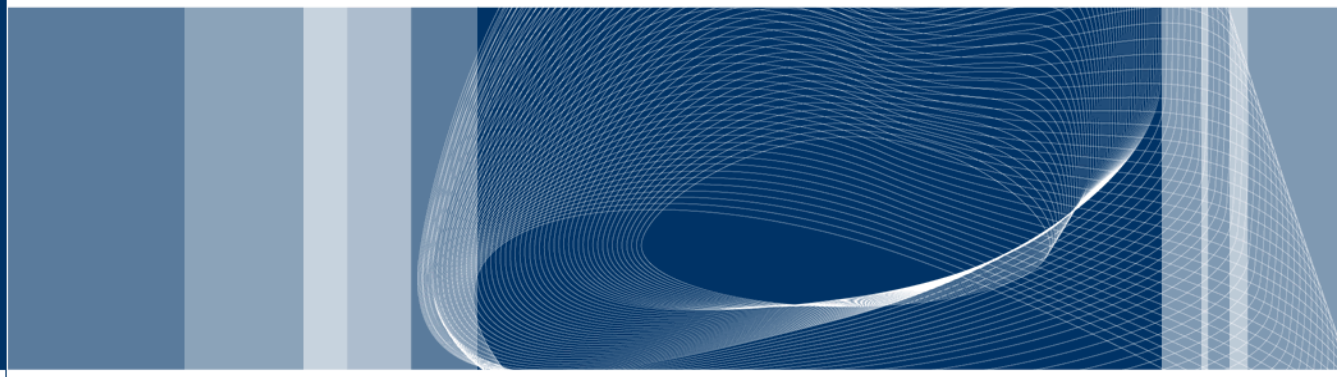
POLITECNICO DI MILANO

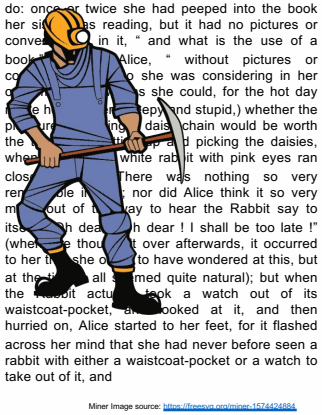
Natural Language Processing



Speech

Natural Language Processing

Some slide content based on textbooks:
Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition *by Daniel Jurafsky and James H. Martin*



Miner Image source: https://freesvg.org/miner-1574424884

Lecture Contents:

- Human Voice
- Mel Spectrogram
- Speech to Text
- Text to Speech

Human Speech

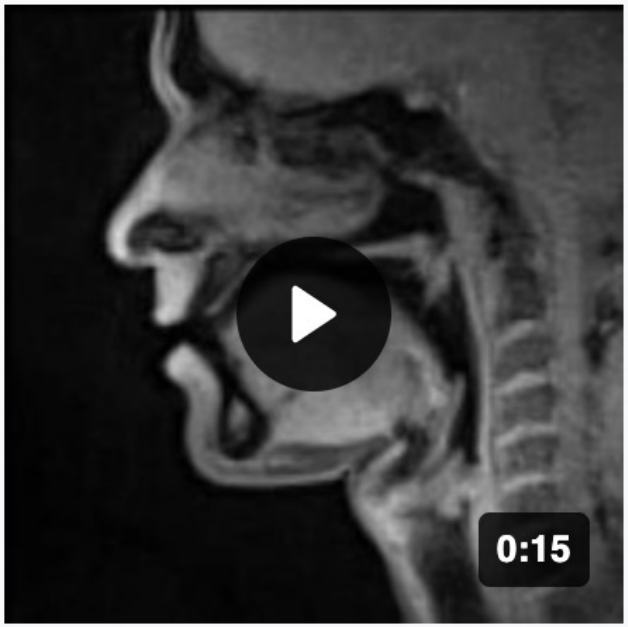
Human Speech

Human speech (see https://en.wikipedia.org/wiki/Speech) consists of:

- **vowels**: sounds pronounced **without restricting** the vocal tract
- **consonants**: sounds made by **partially or completely closing** vocal tract

Different sounds are referred to as phones/**phonemes**

- see https://en.wikipedia.org/wiki/Phoneme

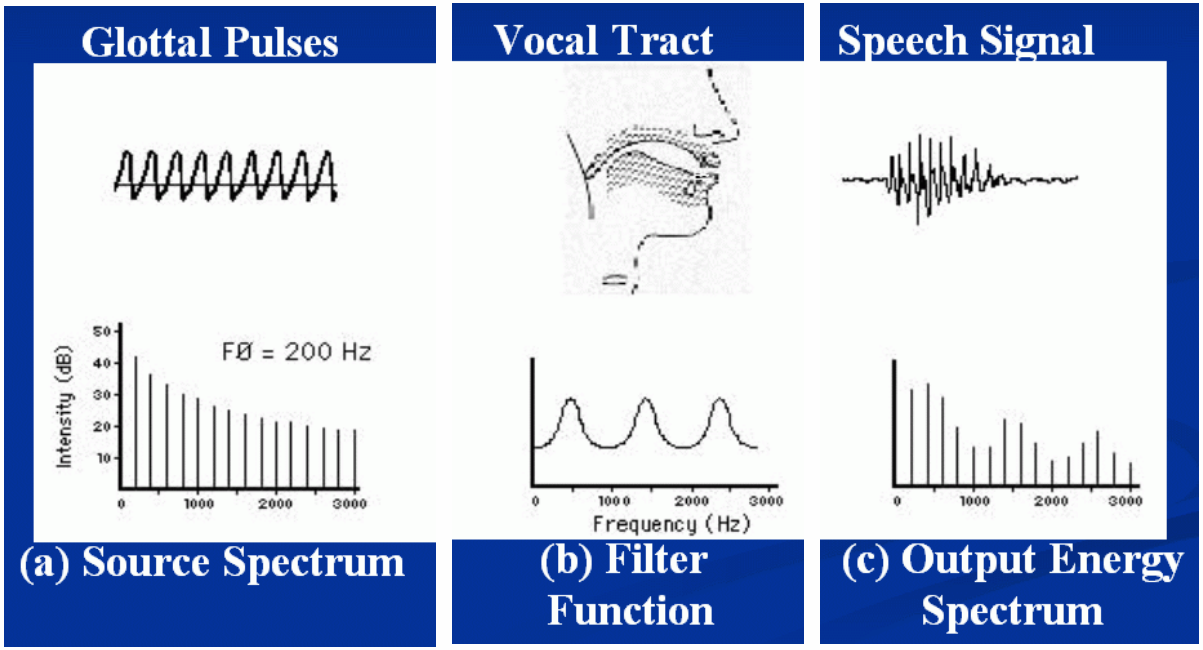


Video source:
Martin Uecker, Shuo Zhang, Dirk Voit, Alexander Karaus, Klaus-Dietmar Merboldt, and Jens Frahm, Real-time magnetic resonance imaging at a resolution of 20 ms, NMR in Biomedicine 23: 986–994 (2010) DOI:10.1002/nbm.1585
https://en.wikipedia.org/wiki/File:Real-time_MRI_-_Speaking_(English).ogv

Source-filter model

Model of human phonation:

- larynx/glottis produces pulses (source)
- vocal tract shapes such pulses (filter)



- source not important for ASR since filter carries important info

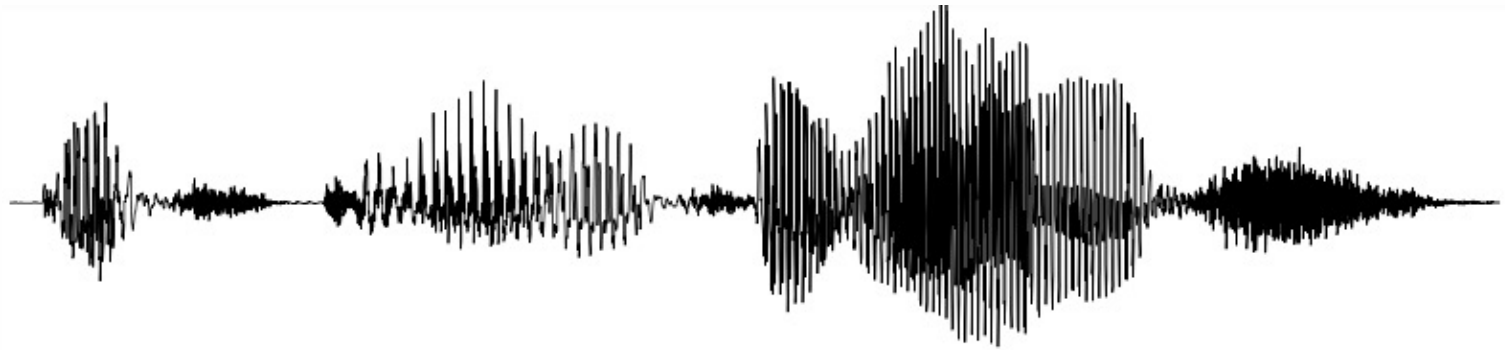
Image source: http://mirlab.org/jang/books/audiosignalprocessing/humanVoiceProduction.asp?title=3-3%20Human%20Voice%20Production

Speech as a Time Series

Speech as a time series

Speech is just a sound wave

- which is a time series of pressure values over time
- example wave for somebody saying: *It's time for lunch!*



Series of sounds

Types of sounds in speech

- **vowel**: periodic signal, e.g. *a* in *has*
- **fricatives**: consonants produced by forcing air through narrow channel, e.g. *ch* in *watch*
- **glides**: smooth transition, e.g. *w*→*a* in *watch*
- **bursts**: rapid transition, e.g. *d* in *dime*

