

AUTOMATIC SPEECH RECOGNITION

&

TEXT
TO
SPEECH

Vladislav Maraev
11.02.2021

Part 1:

ASR Fundamentals

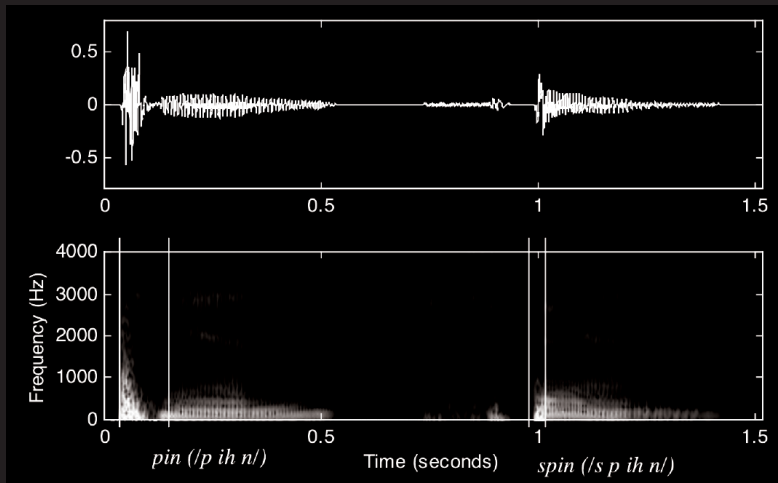
ASR types

- fixed commands
- grammars (e.g. date)
- continuous speech recognition (huge vocabulary)

What is ASR

- The aim is to decode the acoustic signal X into a word sequence \hat{W} which is hopefully close to original word sequence W .
- We need an **acoustic model**: our knowledge about acoustics, phonetics, microphone, environment, speakers, dialects, etc.
- And a **language model** that knows about words (and non-words), how they co-occur and form sequences (utterances)

Challenge: context variability



Variability in speech

- **context variability** (*pin vs spin*),
coarticulation
- **style variability** (*"Ford or" vs "Four door"*),
easier in isolation
- **speaker variability** (*need more than 500 speakers for speaker-independent ASR*)
- **environment variability**

Formal definition

Formal definition

- We can observe the set of parameters O (observations) for the acoustic signal.
- In the following equation acoustic model would be responsible for $P(O|W)$ and language model for $P(W)$

$$W = \arg \max_W P(W|O) = \arg \max_W \frac{P(O|W)P(W)}{P(O)} = \arg \max_W P(O|W)P(W)$$

But how do we calculate $P(O|W)$?

- Can we use words?

But how do we calculate $P(O|W)$?

- Can we use words? Yes, but then there will be a huge range of observations.
- We can minimise the number of possible observations? Yes, by selecting some smaller phonetic units.

Why word is not a good unit:

- New task can have new words but we won't have any training data.
- There are too many words, and each of them has too many acoustic realisations.

Good unit is:

- **accurate**, in order to represent acoustic realisation in different contexts
- **trainable**, we should have enough data to estimate the parameters of the unit
- **generalisable**, so new words could be derived from our units

What about phonemes?

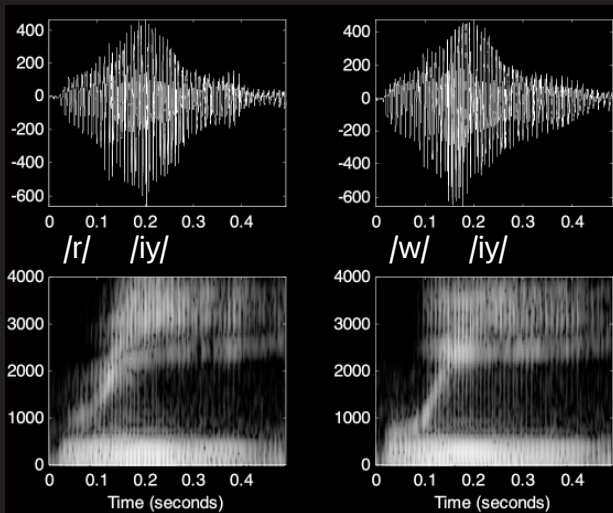
What about phonemes?

