AUTOMATIC SPEECH RECOGNITION

PETR FOUSEK, PAVEL KVETON

Created: 2019-04-23 Tue 14:06

TABLE OF CONTENTS

INTRO

FIRST COMMERCIAL ASR: RADIO REX (1920)

"A celluloid dog released by a spring when triggered with a 500 Hz acoustic energy (roughly the first formant of the vowel [eh] in “Rex”).

INSPIRATION BY HUMANS

"Humans evolved to optimally use the acoustic channel to communicate - why not to inspire by them?"

RELEVANT FINDINGS

- Fletcher (1950's)
- information about a phone spans 250-400ms
- 100ms not enough to tell phone in syllables
- 4-7 frequency channels min. for intelligibility, >10 for fidelity
- modulations about 2-10Hz (peak at 4Hz ~ 250ms)
- auditory cortex response also in 2 - 20Hz range

INFORMATION POINT OF VIEW

- standard speech: 64 kpbs (8kHz, 8bits)
- low-bitrate coding: 500 bps
- text: cca 50 bps

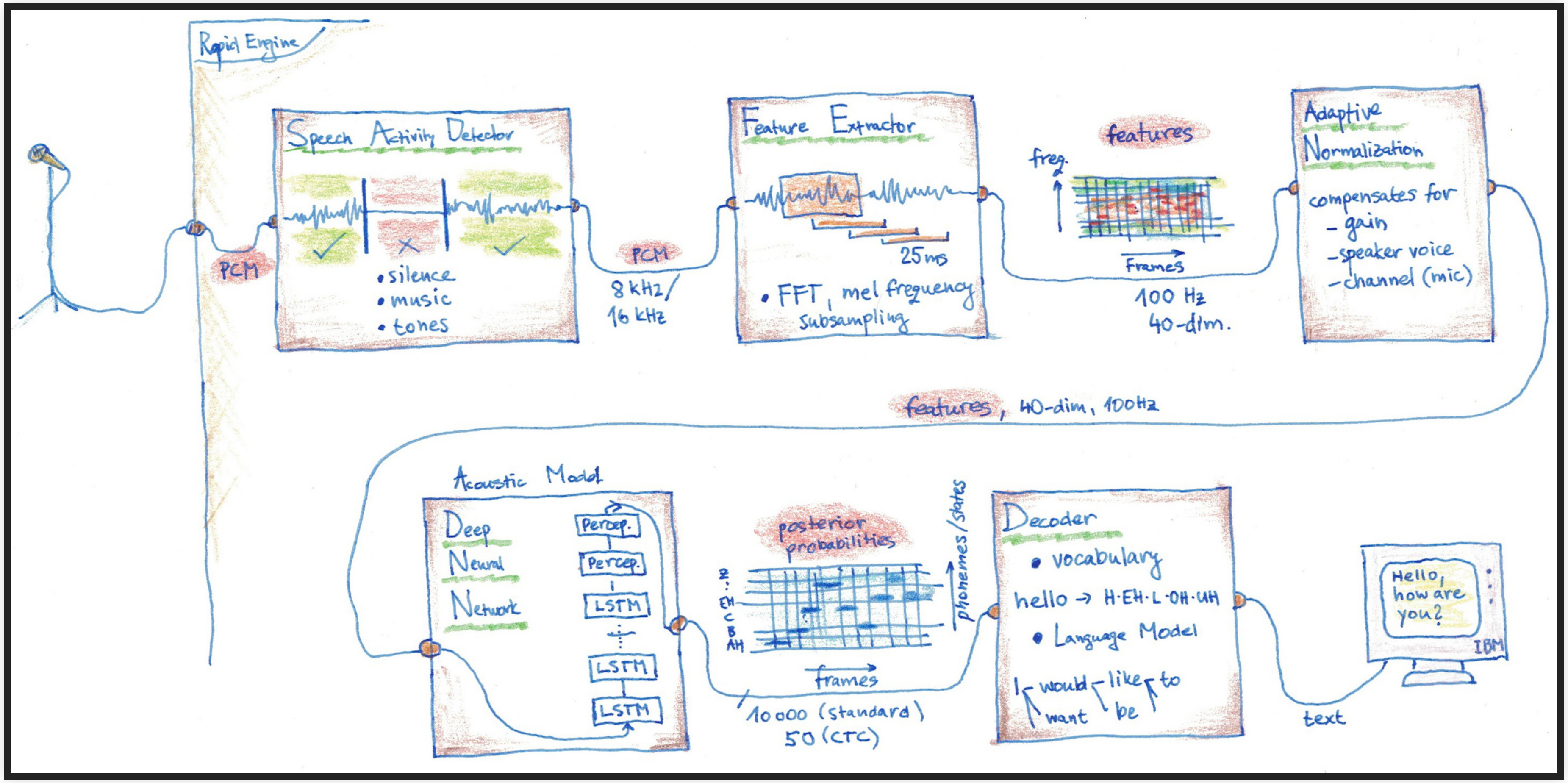
INFORMATION IN AUDIO

Lots of information irrelevant for ASR:

- speaker identity (gender, age, spk. style, dialect)
- speaker's state (emotions, health)
- environment (ambient sounds, reverb/echo of the room)
- distortions (noise, channel effects, distance)

Features for ASR should preserve lingustic content and suppress variability.

ASR PIPELINE



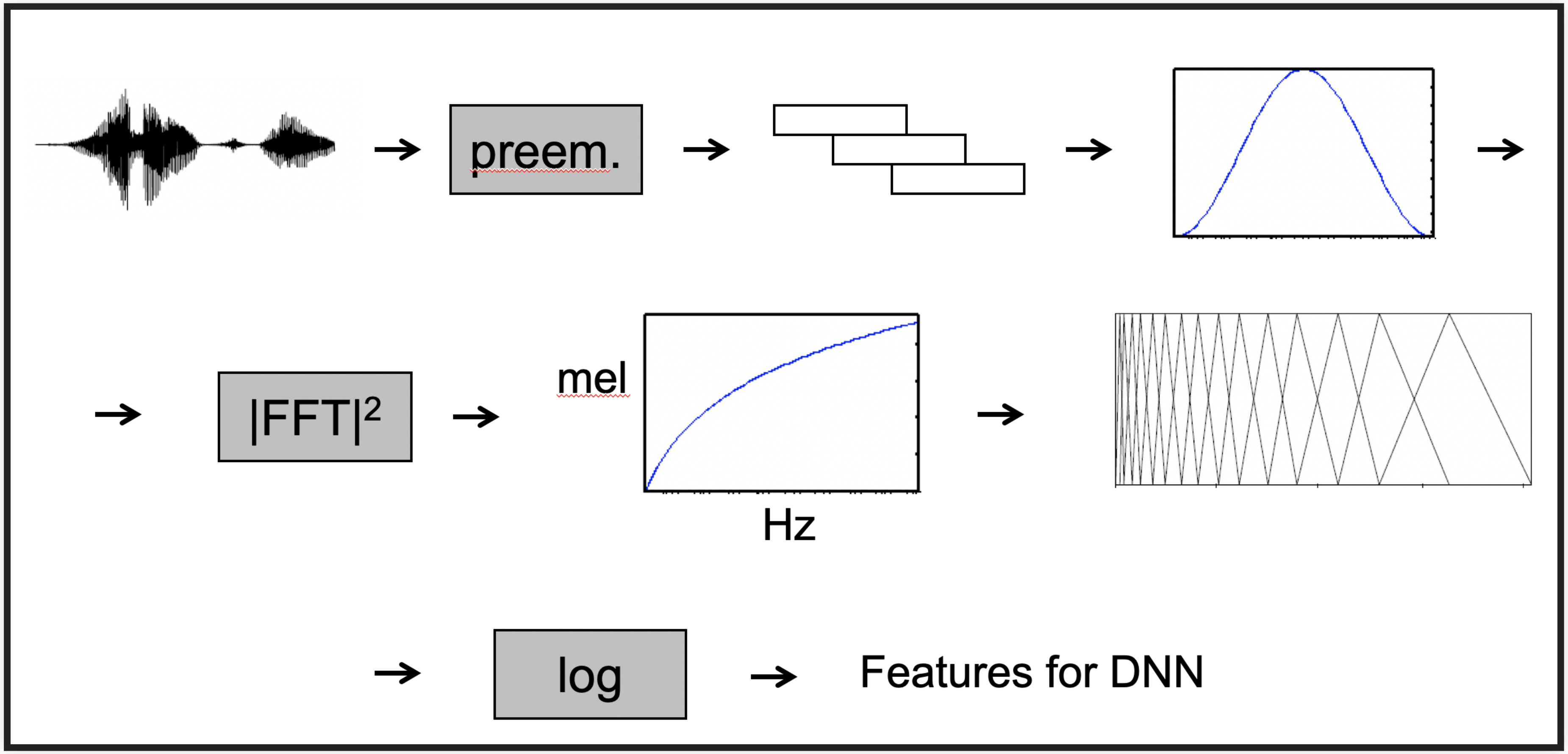
FRONT-END PROCESSING

INPUT AUDIO

- 1 channel
- 8kHz (telephony) or 16kHz (wide-band), 16-bit sampling

FEATURES

Spectral domain: Discrete Fourier Transform, overlapped windows



Features typically are 40-dim vectors sampled 100 times per second.