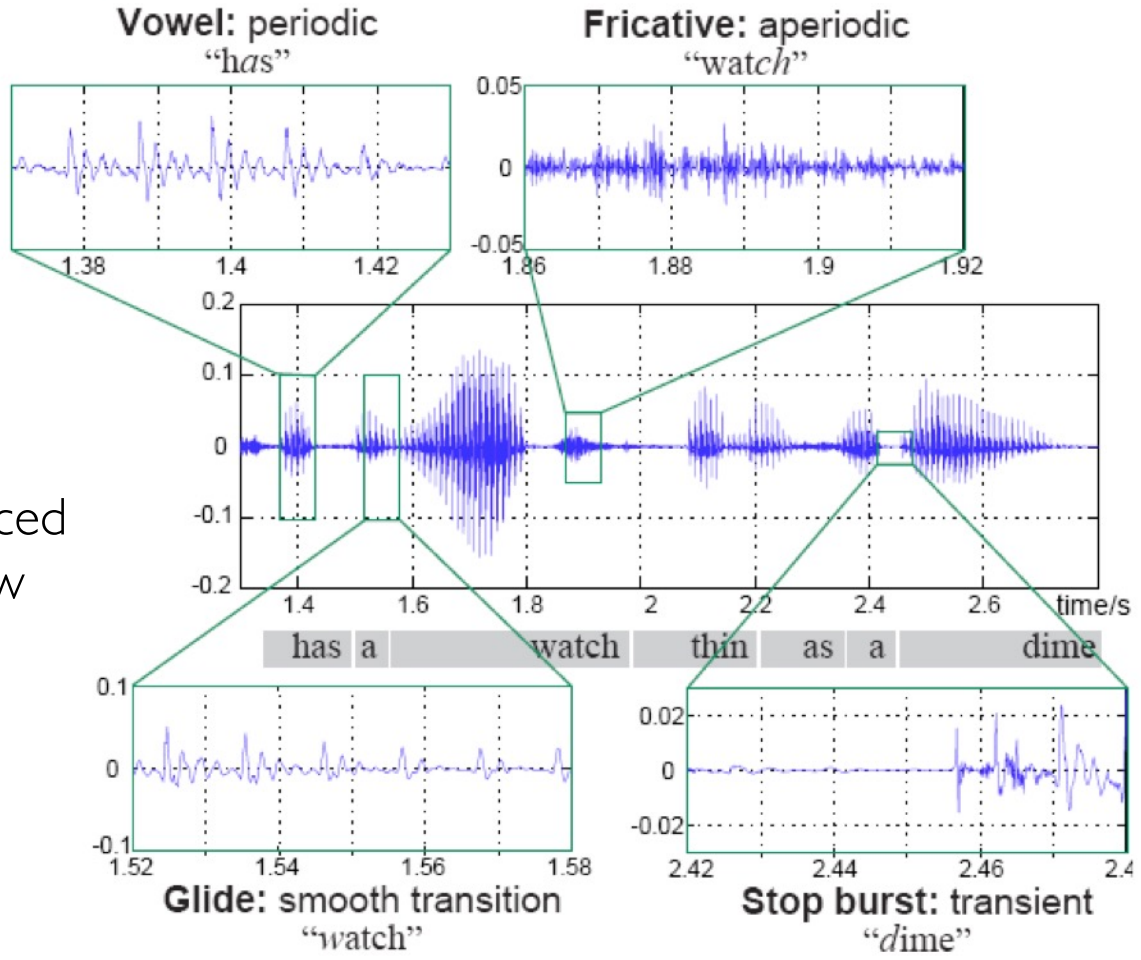


Image source:
Dan Ellis, Columbia University
https://www.ee.columbia.edu/~dpwe/classes/e6820-2004-01/lectures/E6820-L01-intro+dsp-2up.pdf

Viewing sounds in Frequency Domain

To distinguish different types of sounds

- view them in frequency domain using Fourier Transform
- i.e. find frequencies that make up signal

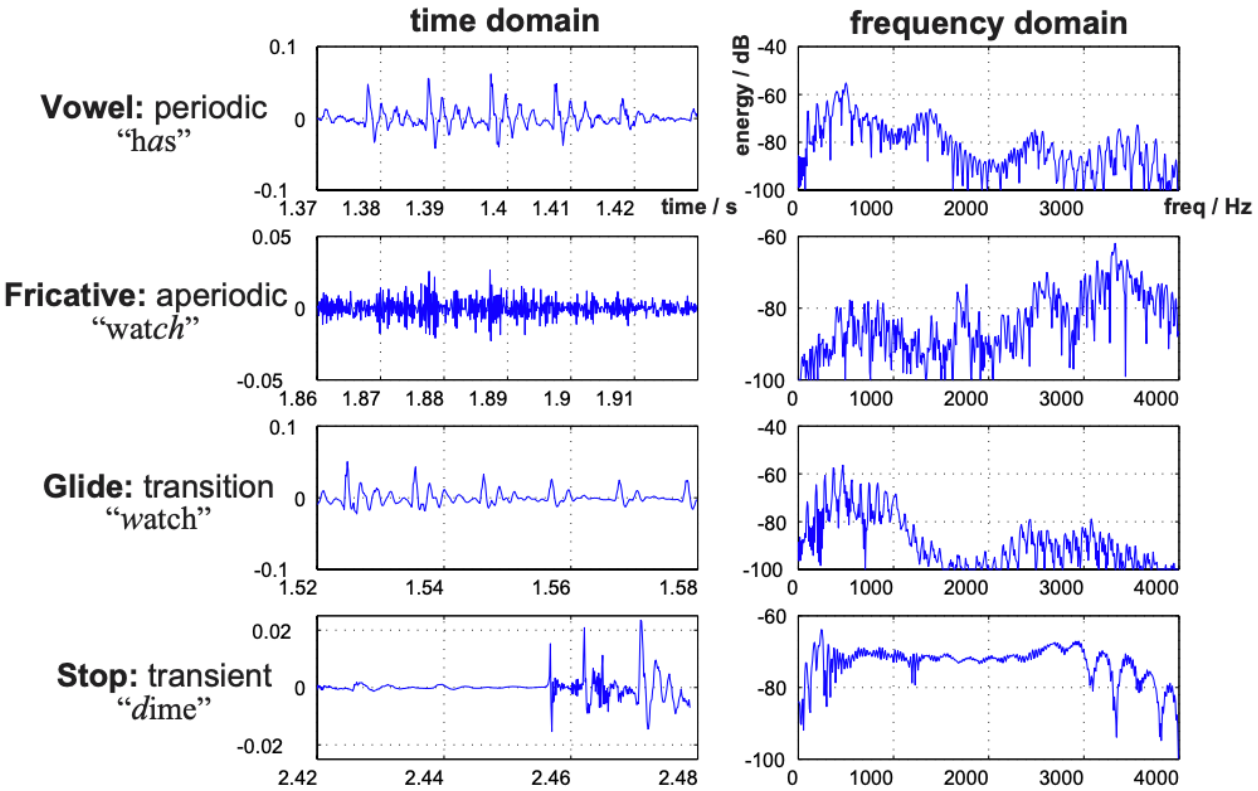


Image source:
Dan Ellis, Columbia University
https://www.ee.columbia.edu/~dpwe/classes/e6820-2004-01/lectures/E6820-L01-intro+dsp-2up.pdf

Converting audio signal into 2 dimensional representation

Spectrogram (3d view)

Audio signal consists of **sequence of sounds**

- so need to see frequency domain representation of **consecutive sounds**
- called a **spectrogram**

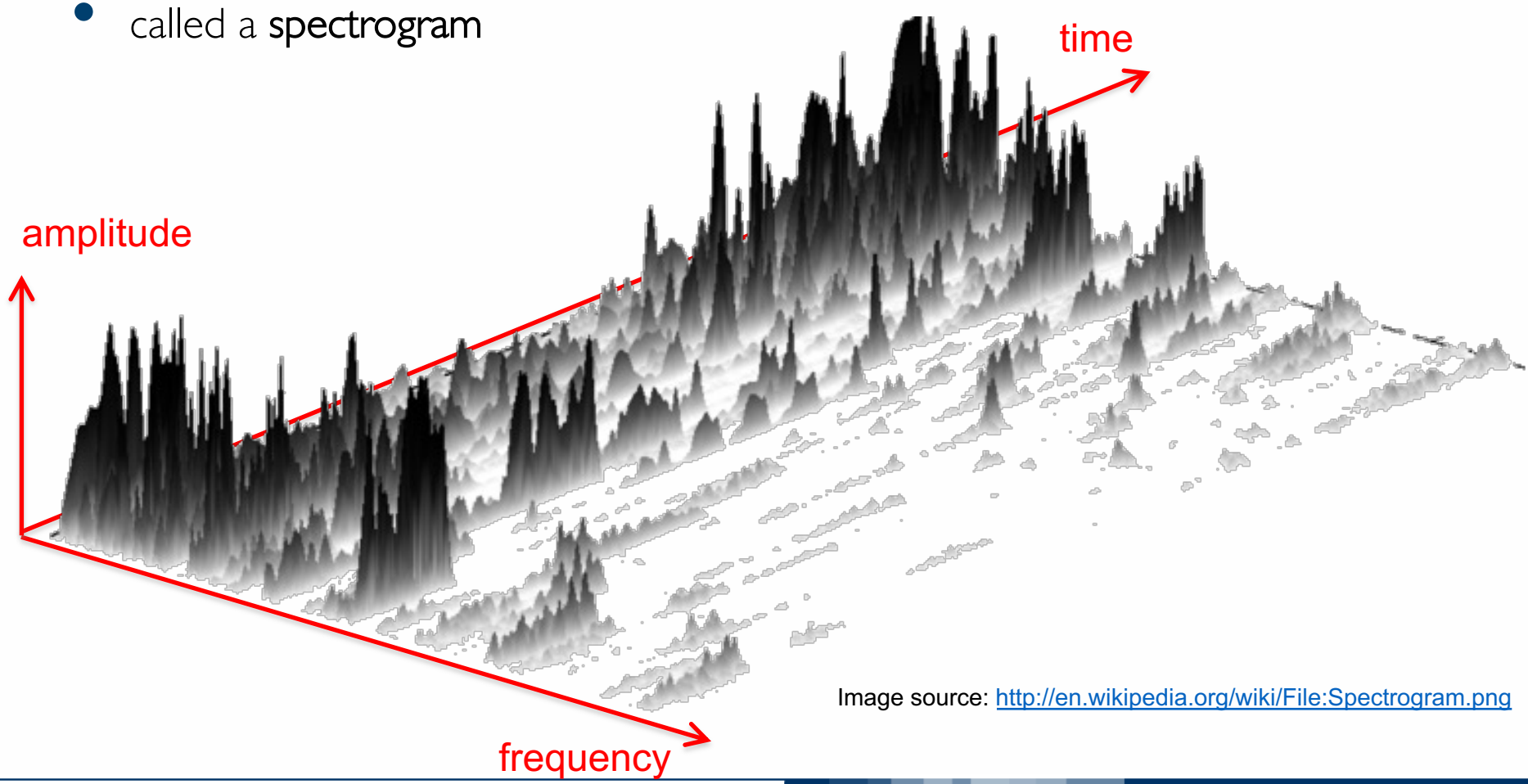
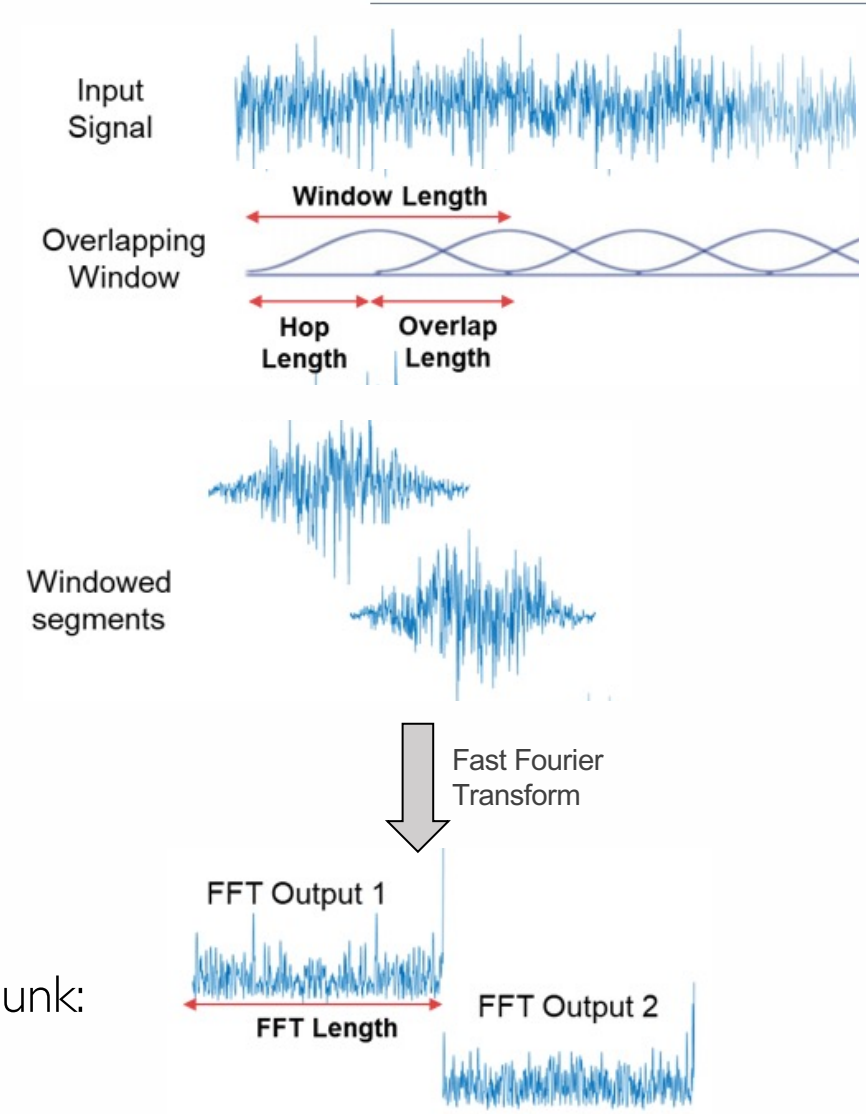