- Just 50 phones in English, no problem to train. And they are vocabulary-independent.
- But phonemes are not produced independently, they are context-dependent, so they can over-generalise.
- For some languages we can use syllables (1200 in Chinese, 50 in Japanese) but for English (30k+) training is challenging.

How can we capture context dependency?

- A triphone phonetic model: it takes into consideration the left and the right contexts.
- They capture coarticulation of a phone in all possible contexts.

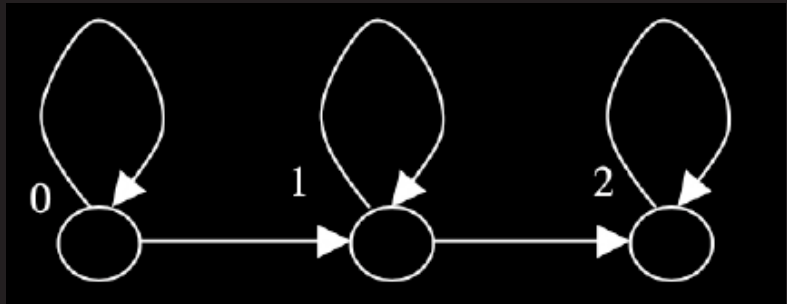
Different but similar



Different but similar

- It is desirable to find instances of similar contexts and merge them.
- This would lead to a much more manageable number of models that can be trained.
- We can use phonetic Hidden Markov Models (HMM) as the basic subphonetic unit.

HMM clustering



*transition from
previous to
current phoneme*

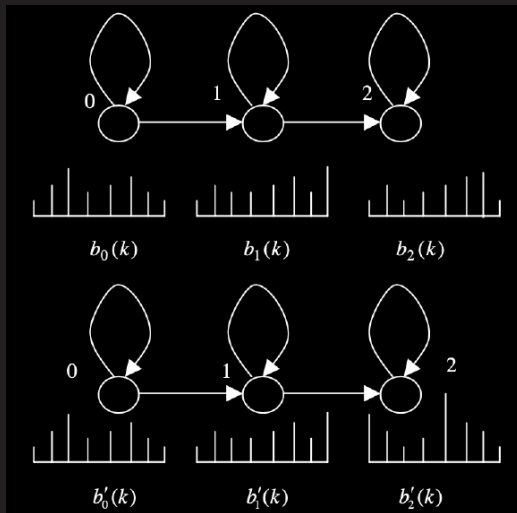
*middle section
of current
phoneme*

*transition out
of current
phoneme*

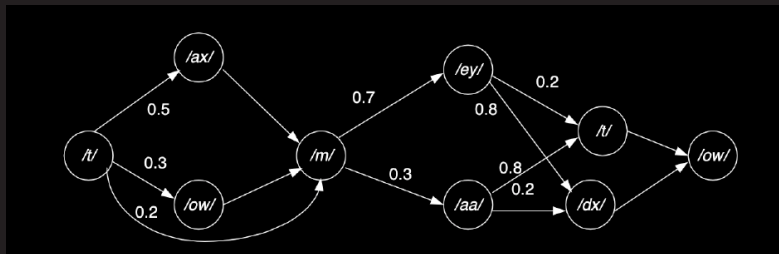
Phonetic HMMs

- Another challenge: variable length of observations for each phoneme.
- HMM can describe how states (set of features) are distributed in the phoneme.
- We observe only the acoustic features, but the states are hidden.