DISCRETE FOURIER TRANSFORM

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp \frac{-i2\pi kn}{N}$$

SPEECH DETECTOR

- Classify speech vs. non-speech
- Main goal: feed the recognition engine only with speech
- Principle:
    - signal-based (simple, less accurate)
    - Deep-learning-based (more CPU-expensive, more accurate)

NORMALIZATION, ADAPTATION, DE-NOISING

- Adaptive Normalization for audio gain and spectral tilt (channel), also DNNs like N(0,1) at input
- Vocal-Tract-Length-Normalization and more advanced techniques being obsoleted by DNNs
- De-noising also being obsoleted by multi-style training and larger models

ACOUSTIC MODELS

- Consume acoustic features and estimate likelihoods/probabilities of fixed acoustic classes
- Formerly generative models based on mixtures of Gaussians modeling states of context-dependent phone units
- Currently rather DNN models of context-dependent phone units (discriminative training)
- Moving towards simpler schemes and larger models (context-independent phones, graphemes or words)

TRAINING ACOUSTIC MODELS (DNNS) FOR ASR

- Need large volumes of (transcribed) audio data (10^{3+} hours)
- Training bootstrapped by ad-hoc alignment between audio and transcript
- Iterative process jointly refining the alignment and the model
- Advanced techniques allowing to use large volumes of untranscribed data
- Internally using highly parallelized Error-Backpropagation training
- Final DNNs allow adaptations

SPEECH DECODING

- input: acoustic features (40-dim every 10ms)
- output: written text
- why "decoder"? - the text has been encoded into acoustic signal (or in our case features), now we attempt to decode this information
- stochastic approach: a model needed

$$P(T|A)$$

(T=text, A=acoustics)

- $P(T|A)$ too complex (really? heading towards end-to-end models, but not yet), so:
- Bayes:

$$P(T|A) = \frac{P(A|T)P(T)}{P(A)}$$

- $P(A)$ constant, ignoring
- $P(A|T)$ = acoustic model P_A
- $P(T)$ = language model P_{LM}

ACOUSTIC MODEL $P_A(A|T)$

- here T is the "candidate" sequence of acoustic classes
- assuming independence per frame (if any dependence, it is covered by front-end: delta/double-delta, LSTM,…)
- i.e. $P_A(A|T) = \prod_i P_A(a_i|t_i)$ (i over the frames)
- a_i = the 40-dim feature vector
- t_i … … … "fixed acoustic classes", "phone units", …
- for simplicity - imagine t_i as a phone