Image source: http://en.wikipedia.org/wiki/File:Spectrogram.png

Short-time Fourier Transform (STFT)



Finds frequency components (amplitude of sinusoids) in signal sections

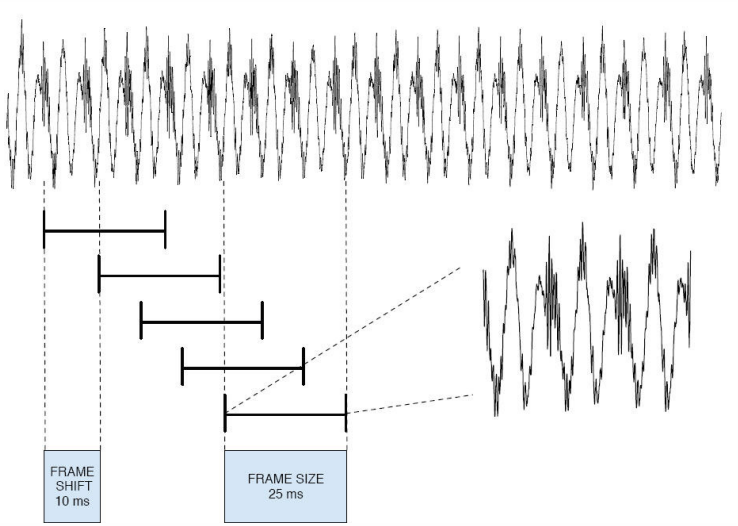
- divides signal into M chunks of L samples $x_m[]$
- each chunk multiplied by window function $w[]$

Vector containing Fourier transform of m^{th} chunk:

$$\text{STFT}_m[k]\{x_m\} = \sum_{n=0}^{L-1} x_m[n] \cdot w[n] \cdot e^{-j\frac{2\cdot\pi}{L}\cdot k\cdot n}$$

Image source: https://it.mathworks.com/help/dsp/ref/dsp.stft.html

Chunks & Windows



Overlapped windows to reduce border effects

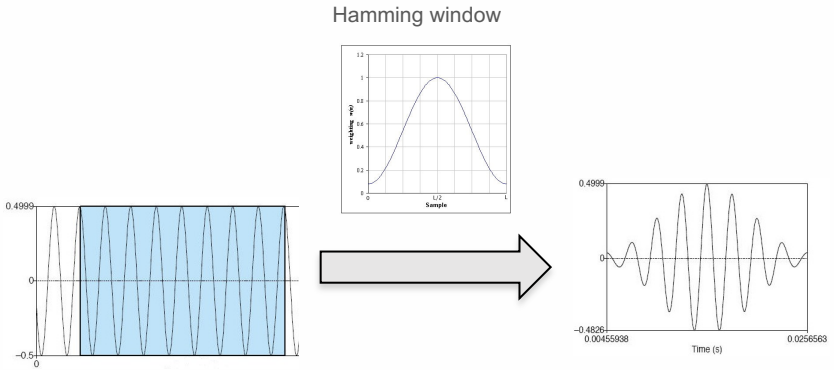
- each chunk usually 25 ms
- separated by 10 ms

Most used window function

- Hamming window:

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{L}\right); \quad 0 \le n \le L-1$$

- further reduces border effects



Spectrogram (2d view)

- Usually view spectrogram as 2 dimensional heatmap:

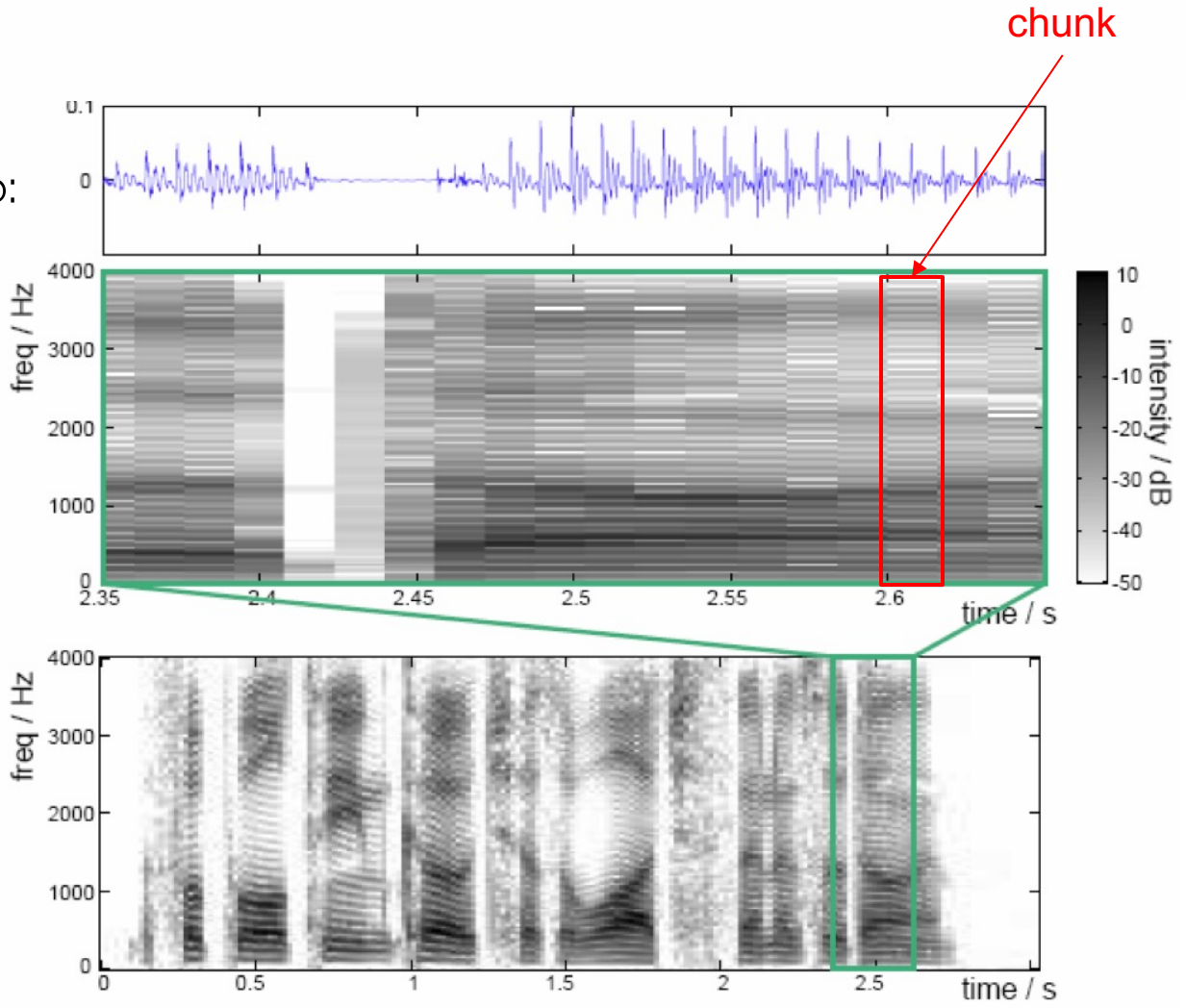


Image source:
Dan Ellis, Columbia University
https://www.ee.columbia.edu/~dpwe/classes/e6820-2004-01/lectures/E6820-L01-intro+dsp-2up.pdf

Pre-emphasis filter

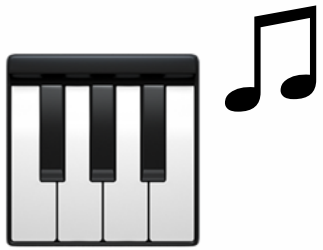
Before running STFT

- apply **pre-emphasis filter** to amplify high frequencies
- useful for:
  - **balancing spectrum** since high frequencies usually have smaller amplitudes
  - **avoiding numerical problems** during Fourier transform
  - may also **improve Signal-to-Noise Ratio** (SNR)
- applied to signal using first order filter: $x(t)=s(t)-\alpha \cdot s(t-1)$
  - typical values for α are 0.95 or 0.97
  - see https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

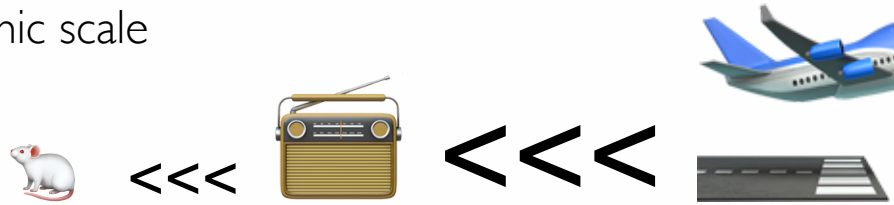
Human hearing

- Humans hear frequencies in range: 20Hz to 20 kHz
  - dogs and cats hear higher frequencies than us

- Distinguish noises based on their relative pitch
  - keys on piano increase pitch by **fixed multiple** of $2^{1/12} = 1.0595$
  - each semitone approximately 6% higher frequency than previous
  - so **not increasing linearly**, but **multiplicatively**



- Humans hear many orders of loudness
  - so represent amplitude on a logarithmic scale



Mel Spectrogram

1. Limit frequency range to maximum 8kHz
2. Represent frequencies on a logarithmic scale
   - pitch f increases multiplicatively in y direction: $y = 2595 \log_{10}(1+f/700)$
3. Represent amplitude on a logarithmic scale
   - measure amplitude of signal in decibels: $db = 10 \log_{10}(z)$

For more info, see:

- https://en.wikipedia.org/wiki/Mel_scale#
- https://ketanhdoshi.github.io/Audio-Mel/#