HMM clustering

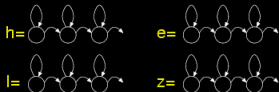


Word modelling

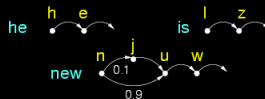


From phonemes to sentences

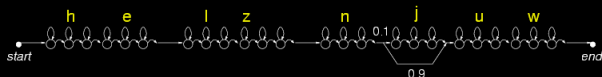
HMM phone models



Lexicon

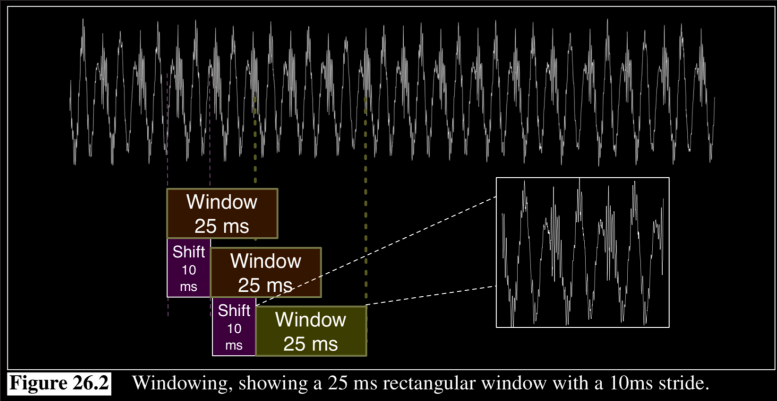


Sentence model: 'he is new'



Feature extraction

Feature extraction I: windowing



Hamming window

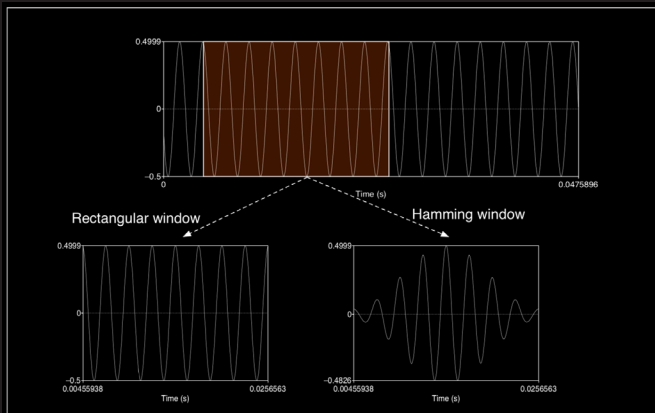


Figure 26.3 Windowing a sine wave with the rectangular or Hamming windows.

Discrete Fourier Transform (DFT)

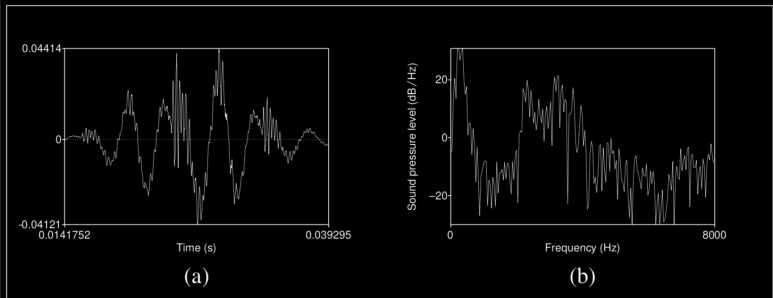


Figure 26.4 (a) A 25 ms Hamming-windowed portion of a signal from the vowel [iy] and (b) its spectrum computed by a DFT.

Mel filter bank

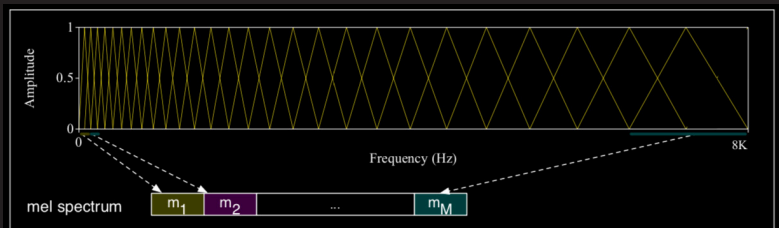
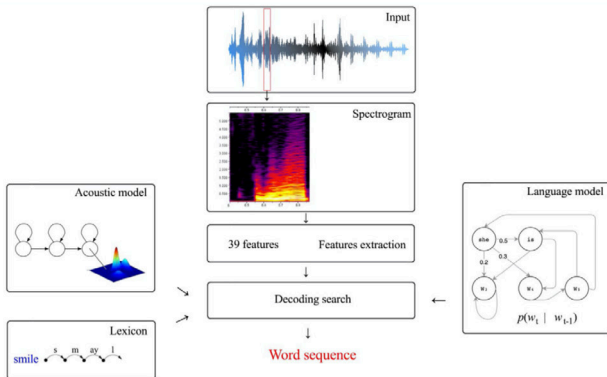


Figure 26.5 The mel filter bank (Davis and Mermelstein, 1980). Each triangular filter, spaced logarithmically along the mel scale, collects energy from a given frequency range.

