Spoken Human Robot Interaction

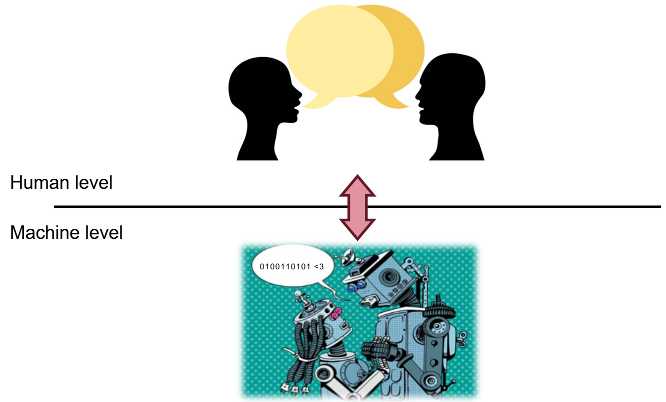
Automatic Speech Recognition

SAPIENZA
UNIVERSITÀ DI ROMA

Andrea Vanzo, **Daniele Nardi**
lastname@dis.uniroma1.it
Department of Computer, Control, and Management Engineering
"Sapienza" University of Rome, Italy

Artificial Intelligence
AY 2018/19

Communication



Advantages of Spoken Language

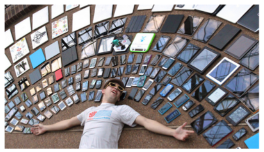
- **Natural**: Requires no special training
- **Flexible**: Leaves hands and eyes free
- **Efficient**: Has high data rate
- **Economical**: Can be transmitted/received inexpensively

Many many applications

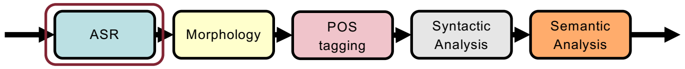

Human-Robot Interaction


Hands-free assistants


Home devices


Mobile devices

Our Command Interpretation pipeline



The ASR is the process that generates a “text”, starting from an audio signal

Issues:

- “recognize speech” vs “wreck a nice beach”
  – Segmentation (missing spaces)
  – Coarticulation (merging sounds)
  – Homophones (e.g. to too two)

Outline

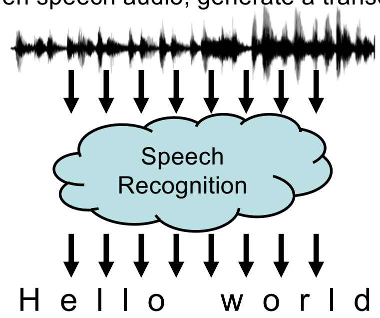
- Classical approaches to ASR
  – Hidden Markov Model (HMM)
- Deep Learning for ASR
  – Connectionist Temporal Classification (CTC)
- Evaluation Metrics
  – Word Error Rate (WER)

Speech Processing

- Many tasks involved
  – Speech transcription
  – Word spotting/trigger word
  – Speaker identification/verification
  – Localizing sound sources
  – …

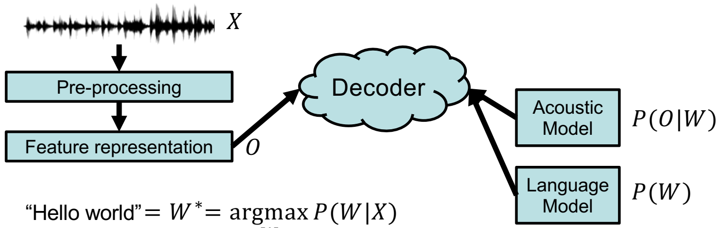
Speech Recognition

- Given speech audio, generate a transcript



Basic ASR architecture

- ASRs break the problem into several components



"Hello world" $= W^* = \underset{W}{\operatorname{argmax}}\, P(W|X)$

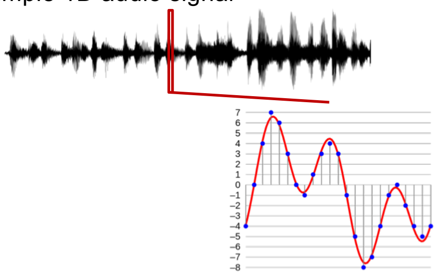
$= \underset{W}{\operatorname{argmax}}\, P(O|W)P(W)$

Pre-processing

- First step: feed sound waves into a computer
- How do we turn sound waves into numbers?