Mel (melody) scale ← human perception of equidistant pitches

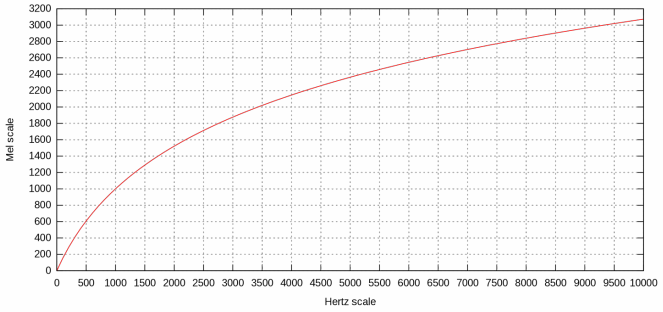


Image source https://en.wikipedia.org/wiki/Mel_scale#/media/File:Mel-Hz_plot.svg

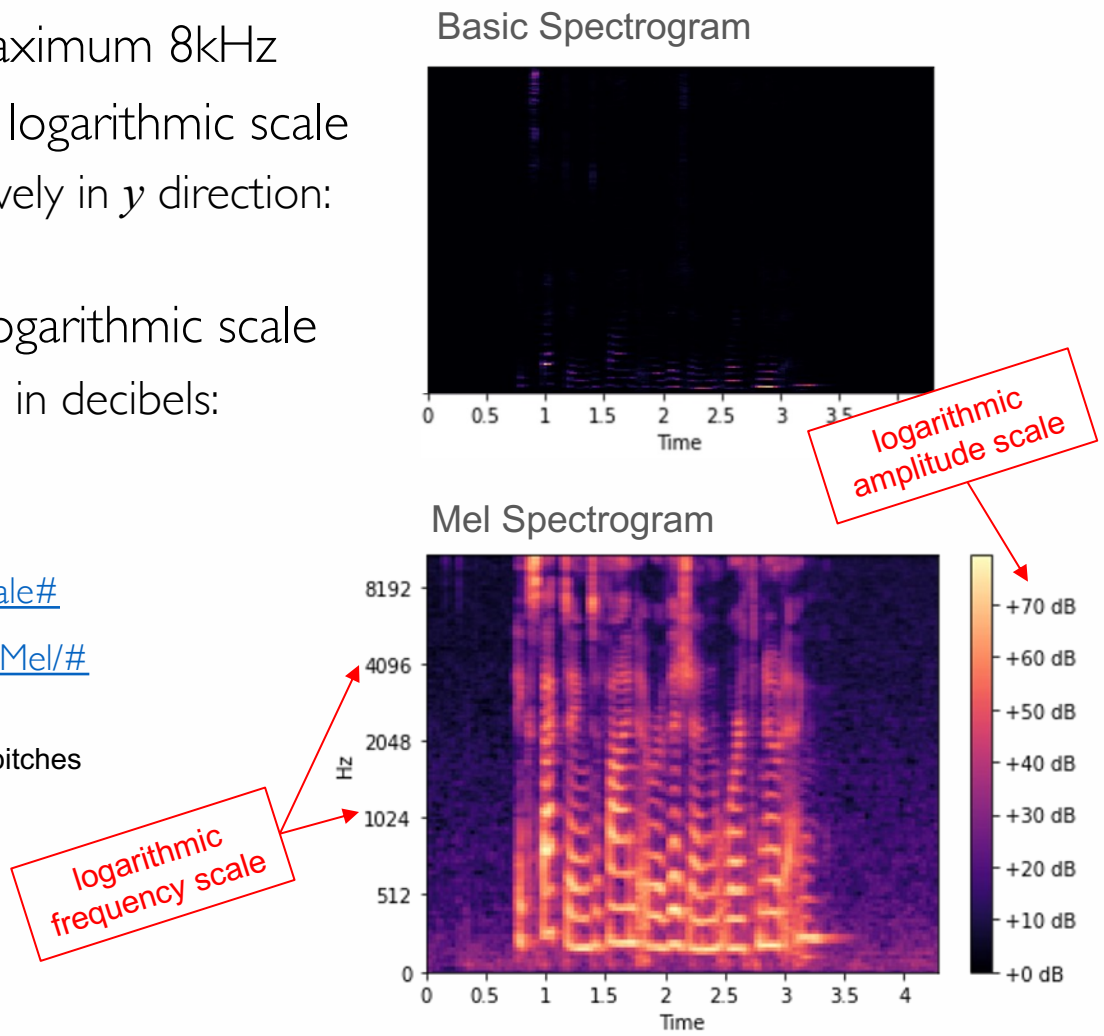


Image source: Ketan Doshi's blog post
https://ketanhdoshi.github.io/Audio-Mel/#

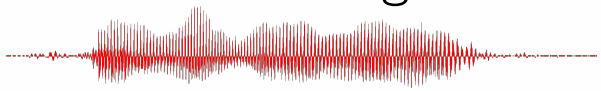
Speech-to-Text

a.k.a.

Automatic Speech Recognition (ASR)

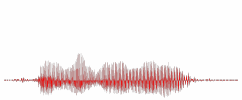
Speech Recognition

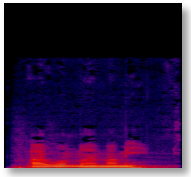
Problem of converting audio signal to text:



"I want my mommy!"

- up until relatively recently was tackled using:
  - features extracted from Mel Spectrogram (mel-frequency cepstrum coefficients)

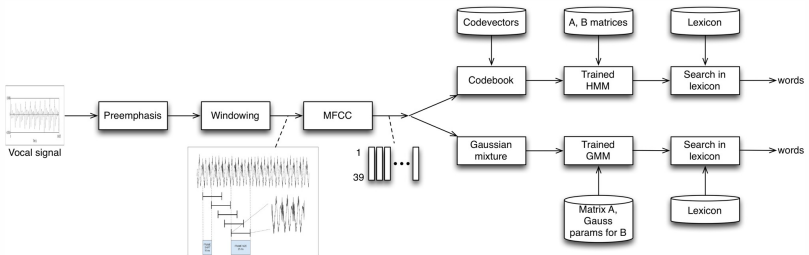




MSCC features

"I want my mommy!"

  - and Hidden Markov Models with Gaussian Mixture Models:



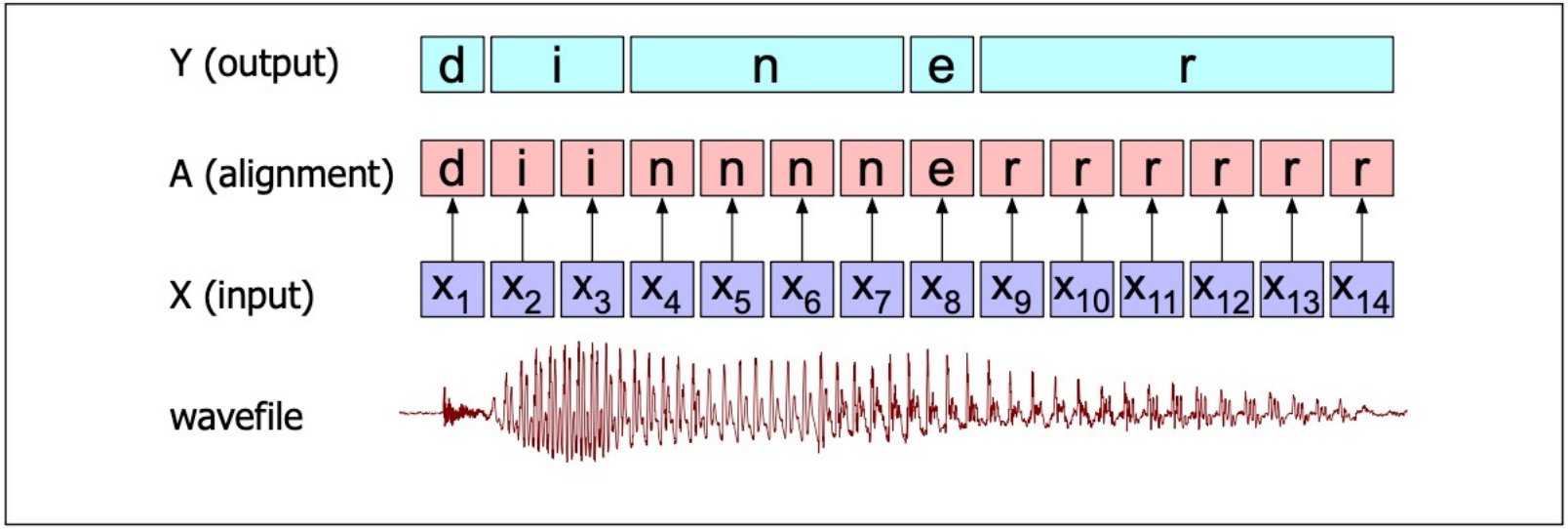
- but then **Deep Learning** came along …

Problems with classifiers:

Can train a convolutional neural network (CNN) model:

- directly on the input waveform
- or on the chunks from the Mel Spectrogram

Problem: classifiers work on fixed windows, so need to work out how



- So was the word *diner* or *dinner*?

Use sequence-2-sequence model

- Solution: **Encoder-decoder model** can deal with problem of producing output that isn't the same length as the input

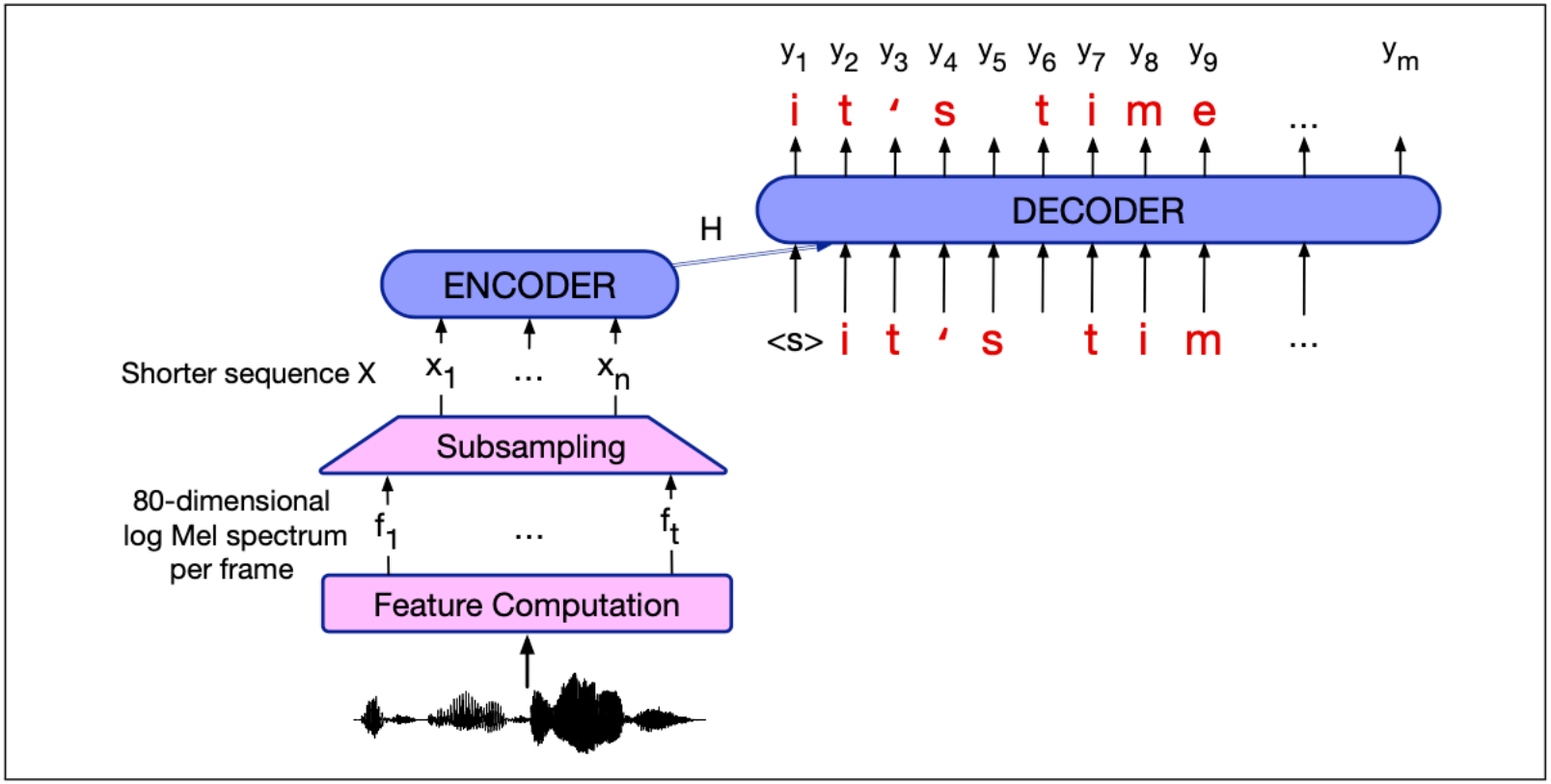


Figure 16.6 Schematic architecture for an encoder-decoder speech recognizer.