ASR architecture



$$\mathbf{W}^* = \arg \max_{\mathbf{W}} P(\mathbf{W} | \mathbf{X})$$

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} p(\mathbf{X} | \mathbf{W}) P(\mathbf{W})$$

word sequence acoustic model language model

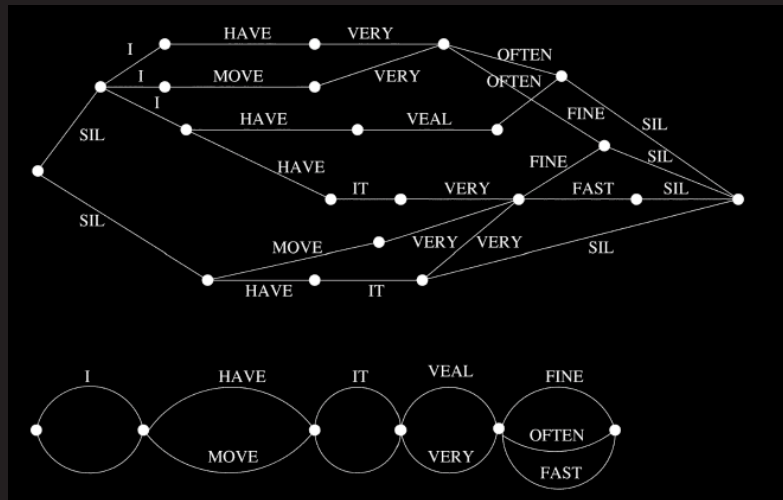
ASR architecture

- frames (width: 25ms, shift: 10ms), feature extraction, Hamming window
- Mel-frequency cepstral coefficients (MFCCs) are used as an input to individual Gaussian distribution for each phone
- models (e.g. HMM model of a triphone)
- decoding is integral path: the process which transforms the signal into word hypotheses.

ASR decoding:

- Using Viterbi or other methods, multiple 'top' hypotheses can remain
- The possible outcomes can be stored in a word-confusion network (sausage) or a lattice.

Word lattice and confusion network



ASR evaluation

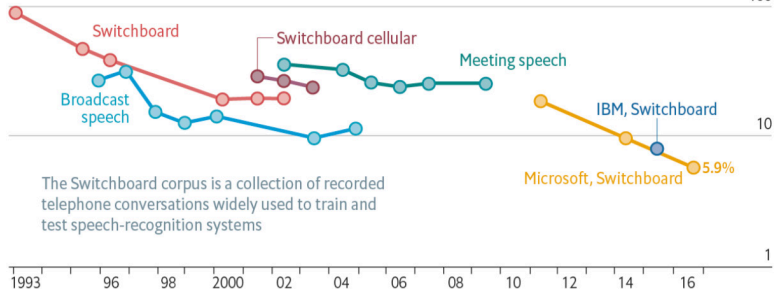
- Levenshtein distance/alignment
- Word-error rate (WER)

$$WER = (I + S + D) / N$$

Loud and clear

Speech-recognition word-error rate, selected benchmarks, %

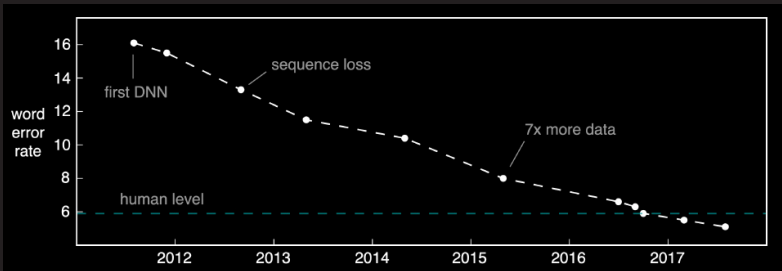
Log scale
100



The Switchboard corpus is a collection of recorded telephone conversations widely used to train and test speech-recognition systems