Analog-To-Digital

- Input: simple 1D audio signal



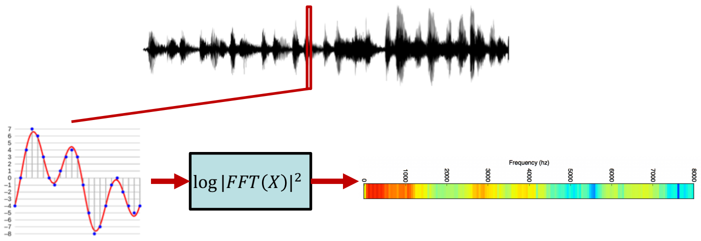
- Output: 1D vector $X = [x_1, x_2, \dots]$

Features of speech signal

- Most frequencies in the range 100 –1000 Hz
- Typical sample rates are 8KHz or 16KHz
  - 8K (16K) samples per second
  - CD and MP3 files are sampled at 44.1KHz
- Quantization (8-12 bits to record amplitude)

Frequency Representation

- Take a small window (e.g. 20ms) of waveform
  - Compute FFT and take magnitudes (i.e., power)
  - Show frequency content in local window



Words to phonemes

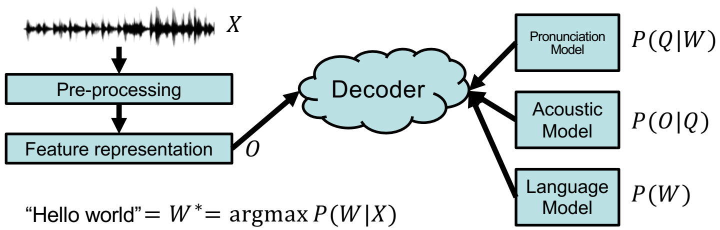
- Words are usually represented as sequences of phonemes

  w_1 = "hello" = [HH AH L OW] = $[q_1q_2q_3q_4]$
- Phonemes are the perceptually distinct units of sound that distinguish words (in a language)
  - Rather approximate, but sorta recognized by the community
  - Corpora available (e.g., TIMIT)

Phoneme	*Example*	Phoneme	*Example*
ch	***ch**oke*	b	***b**ee*
en	*butt**on***	eng	*Washi**ng**ton*

Refined ASR architecture

- ASRs usually model phonemes instead of words → additional resource required

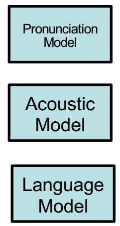


"Hello world" $= W^* = \underset{W}{\operatorname{argmax}} P(W|X)$

$$= \underset{W}{\operatorname{argmax}} \sum_Q P(O|Q)P(Q|W)P(W)$$

Classical approaches

- Additional pre-processing to extract an effective set of feature (based on frequencies and frequence differences)
- Approaches :
  - Hidden Markov Model (HMM)
    - Gaussian Mixture Model (GMM)
    - Support Vector Machines (SVM)
    - Artificial Neural Networks (ANN)



Hidden Markov Model for ASR

Probably the most used approach for ASR (since 70ies)

- Language Model: *P(W)*
  - Costruction
  - Evaluation
- Acoustic *P(Q|W)*/ Pronunciation *P(O|Q)* Models
  - Decoding
  - Training

Language Model: Construction

Language Model $P(W)$

Which sequences of characters (words) are more likely?