Wav2vec (2020)

Powerful recent **transformer-based** architecture

- Works with original/raw audio in time series representation

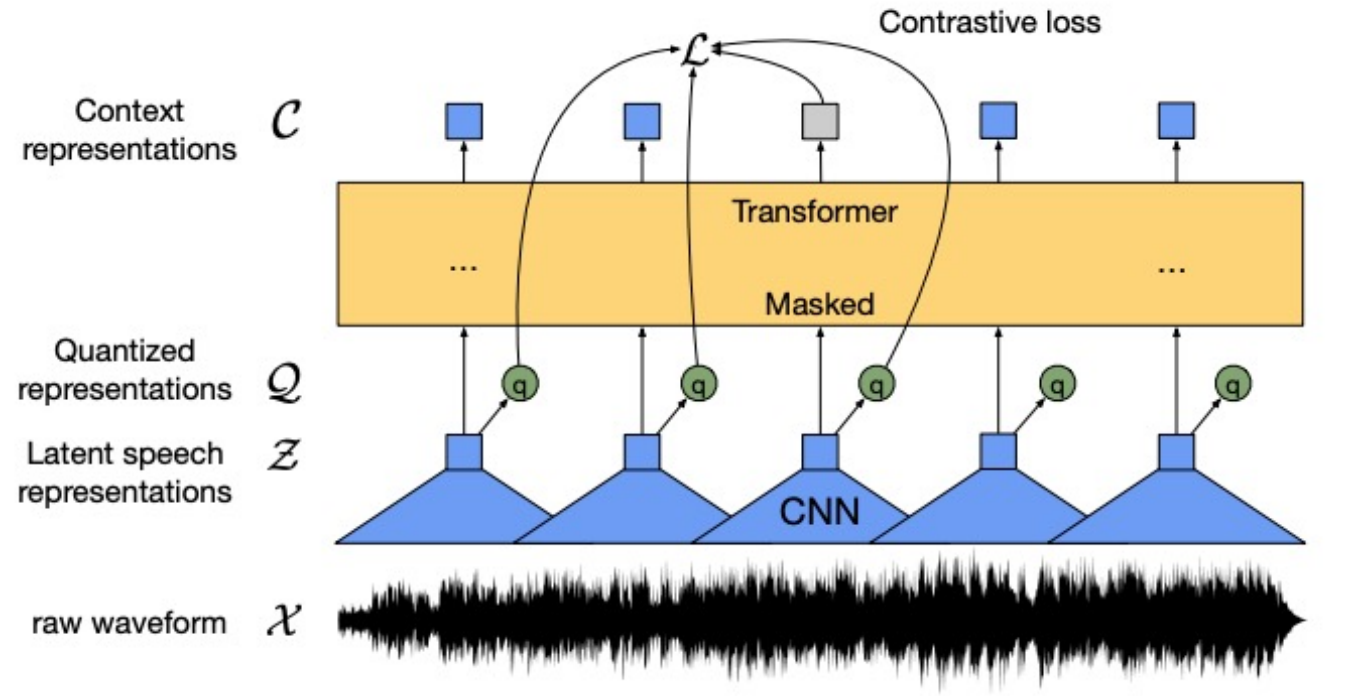


Image source:
wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations
Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli,
https://arxiv.org/pdf/2006.11477.pdf

Whisper (2022)

Recent Transformer-based system with state-of-the-art performance

- makes use of Mel spectrogram as input representation
- https://arxiv.org/pdf/2212.04356.pdf

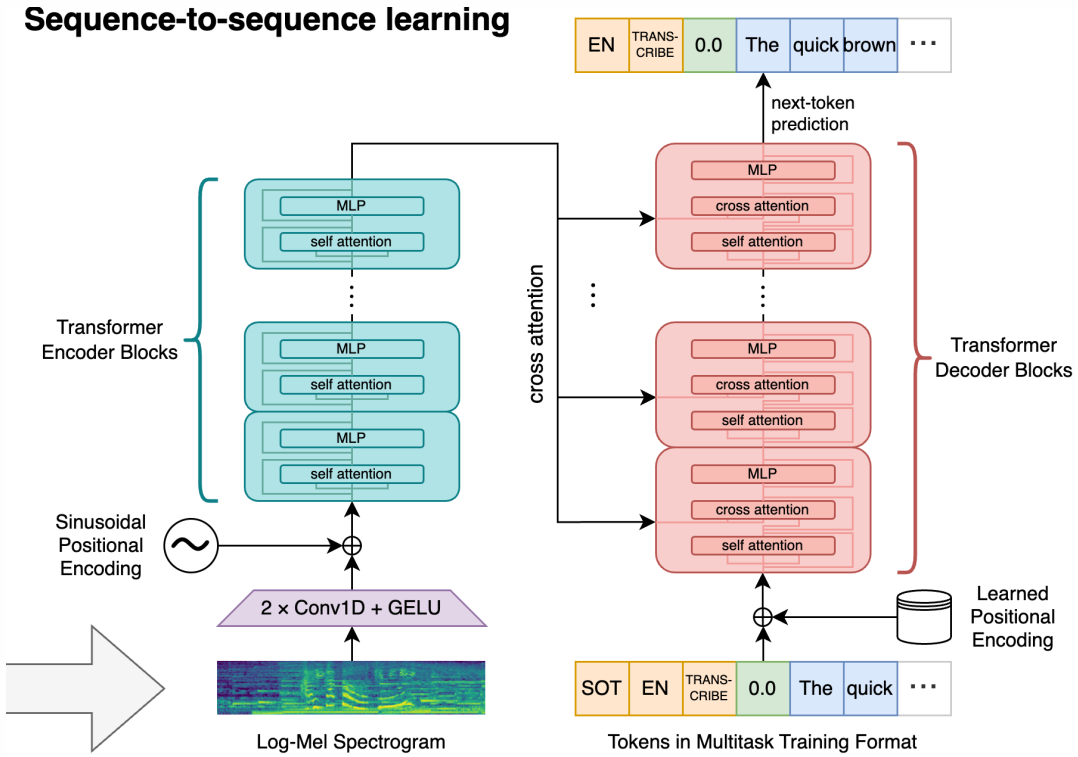
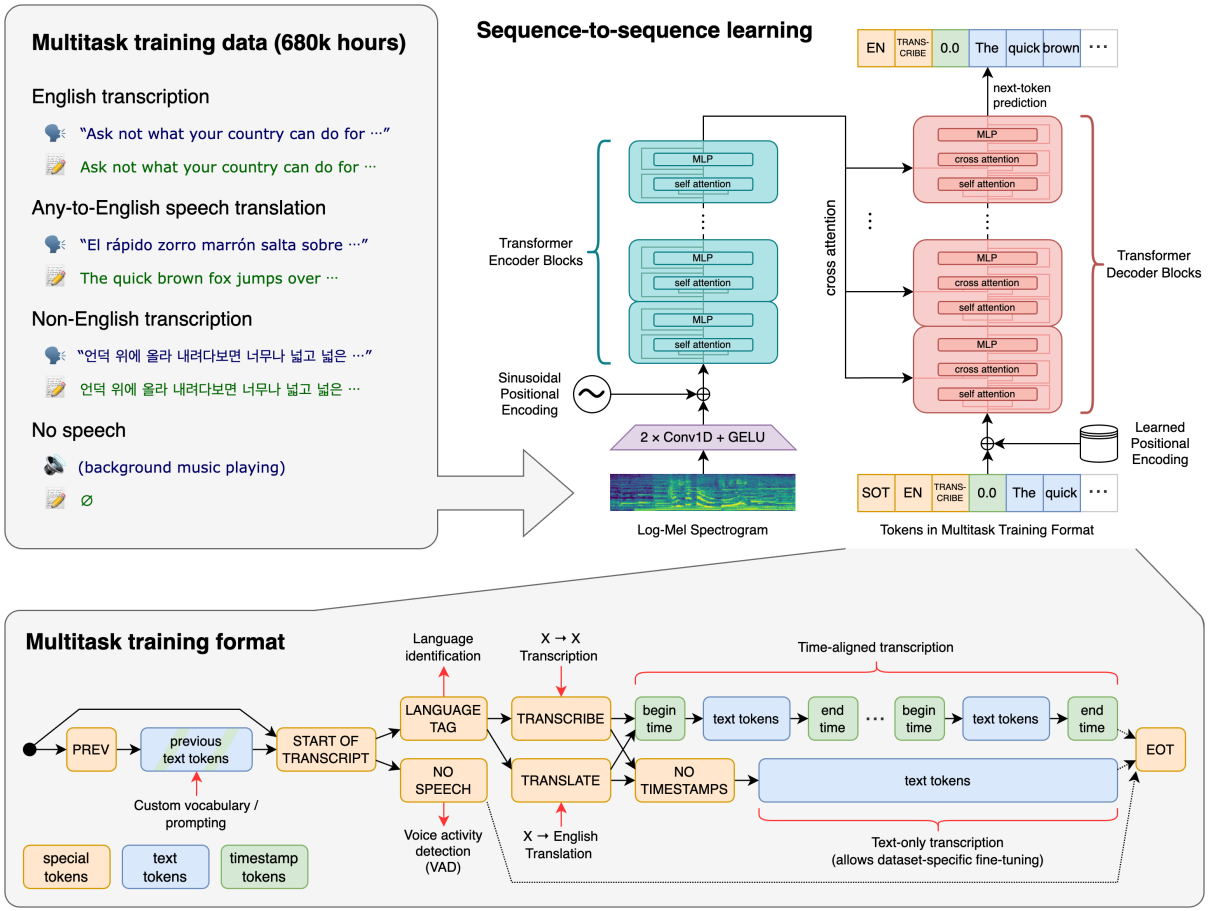


Image source:
Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever
Robust Speech Recognition via Large-Scale Weak Supervision
https://arxiv.org/pdf/2212.04356.pdf

Whisper (cont.)