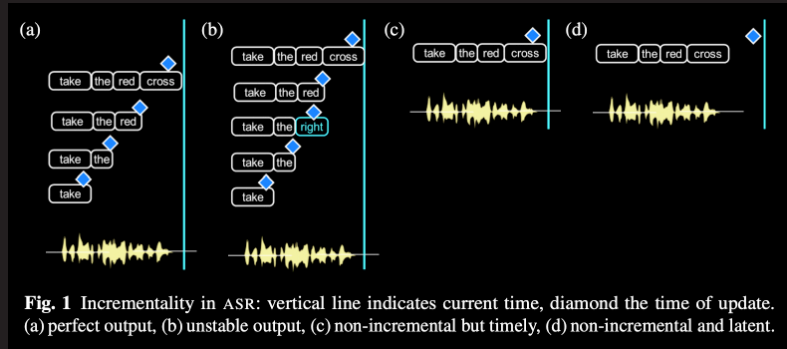
Sources: Microsoft; research papers

The age of neural networks



Incremental ASR

Incrementality in ASR



Incrementalising ASR

- We need incremental ASR, which would require incrementalising all the internal processes.
- We need incremental LMs and AMs, in order to get the best sequences unit-by-unit.
- It is critical that ASR will provide timing for recognised words in a timely manner.

ASR should support disfluencies

*"have the engine [take the oranges to Elmira,
+ { um, I mean, } take them to Corning] "* (Core
and Schubert 1999)

- 'um' and 'uh' should be considered English words
- it is important to have access to "the oranges" in ASR output
- incremental disfluency detector (Hough and Purver, 2014)

Incremental evaluation I *(Baumann et al., 2017)*

Utterance-level Accuracy and Disfluency Suitability

- *WER disfluency gain to determine how much of disfluent material is recovered*

Incremental evaluation II

Timing: first occurrence (FO) and final decision (FD):

- FO is the time between the (true) beginning of a word and the first time it occurs in the output (regardless if it is afterwards changed)
- FD is the time between the (true) end of a word and the time when the recognizer decides on the word, without later revising it anymore.

Incremental evaluation III

Diachronic Evolution: how often consuming processors have to re-consider their output and for how long hypotheses are likely to still change.

- **stability of the hypotheses.** For words that are added and later revoked or substituted we measure the “survival time” and report aggregated plots of word survival rate (WSR) after a certain age.

Systems

- Off-the-shelf web-based ASR (e.g. Google Speech API) has improved to a very good overall accuracy level (particularly its acoustic model).
- However, training your ASR for your language/domain to improve accuracy can reap rewards (Sphinx, Kaldi).

Part 1:

TTS Fundamentals

TTS challenges

- coarticulation
- Text normalisation
- Homography
- Code-switching (and borrowed words)
- Morphology

Text normalisation

- abbreviations and acronyms: *Dr.*, *DC*, *NASA*, *COVID-19*
- number formats, e.g. *IBM 370*, dates, times and currencies
- ~ *Ü * " UPPER CASE :-)*

Homograph normalisation

- Homograph variation can often be resolved on PoS (grammatical) category, e.g. *object*, *bass*, *absent*, *-ate*.
- But sometimes PoS does not help, e.g. *read*, *kinda*
- Variation of dialects
- Rate of speech (e.g. 'g' in *recognise*)

What is TTS

- text and phonetic analysis,
grapheme-to-phoneme, morphology
- prosody generation/modification
- speech synthesis

ALGORITHM 14.4 MORPHOLOGICAL ANALYSIS

1. Dictionary Lookup

Look up word w in lexicon

If found

Output attributes of the found lexical entry and exit

2. Suffix Stripping

If word ends in $-s$, $-s$'s, $-ing$, $-ed$, $-est$, $-ment$, etc

Strip the suffix from word w to form u

If stripped form u found in lexicon

Output attributes of the stem and suffix and exit

3. Prefix Stripping

If word begins with $in-$, $un-$, $non-$, $pre-$, $sub-$, etc

Strip the prefix from word w to form u

If stripped form u found in lexicon

Output attributes of the prefix and stem and exit

4. Compound word decomposition

If detect word-medial case differences within word w

Break word w into a multiple words u_1, u_2, u_3, \dots according to case changes

For words u_1, u_2, u_3 , **goto** 1.