LANGUAGE MODEL $P_{LM}(T)$

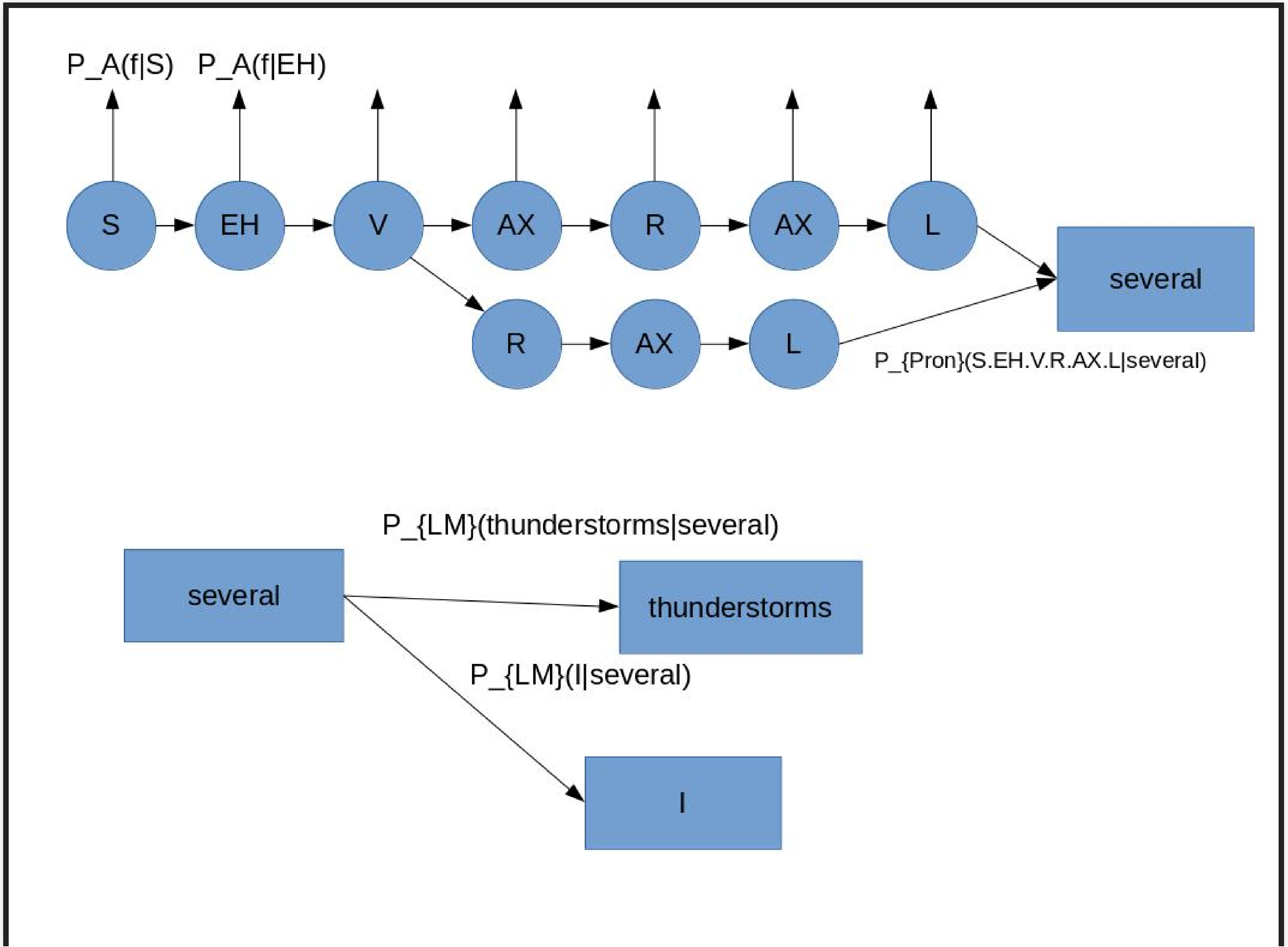
- here T = sentence, consists of words $w_1, \ldots, w_N$
- word sequences modelled with n-gram (or other) language models
- $P_{LM}(T) = \prod_i P(w_i | w_{i-1}, w_{i-2}, \ldots)$
- what is missing?

DICTIONARY

- glue between acoustic classes t_i and words - how words map to "acoustic classes"
- a word may have multiple pronunciations
  - several S EH V AX R AX L
  - several S EH V R AX L
- $P_{Pron}(pron|w)$
- pronunciation consists of phones (EH, V,…) - for us, these are the acoustic classes t_i
  - no model here

PUTTING IT TOGETHER

- homework: add the theory behind
- Hidden Markov Models - decoding hidden message (sentence) from its "visible" ("audible") encoded form
- combining WFSTs (acoustic model, dictionary, LM)
- dynamic programming (Viterbi) to find the best path through the (virtual) WFST given the stream of acoustic feature vectors: variety of approaches (one big WFST, dynamic on-the-fly construction, …)


P_A(f|S)
P_A(f|EH)
S
EH
V
AX
R
AX
L
R
AX
L
several
P_{Pron}(S.EH.V.R.AX.L|several)
P_{LM}(thunderstorms|several)
several
thunderstorms
P_{LM}(I|several)
I

