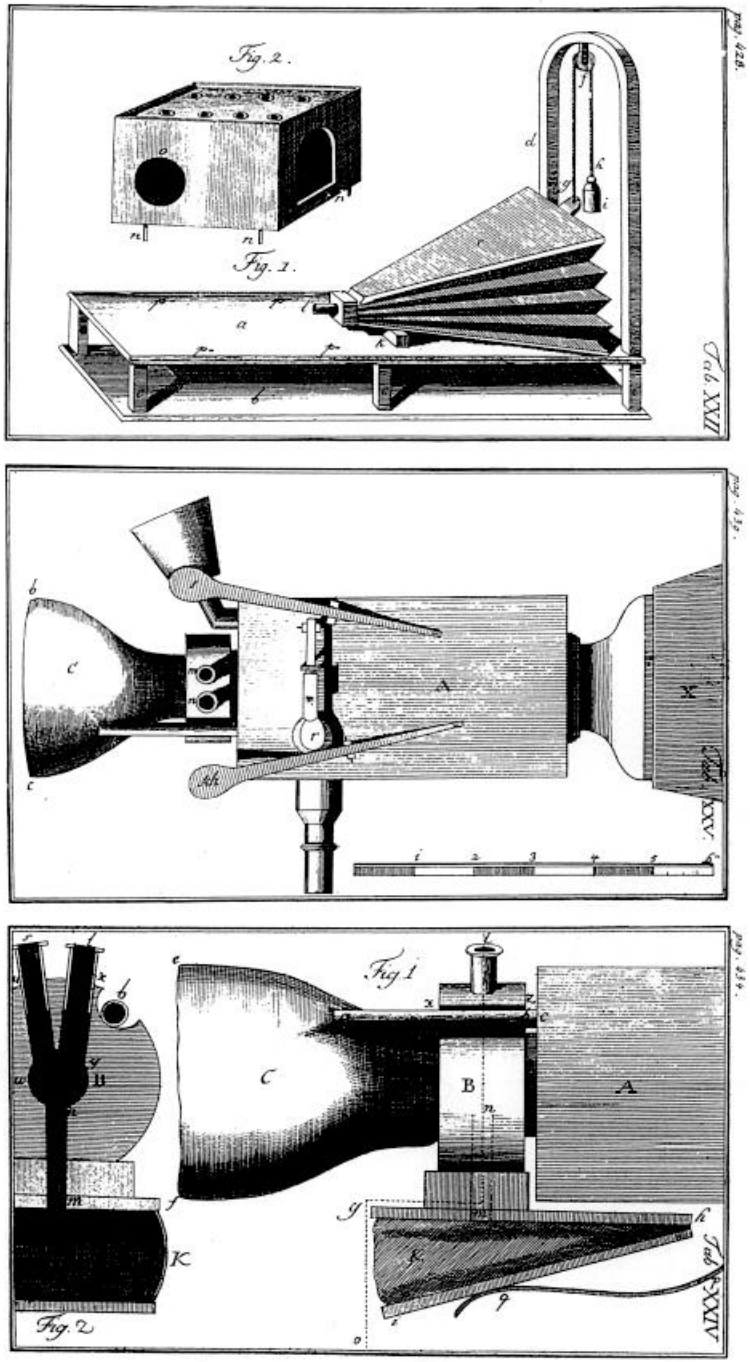
- Automaton:
  - Markov 1911
  - Turing 1936
  - McCulloch-Pitts neuron (1943)
    - http://marr.bsee.swin.edu.au/~dtl/het704/lecture10/ann/node1.html
    - http://diwww.epfl.ch/mantra/tutorial/english/mcpits/html/
  - Shannon (1948) link between automata and Markov models
- Human speech processing
  - Fletcher at Bell Labs (1920's)
- Probabilistic/Information-theoretic models
  - Shannon (1948)

Speech synthesis is old!