Else if word w can be decomposed into two nouns u_1, u_2

Output attributes of the u_1, u_2 and exit

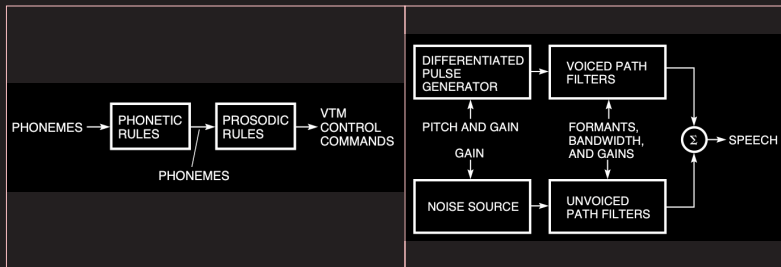
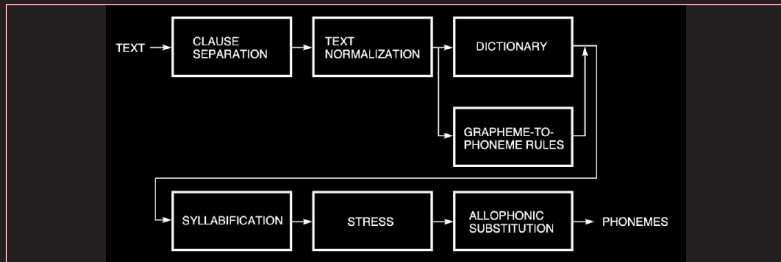
5. Pass word w to letter-to-sound module

Prosody

- speaking style (character and emotion)
- symbolic prosody (pauses, punctuation)
- accent and stress
- tone (e.g. yn- vs. wh-questions)

Speech synthesis

- Articulatory synthesis
- Rule-based formant synthesis
- Concatenative synthesis
- Statistical parametric synthesis
(generative synthesis)
- Deep learning-based synthesis



Concatenative synthesis (unit-selection)

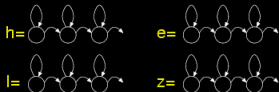
- Searches for the best sequence of (variably sized) units of speech in a large, annotated corpus of recordings, aiming to find a sequence that closely matches the target sequence.
- Sounds good if you fit the bits well. Also can be tuned.
- Can be really expensive to build: need professional speakers, many hours of stable recordings.

Statistical parametric speech synthesis

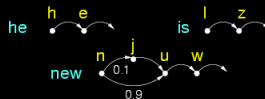
- Essentially, reproduces the speech signal.
- Break speech into several factors and represent them as numbers.
- Learns the model from data.
- From the model we can generate a spectrogram and then create a waveform.

HMM: from phonemes to sentences

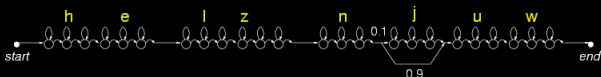
HMM phone models



Lexicon



Sentence model: 'he is new'



HMM speech synthesis

- a) sequence model: a weighted finite state network of states and transitions, and 2) observation model: multivariate Gaussian distribution in each state.
- Generates most likely sequence for the given phoneme(s).
- A global optimisation then computes a stream of vocoding features that optimise both HMM emission probabilities and continuity constraints.

Neural speech synthesis

- Essentially it is a type of statistical parametric speech synthesis
- text to features and then features to sound (neural vocoder)
- sequence2sequence models (e.g. RNN, GRU)
- e.g. LPCNet, WaveNet

Some advanced matters

- adapting voices, by changing a few parameters in the model
- repairing voices
- Simon King - Using Speech Synthesis to give Everyone their own Voice: <https://youtu.be/xzL-pxcpo-E>
- Incrementality: INPRO_iSS <https://www.youtube.com/watch?v=kwNvcUXfD7Y>

Evaluation

- Intelligibility tests, e.g. rhyme tests
- Quality tests: mean opinion scores (MOS) and preference test
- Functional tests
- Automated tests, mainly for letter-to-sound or prosody