- An *n*-gram model is a Markov Chain of order *n-1* (unigram, bigram, trigram …)
- Trigram: P(c_i| $c_{1:i-1}$)= P(c_i| $c_{i-2:i-1}$)

Built from corpora (specific for spoken language)

Used for language identification, spelling correction, genre classification, Name-Entity recognition, …

Language model: Evaluation

Language Model $P(W)$

- Determine the probability of the model in generating the sequence $W = (w_1, \dots, w_T)$ given a HMM *model* λ is:

$$P(W|\lambda) = \sum_{\forall S} P(W, S|\lambda)$$

where $S = s_1, \dots, s_T$ is a state sequence

- Not feasible: search space is huge ($O(N^T)$)
- Solution: Forward algorithm (dynamic programming)

Hidden Markov Model for ASR: Decoding

Pronunciation Model $P(Q|W)$ Acoustic Model $P(O|Q)$

- Given a sequence of symbols W (or Q) and a model γ (or φ), what is the most likely sequence of states Q (or O) that produced the sequence

$$Q^* = \operatorname*{argmax}_Q P(Q|W,\gamma) = \operatorname*{argmax}_Q P(Q,W|\gamma)$$

(or $O^* = \operatorname*{argmax}_Q P(O|Q,\varphi) = \operatorname*{argmax}_O P(O,Q|\varphi)$)

- Not feasible: search space is huge
- Viterbi algorithm (dynamic programming)

Hidden Markov Model for ASR: Training

Given a model structure and a set of sequences, find the model that best fits the data

- No efficient algorithm for global optimum
- Efficient iterative algorithm finds local optima

However…

- Classical architecture is highly tweak-able, but also hard to get working well
- Historically, each part of the architecture has its own set of challenges
  - Feature representation/extraction
  - Decoding algorithm
  - …

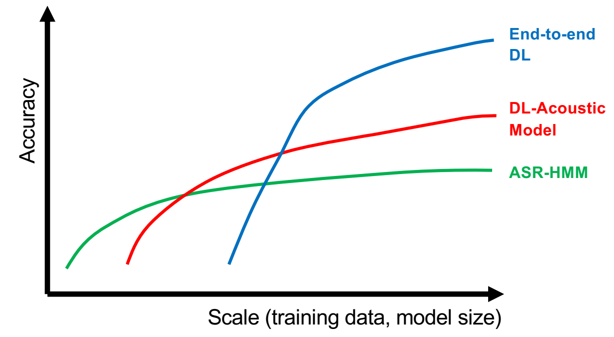
Deep Learning in ASR

Acoustic Model $P(O|Q)$

- How to apply DL to make ASR better?
  - First attempt: improve the acoustic model
- Deep Belief Networks (DBNs)
  - Probabilistic generative models
  - Composed of multiple layers of stochastic, latent variables
  - Latent variables typically have binary values (*hidden units* or *feature detectors*)

Deep Learning as alternative to HMM in ASR

- Can we do better?

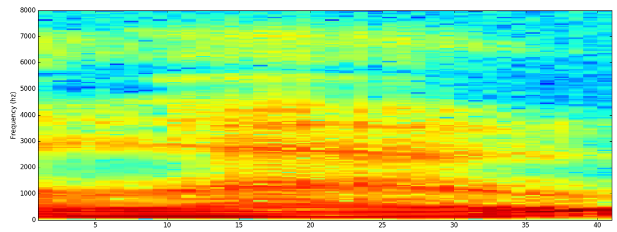


Deep Learning as alternative to HMM in ASR

- An end-to-end DL-based architecture for SR
  - Feature extraction
  - Connectionist Temporal Classification
  - Training
  - Decoding and Language model

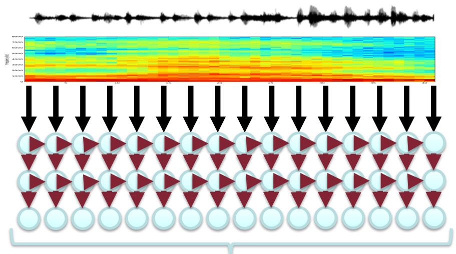
Spectrogram

Concatenation of frames from adjacent windows



Deep Learning ASR

- Goal: create a neural network (DNN/RNN) from which we can extract a transcription W^*
  - Train from labeled pairs (X, W^*)



W^* =Hello world

Deep Learning ASR

- Main issue: $length(X) \neq length(W^*)$
  - Don't know how symbols in W map to frames of audio
- Multiple ways to solve
  - Attention
  - Sequence to sequence models
  - Connectionist Temporal Classification