- Pictures and some text from Hartmut Traunmüller's web site:
  - http://www.ling.su.se/staff/hartmut/kemplne.htm
- Von Kempeln 1780 b. Bratislava 1734 d. Vienna 1804
- Leather resonator manipulated by the operator to try and copy vocal tract configuration during sonorants (vowels, glides, nasals)
- Bellows provided air stream, counterweight provided inhalation
- Vibrating reed produced periodic pressure wave

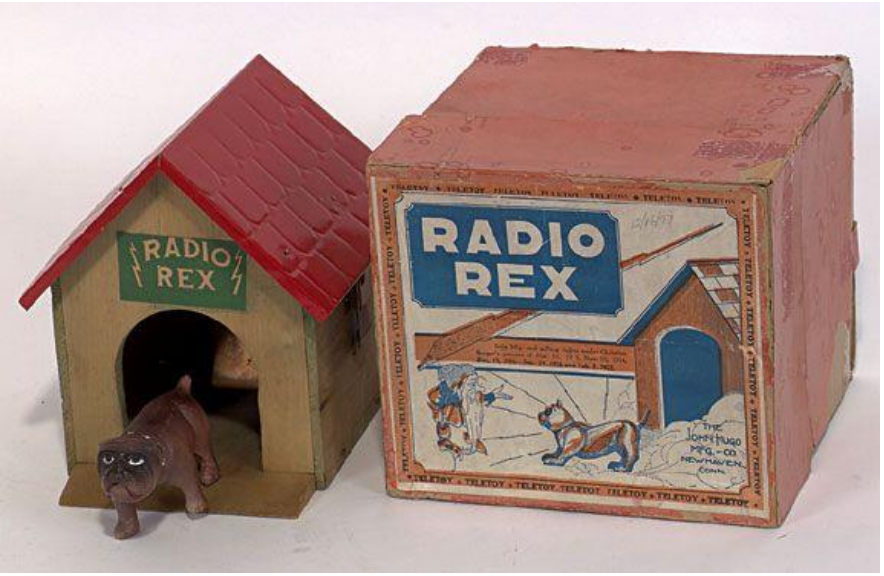
Von Kempelen:

- Small whistles controlled consonants
- Rubber mouth and nose; nose had to be covered with two fingers for non-nasals
- Unvoiced sounds: mouth covered, auxiliary bellows driven by string provides puff of air



From Traunmüller's web site

History: Early Recognition



- 1920's Radio Rex
  - Celluloid dog with iron base held within house by electromagnet against force of spring
  - Current to magnet flowed through bridge which was sensitive to energy at 500 Hz
  - 500 Hz energy caused bridge to vibrate, interrupting current, making dog spring forward
  - The sound "e" (ARPAbet [eh]) in Rex has 500 Hz component

History: early ASR systems

- 1950's: Early Speech recognizers
  - 1952: Bell Labs single-speaker digit recognizer
    - Measured energy from two bands (formants)
    - Built with analog electrical components
    - 2% error rate for single speaker, isolated digits
  - 1958: Dudley built classifier that used continuous spectrum rather than just formants
  - 1959: Denes ASR combining grammar and acoustic probability