- Weakly supervised allowing for bigger training sets
- Multi-lingual training
- Multi-task training



Evaluation of Speech to Text

Word Error Rate (WER) in a string:

- how many *detected* words differ from *correct* words
- based of edit distance

$$WER = 100 \cdot \frac{\text{Insertions + Substitutions + Deletions}}{\text{Total words in correct transcript}}$$

Sentence Error Rate (SER):

- how many sentences had at least one error?

$$SER = 100 \cdot \frac{\text{\# of sentences with at least one errror}}{\text{Total number of sentences}}$$

Advanced: speaker dependence

- Vocal tract length normalization
  - warping frequency axis of speech power spectrum
  - accounts for fact that precise locations of vocal-tract resonances vary roughly monotonically with physical size of speaker
  - normalize on speakers' age
- Pitch normalization
  - Normalization could help improving accuracy for children
- Modify acoustic model
  - start with trained model and small dataset from new speaker
  - transform learned to maximize performance for new speaker

Advanced ASR: non-words

- Non words
  - short non-verbal sounds (coughs, loud breathing, throat clearing, …) → filled pauses
  - environmental sounds (beeps, telephone rings, door slams, …)
- For each non-verbal sound:
  - create special phone and add special word to lexicon for that phone
- use normal training to train these phones
  - training data transcripts include labels for these new special words
  - these words need to be added to language model; often by just allowing them to appear in between any word

Text-to-Speech
a.k.a.
Speech Synthesis

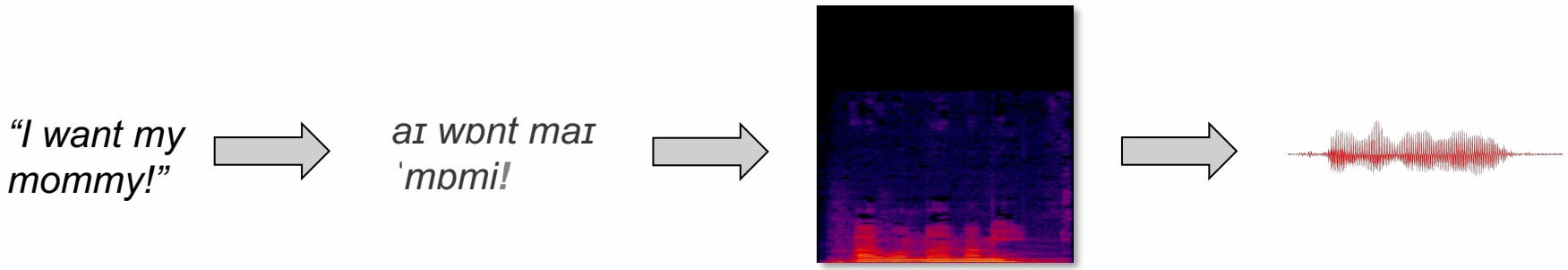
Text-to-speech

As with speech-to-text,

- text-to-speech technology has got a LOT better over the last few years

Aim is to convert **text string into audio waveform**

- often implemented as 3 stage system:
  1. **text** to **phoneme**
  2. **phoneme** to **mel spectrogram**
  3. **mel spectrogram** to **audio signal**



Graphic created with: https://tophonetics.com/, https://convert.ing-now.com/mp3-audio-waveform-graphic-generator/

Text Normalization

Abbreviated text needs to be **expanded** during text-to-speech process,

- e.g. to convert the phrase: *They live at 224 Mission St.*
- into: *They live at two twenty four Mission Street*

Normalization process depends on context:

- e.g., number *1750* can be verbalised in many different ways:
  - *The economy in the year 1750* → *The economy in the year seventeen fifty*
  - *The password is 1750* → *The password is one seven five zero*
  - *It costs 1750 dollars* → *It costs one thousand seven hundred and fifty dollars*
- So need to train a seq2seq model to expand appropriately

semiotic class	examples	verbalization
abbreviations	**gov't**, *N.Y., mph*	government
acronyms read as letters	**GPU**, *D.C., PC, UN, IBM*	G P U
cardinal numbers	**12**, *45, 1/2, 0.6*	twelve
ordinal numbers	*May* **7**, *3rd, Bill Gates III*	seventh
numbers read as digits	*Room* **101**	one oh one
times	*3.20,* **11:45**	eleven forty five
dates	**28/02** *(or in US, 2/28)*	February twenty eighth
years	**1999**, *80s, 1900s, 2045*	nineteen ninety nine
money	**$3.45**, *€250, $200K*	three dollars forty five
money in tr/m/billions	**$3.45 billion**	three point four five billion dollars
percentage	**75%** *3.4%*	seventy five percent

Text Normalization (cont.)

Homograph disambiguation

- The English language contains some words that are
  - Like "bass" as fish → /b ae s/ or as an instrument → /b ey s/

- I like playing my bass guitar
- I just cooked bass for dinner


or

?

Tacotron2

- text2speech architecture from 2018
- still uses LSTMs?
  - surprising if a Transformer-based architecture doesn't outperform it …

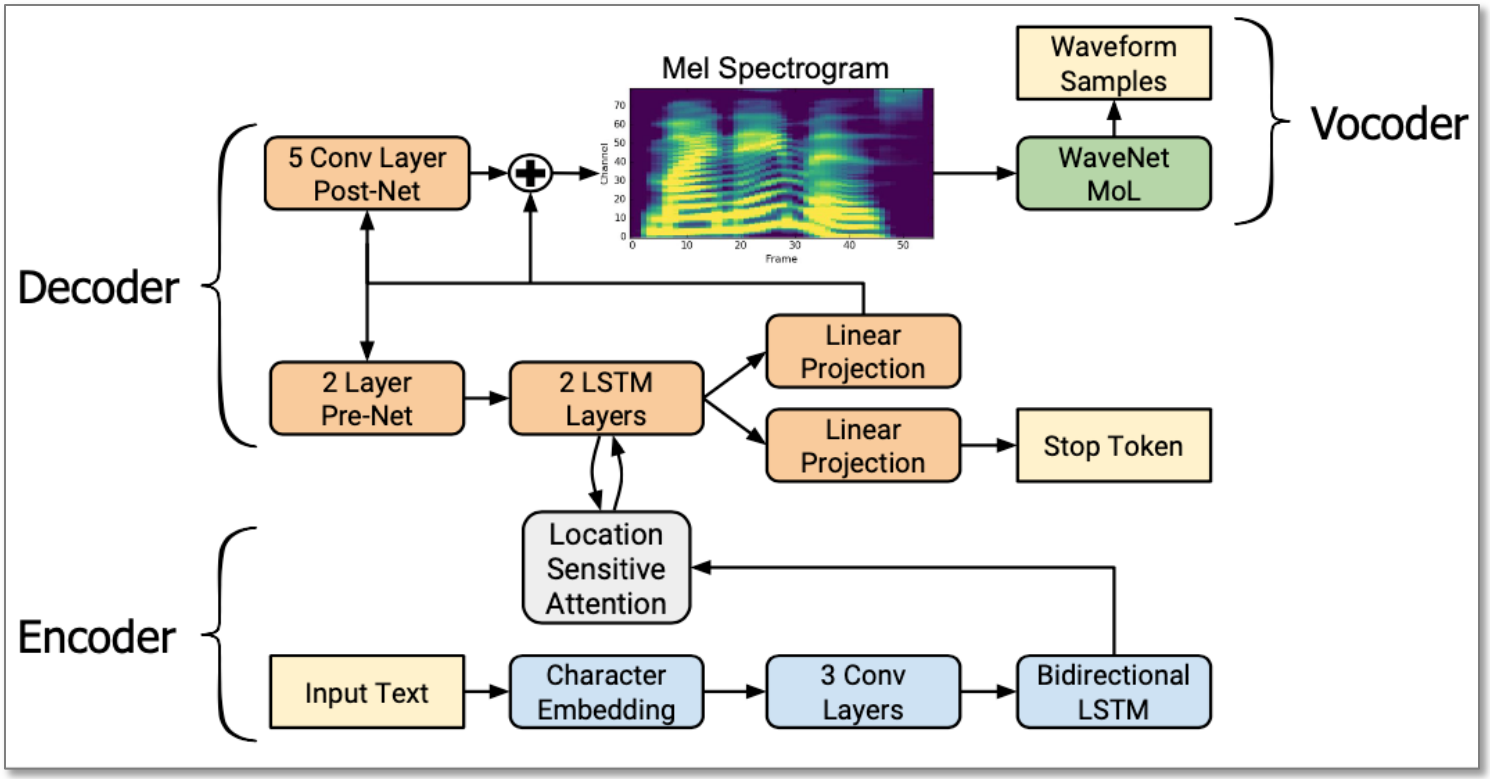


Image source:
https://pytorch.org/hub/nvidia_deeplearningexamples_tacotron2/

WaveNet

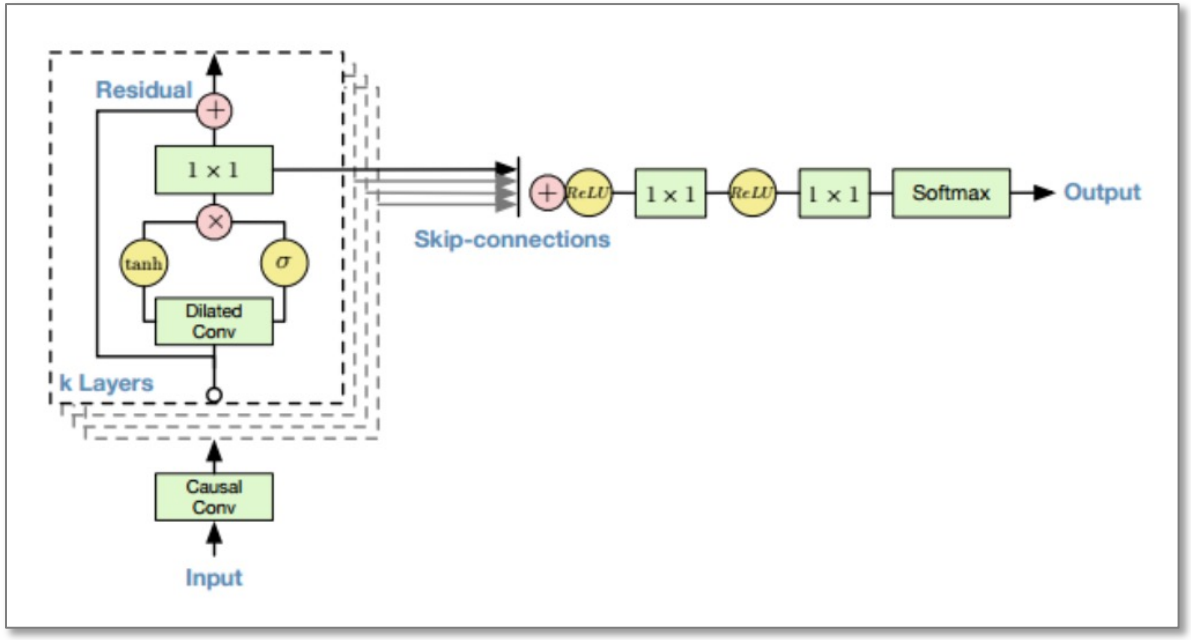


Image source:
https://nvidia.github.io/OpenSeq2Seq/html/speech-synthesis/wavenet.html

Vocoder

- converts from Mel Spectrogram to audio signal
- used in Tacotron2

Architecture:

- Autoregressive diluted convolution based generation of signal

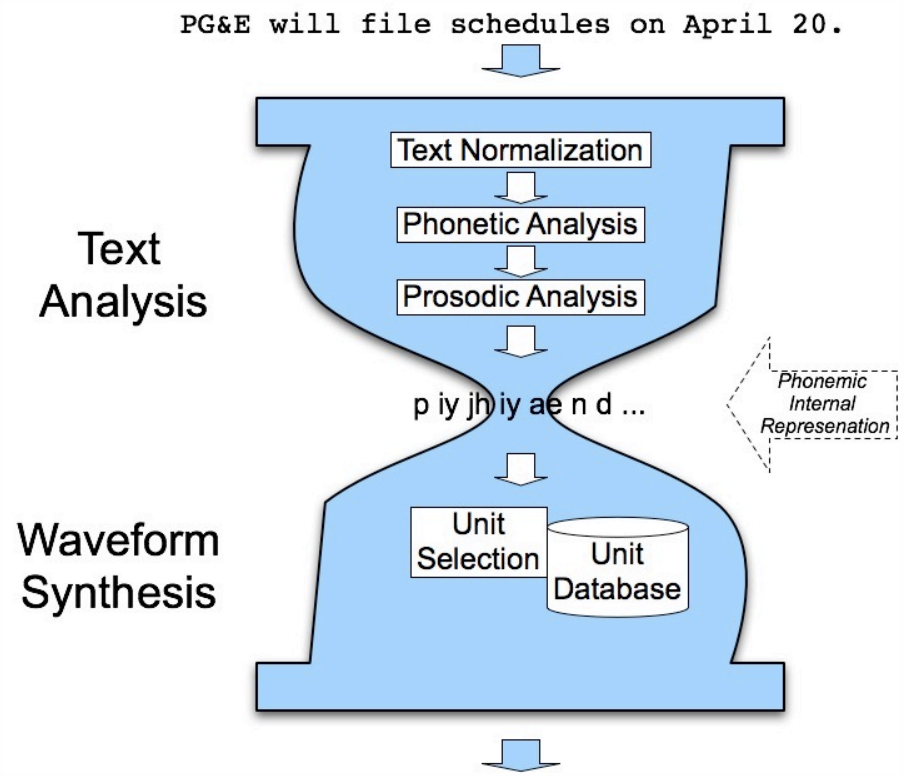
Speech Synthesis

- Goal: transforming a text string into a waveform
- Steps:
  1. Text analysis: text string → phonetic representation
  2. Waveform synthesis: phonetic representation → waveform
- Approaches for waveform synthesis:
  - Formant synthesis: rules/filters are used to create speech using additive synthesis and an acoustic model (physical modelling synthesis) → "robotic" voice…
  - Articulatory synthesis: modelling and simulating movements of articulators and acoustics of vocal tract → complex
  - Concatenative synthesis: most used approach in modern TTS

Concatenative TTS

Hourglass model has two steps

1. Text analysis
   - From text to phonemes
2. Waveform synthesis
   - From phonemes to waves
   - Concatenates small, prerecorded wave units



1. Text analysis

- Text normalization
  - Tokenization
  - Dealing with non-standard words
  - Homograph disambiguation
- Phonetic analysis
  - Producing the string of phonemes
- Prosodic analysis
  - Add information encoding prosody

Phonetic analysis

From the preceding phase

- transform a word into a list of phonemes

Two general approaches

- dictionary-based: collection of pronunciations, for each word
- grapheme-to-phoneme (aka g2p) conversion: for words non present in the dictionary (mostly, names)

For transparent languages (e.g., Italian)

- use pronunciation rules
- dictionary for irregular forms and foreign words

Phonetic analysis

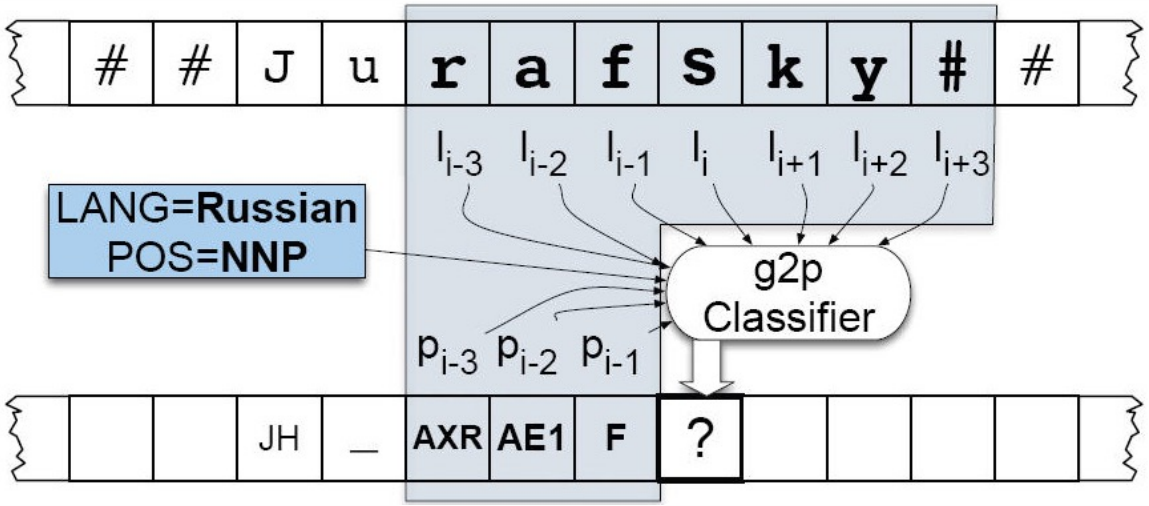
Dictionary contains the phonetic representation of words

- for example, the CMU dictionary

ANTECEDENTS	AE2 N T IH0 S IY1 D AH0 N T S	*PAKISTANI*	P AE2 K IH0 S T AE1 N IY0
CHANG	CH AE1 NG	*TABLE*	T EY1 B AH0 L
DICTIONARY	D IH1 K SH AH0 N EH2 R IY0	*TROTSKY*	T R AA1 T S K IY2
DINNER	D IH1 N ER0	*WALTER*	W AO1 L T ER0
LUNCH	L AH1 N CH	*WALTZING*	W AO1 L T S IH0 NG
MCFARLAND	M AH0 K F AA1 R L AH0 N D	*WALTZING(2)*	W AO1 L S IH0 NG

Names and other unknown words:

- g2p: a classifier
- Uses several features



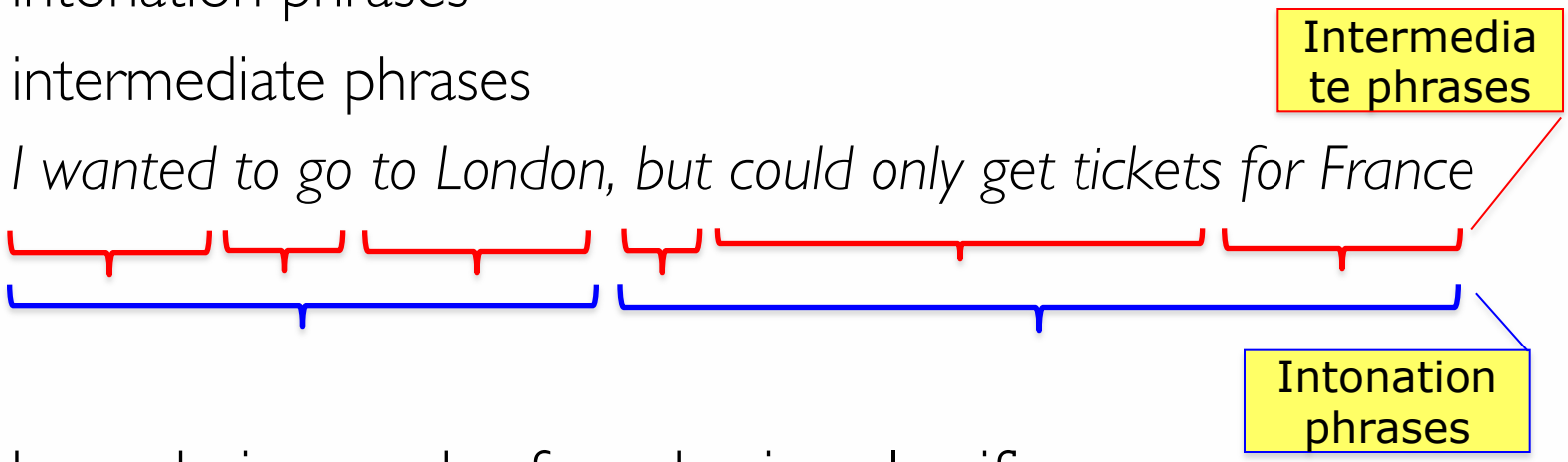
Prosodic analysis

Prosody: intonation, stress, and rhythm

- changes in pitch, phoneme duration and energy

Utterances have prosodic structure

- intonation phrases
- intermediate phrases
- *I wanted to go to London, but could only get tickets for France*



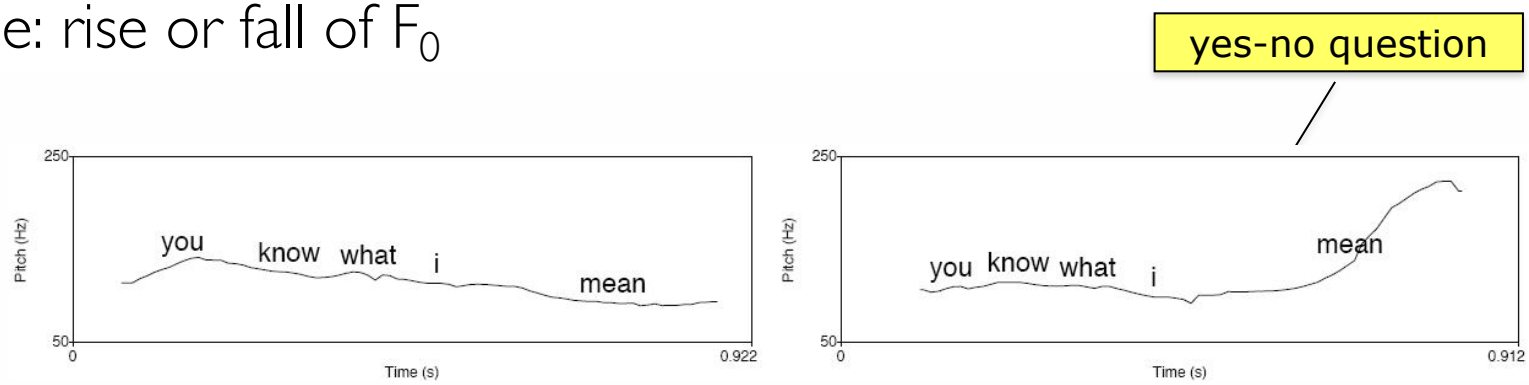
- boundaries can be found using classifiers

Prosodic analysis: prominence

Prosodic prominence (stress

- in phrase, some words are stressed and made more prominent → "pitch accent"
- realized by means of pitch and/or rhythm and/or energy
  - indicates "stress"
  - The pitch accent on a word is realized on its stressed syllable
- empathic accent: stress related to semantic reasons
- unaccented words: "normal" stress
- reduced accent: less than usual stressed words