Connectionist Temporal Classification

RNN output neurons *c* encode distribution over symbols. In this case, $length(c) = length(X)$

Phoneme-based model $c \in \{AA, AE, AX, \dots, blank\}$

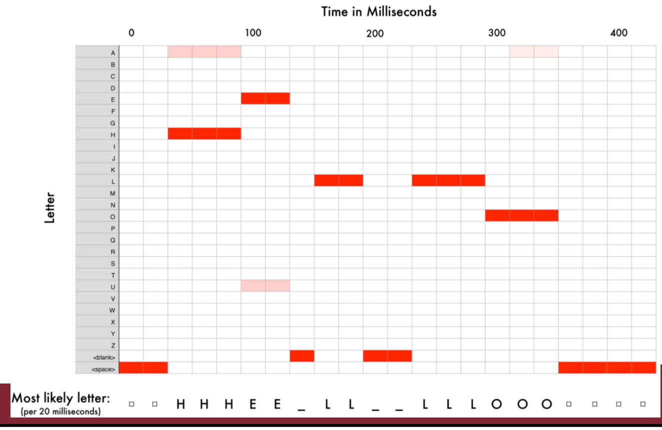
Grapheme-based model $c \in \{A, B, C, \dots, blank, space\}$

Define a mapping $\beta(c) \to W$

Maximize the likelihood of W^* under this model

Connectionist Temporal Classification

CTC-NN computes most likely spoken letters from audio



Connectionist Temporal Classification

NN predicts the following transcriptions:
"HHHEE_LL_LLLOOO"
(but also "HHHUU_LL_LLLOOO" or "AAAUU_LL_LLLOOO")

Post-processing cleans the output

- Replace any repeated char with single one
  - HHHEE_LL_LLLOOO becomes HE_L_LO
  - HHHUU_LL_LLLOOO becomes HU_L_LO
  - AAAUU_LL_LLLOOO becomes AU_L_LO
- Remove any blanks
  - HE_L_LO becomes HELLO
  - HU_L_LO becomes HULLO
  - AU_L_LO becomes AULLO

Connectionist Temporal Classification

Last, we choose the most likely one according to likelihood scores based on large text corpus:

"Hello" appears more frequently than "Hullo" and "Aullo"

Notice:

- almost impossible to recognize "Hullo" if we say it.
- Almost impossible to build your own ASR

Evaluating ASRs

How to evaluate the "goodness" of a word string output by a speech recognizer?

Terms:
- ASR hypothesis: ASR output
- Reference transcription: ground truth – what was actually said

Transcription Accuracy

Word Error Rate (WER)

- **Minimum Edit Distance**: Distance in words between the ASR hypothesis and the reference transcription
  - Edit Distance = (Substitutions+Insertions+Deletions)/N
  - For ASR, usually all weighted equally (different weights can be used to model different types of errors)
- WER = Edit Distance * 100

Word Error Rate: Example

REF: portable **** PHONE UPSTAIRS last night so
HYP: portable FORM OF STORES last night so
Eval I S S
WER = 100 x (1+2+0)/6 = 50%

Word Error Rate – character level

- One might compute the Word Error Rate at character level
- Insertions, Substitution and Deletions are computed looking at the single symbol

REF: portable **** PHONE UPSTAIRS last night so
HYP: portable FORM OF STORES last night so
Eval I S S
WER = 100 x (5+3+5)/36 = 36.1%

ASR – off-the-shelf solutions

hello

Hi. I'm Cortana.
Ask me a question!

Siri
Speech Interpretation and Recognition Interface

NUANCE

amazon alexa

CMU Sphinx

References

Basic:

[RN] Speech Recognition Sec. 23.5, Language Models Sec. 22.1

Speech Recognition with Deep Learning. Lecture by Adam Coates (at Baidu): https://goo.gl/upKcmR

Additional:

Graves, Fernandez, Gomez, Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.6306&rep=rep1&type=pdf

Padmanabhan, Premkumar. 2015. Machine Learning in Automatic Speech Recognition: A Survey.
http://www.tandfonline.com/doi/pdf/10.1080/02564602.2015.1010611?needAccess=true