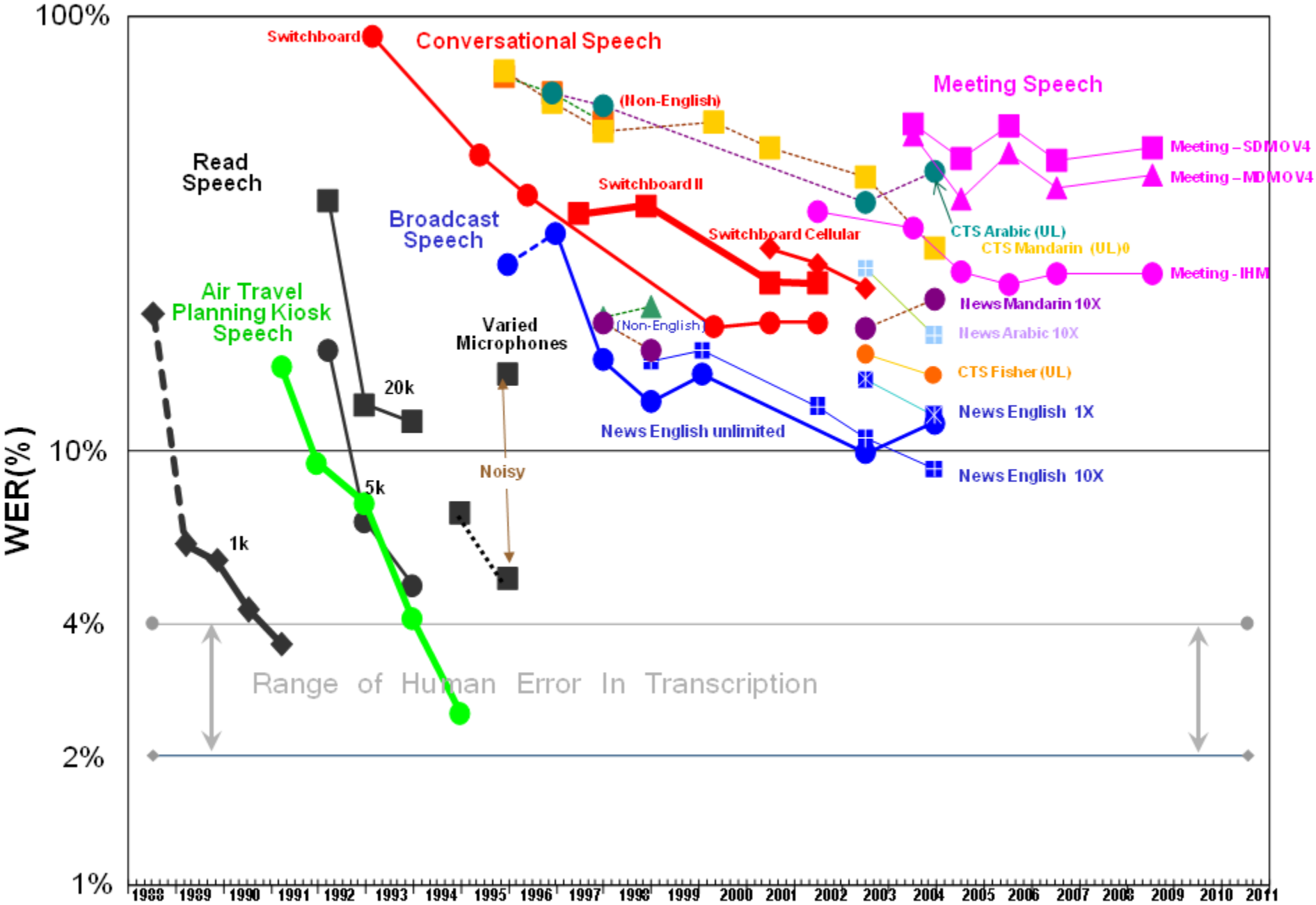
History: early ASR systems

- 1960's
  - FFT - Fast Fourier transform (Cooley and Tukey 1965)
  - LPC - linear prediction (1968)
  - 1969 John Pierce letter "Whither Speech Recognition?"
    - Random tuning of parameters,
    - Lack of scientific rigor, no evaluation metrics
    - Need to rely on higher level knowledge

ASR: 1970's and 1980's

- Hidden Markov Model 1972
  - Independent application of Baker (CMU) and Jelinek/Bahl/Mercer lab (IBM) following work of Baum and colleagues at IDA
- ARPA project 1971-1976
  - 5-year speech understanding project: 1000 word vocab, continous speech, multi-speaker
  - SDC, CMU, BBN
  - Only 1 CMU system achieved goal
- 1980's+
  - Annual ARPA "Bakeoffs"
  - Large corpus collection
    - TIMIT
    - Resource Management
    - Wall Street Journal

NIST STT Benchmark Test History – May. '09



Course Logistics

Course Logistics

- http://www.stanford.edu/class/cs224s
- Homeworks released and due on Wednesdays
- Gradescope for homework submission
- Piazza for questions. Email staff only for personal/confidential questions
- Project poster session tentatively June 7 (during class time)

Admin: Requirements and Grading

- Readings:
  - Selected chapters from
    - Jurafsky & Martin. Speech and Language Processing.
      - Will mix chapters from 2nd and in progress 3rd editions
    - A few conference and journal papers
- Grading
  - Homework: 40%
    - 4 assignments. Will use Python, Tensorflow, and command line tools
  - Course Project: 50%
    - Group projects of 3 people
  - Participation: 10%

Necessary Background

- Foundations of machine learning and natural language processing
  - CS 124, CS 224N, CS 229, or equivalent experience
- Mathematical foundations of neural networks
  - Understand forward and back propagation in terms of equations
- Proficiency in Python
  - Programming heavy homeworks will use Python and Tensorflow