OUT OF SCOPE

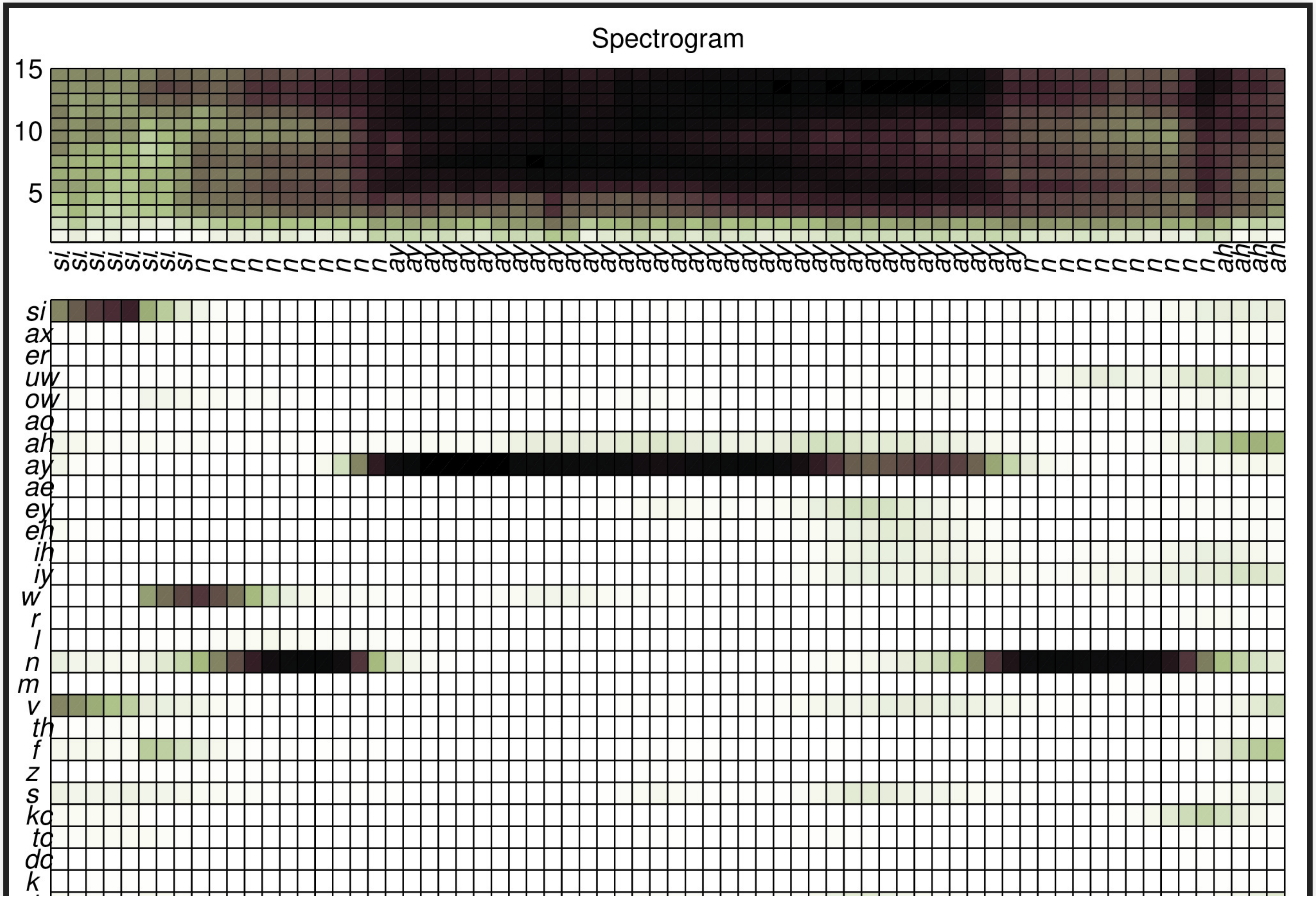
- puncutation, capitalization,…

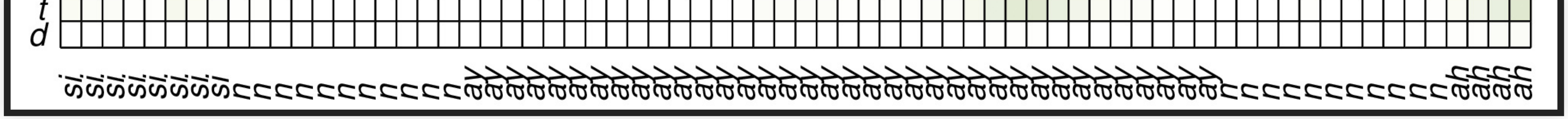
OUTRO

If you want to learn more and/or participate, come to IBM or contact your teachers! :)

EXTRA SLIDES

EXAMPLE DNN PHONE POSTERIOGRAM




t
d