- Tune: rise or fall of F_0



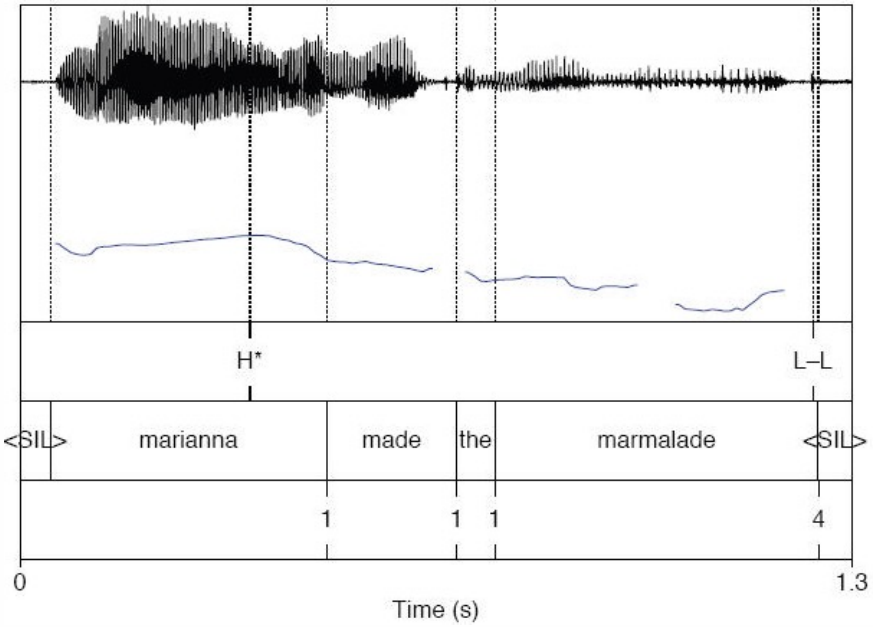
Prosodic analysis: ToBI

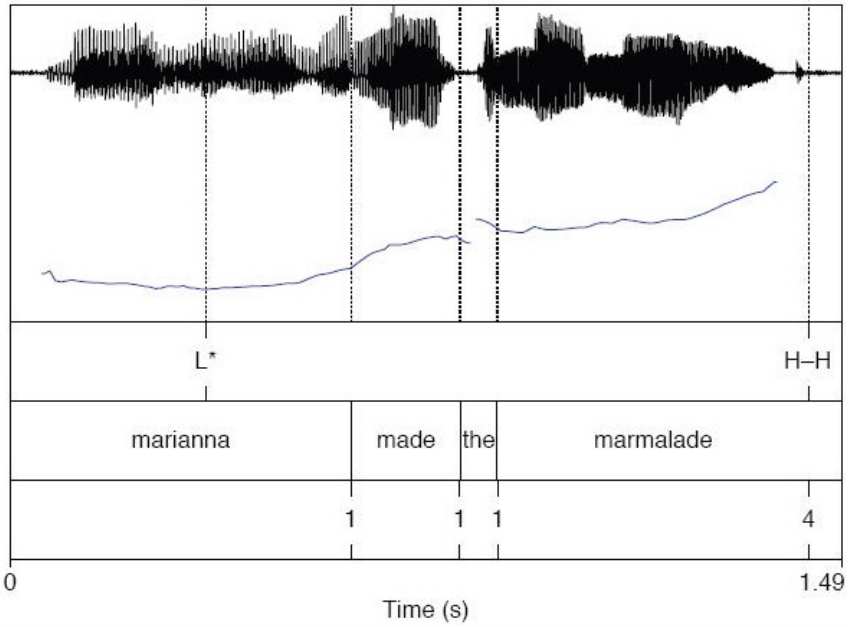
- ToBI: a phonological theory of intonation
  - Models prominence, tune, and boundaries
- Five pitch accents and four boundary tones
- Four phrasal breaks (intonational phrase, intermediate phrase, word breaks, word boundaries)

Pitch Accents		Boundary Tones	
H*	peak accent	**L-L%**	"final fall": "declarative contour" of American English
L*	low accent	**L-H%**	continuation rise
L*+H	scooped accent	**H-H%**	"question rise": cantonical yes-no question contour
L+H*	rising peak accent	**H-L%**	final level plateau (plateau because H- causes "upstep" of following)
H+!H*	step down		

Prosodic analysis: ToBI

- An example of ToBI annotation





Prosodic analysis: phone duration

- *Rules* that look at context and change typical duration of phone (Klatt, 1979) → factor weights f_i
  1. Prepausal Lengthening: The vowel or syllabic consonant in the syllable before a pause is lengthened by 1.4.
  2. Non-phrase-final Shortening: Segments which are not phrase-final are shortened by 0.6.
  3. …

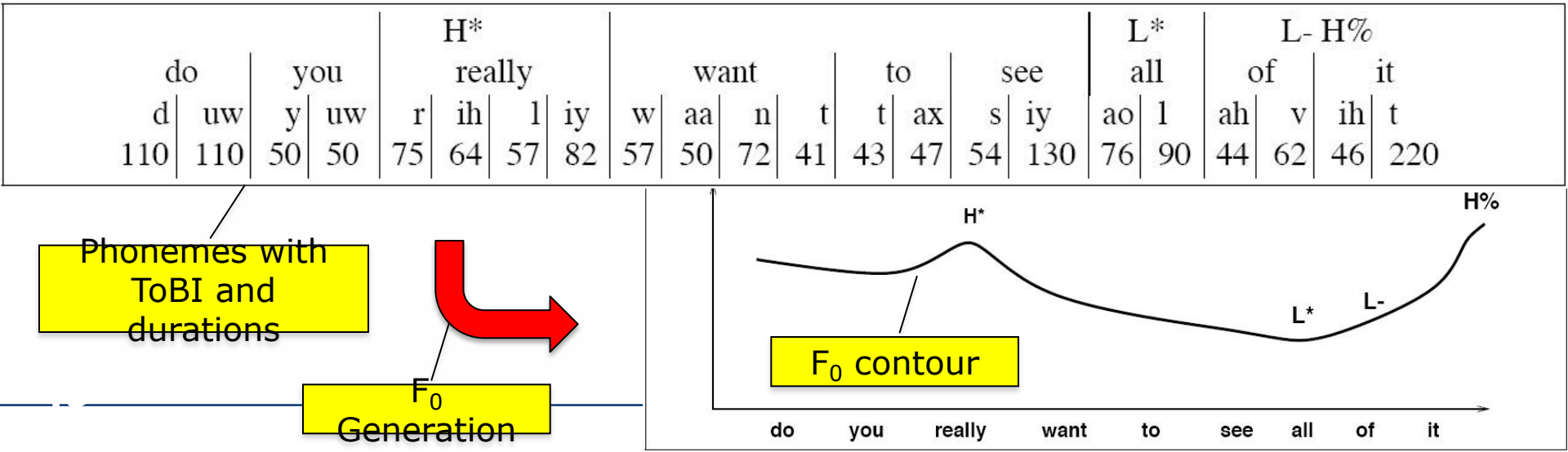
  Given the set of N factor weights f_i , the Klatt's formula for the duration d of a phone is:

$$d = d_{\min} + \prod_{i=1}^{N} f_i \times (\bar{d} - d_{\min})$$

- Or *Machine Learning*

Prosodic analysis

- From the annotation, F_0 is generated in two steps:
  - Defining "key" F_0 values for pitch accents and breaks
  - Generate the F_0 contour interpolating such F_0 values
- The "key" F_0 values are defined as a percentage of a reference F_0



2. Waveform synthesis

- Two models for concatenative waveform synthesis
  - Diphone synthesis
  - Unit selection synthesis

Diphone synthesis

ciao → /tS a o/ → (<s>, tS), (tS, a), (a, o), (o, </s>)

- Diphone: phone-like unit going from the middle of one phone to the middle of the following one
  - *Coarticulation* phenomenon: each phone differs slightly, depending on the preceding and following ones
- A diphone database is a collection of recorded diphones
  - Record a speaker saying one example of each diphone
  - Mark the boundaries of each diphones
  - Cut each diphone out and create a diphone database
- Synthesizing an utterance:
  - Grab relevant sequence of diphones from database
  - Concatenate the diphones; clean boundaries
  - Use signal processing to change the prosody

Unit selection synthesis

Database contains *units*:

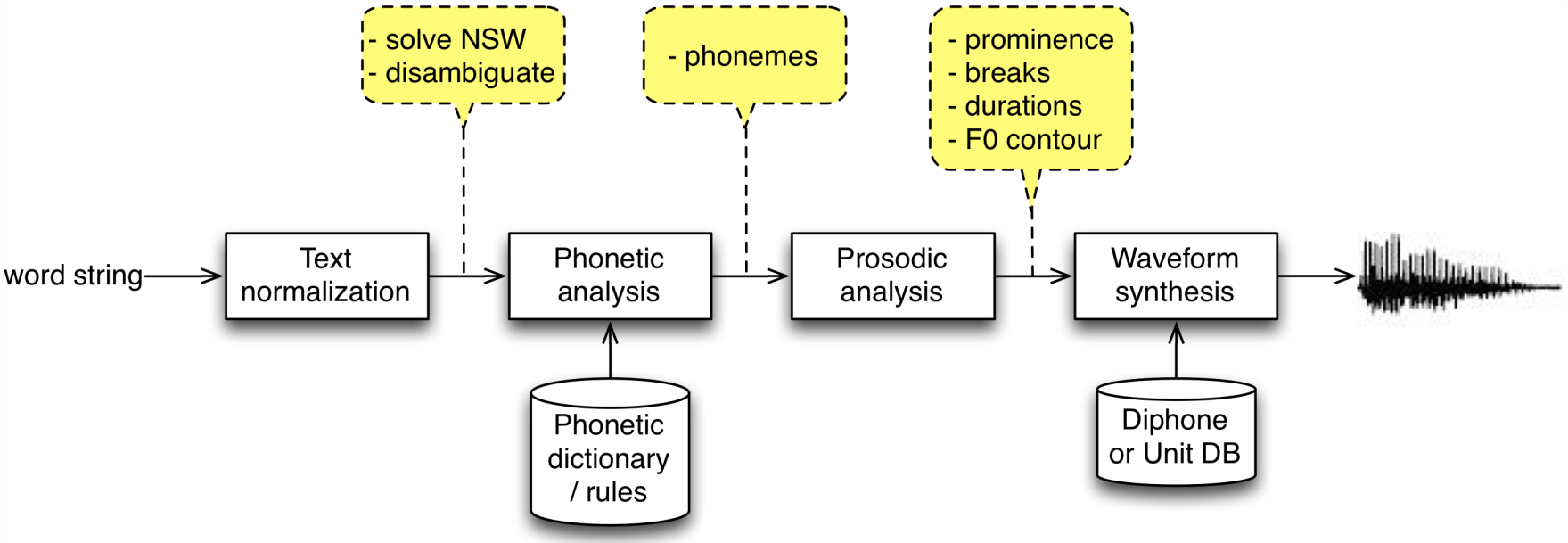
- any piece of speech that can be concatenated
  - Diphones, syllables, …

Given list of phonemes with prosodic annotation, find in DB best list of units → cost function:

- "Target cost": Closest match to the target description, in terms of
  - Phonetic context
  - F_0, stress, phrase position
- "Join cost": Best join with neighboring units
  - Matching formants + other spectral characteristics
  - Matching energy
  - Matching F_0

Solved using Viterbi or beam search

Recap


- solve NSW
- disambiguate
- phonemes
- prominence
- breaks
- durations
- F0 contour
word string
Text normalization
Phonetic analysis
Prosodic analysis
Waveform synthesis
Phonetic dictionary / rules
Diphone or Unit DB

Evaluation of text-to-speech

Requires human testers, checking:

- **Intelligibility**: ability of tester to correctly interpret meaning of utterance
  - Phone discrimination
  - Diagnostic Rhyme Test (DRT): intelligibility of initial consonants in standard word set
  - Modified Rhyme Test (MRT): identify words in a standard set
- **Quality**: measure of naturalness, fluency, clarity of the speech
  - Mean Opinion Score (score 1—5 given to each system to compare)
  - AB test: the same utterance is produced by the two system; testers choose the best system. Repeat for 50 utterances

Conclusions

Conclusions

TODO