Key ideas in speech and audio processing

Mark Hasegawa-Johnson

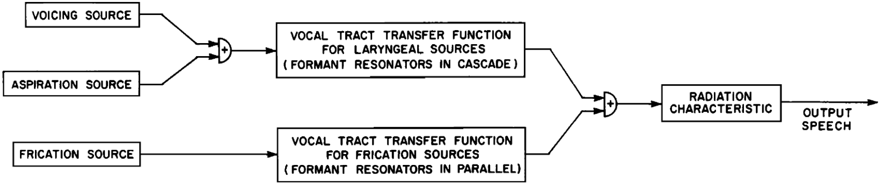
University of Illinois

January 27, 2021

Outline

1. Synthesis
   - Formant-Based
2. Recognition
   - Dynamic Time Warping
   - Hidden Markov Models
   - Weighted Finite State Transducers
   - Connectionist Temporal Classification
   - Listen, Attend and Spell
3. Emotion
   - OpenSmile

"Software for a Cascade/ Parallel Formant Synthesizer," Klatt, 1980



Key Idea: Intelligible speech can be synthesized using very simple, very cheap second-order filters, connected in cascade for vowels and glides, connected in parallel for consonants.

Key Equation

The key equation is just the equation for a second-order resonator:

$$
\begin{aligned}
y[n] &= Ax[n] + By[n-1] + Cy[n-2] \\
C &= -\exp(-2\pi BT) \\
B &= 2\exp(-\pi BT)\cos(2\pi FT) \\
A &= 1 - B - C
\end{aligned}
$$

- $x[n]$ = filter input, $y[n]$ = filter output
- A, B, C are the filter coefficients
- F and B are formant frequency and bandwidth, in Hertz
- $T = 1/F_s$ is the sampling interval

Key Results

- Describes realistic excitation functions for voicing, frication, and aspiration
- Describes relationship of the signal model to the physical vocal tract
- Tells you how to set the model parameters for every phoneme of English
- Provides complete FORTRAN code

Outline

1 Synthesis
- Formant-Based

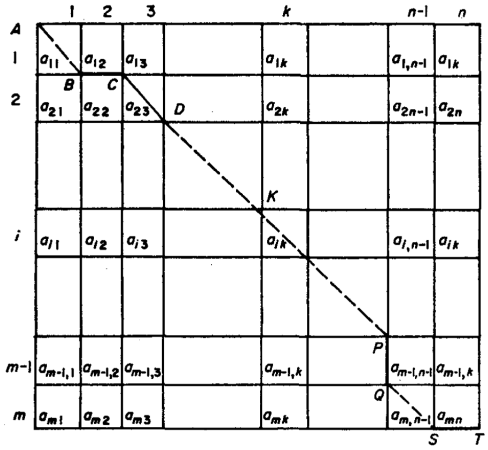
2 Recognition
- Dynamic Time Warping
- Hidden Markov Models
- Weighted Finite State Transducers
- Connectionist Temporal Classification
- Listen, Attend and Spell

3 Emotion
- OpenSmile

"Automatic Recognition of 200 Words," Velichko and Zagoruyko, 1970

Key Ideas:

- ASR is performed by comparing the test word, x, to a set of training words, x_1 through x_n, and output the label of the most similar recorded word.
- Similarity is computed by finding the permissible time alignment that makes them as similar as possible.



Key Equation

The similarity between two words is calculated backward in time, as

$$\textbf{Initialize: } A_{MN} = a_{MN}$$

$$\textbf{Iterate: } A_{mn} = \max\left[A_{m,n+1}, A_{m+1,n}, a_{mn} + A_{m+1,n+1}\right]$$

$$1 \le m \le M, \quad 1 \le n \le N$$

$$\textbf{Terminate: } A_{11} = \max\left[A_{12}, A_{21}, a_{11} + A_{22}\right]$$

where

- a_{mn} = similarity between the m^{th} test frame and the n^{th} training frame, and
- A_{11} = similarity between the whole test word, and the whole training word.

Key Results:

- Accuracy is about 95% on a speaker-dependent, 4-word vocabulary.
- Accuracy is "acceptable" for a speaker-dependent vocabulary of up to 200 words.

TABLE 1

Recognition results of speaker No. 1

N ts \ N cs	No. of errors 1	2	3	4	Recognition reliability
1		5	16	12	94·5
2	8		16	10	94·4
3	17	16		13	92·5
4	19	13	12		92·8

cs, Control sequence; ts, training sequence.

"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Rabiner, 1989

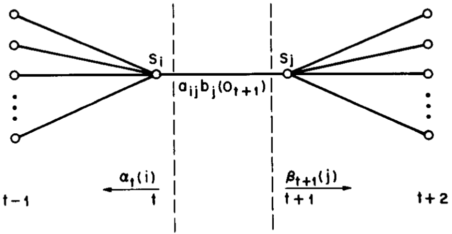


Fig. 6. Illustration of the sequence of operations required for the computation of the joint event that the system is in state S_i at time t and state S_j at time $t+1$.

Key Idea: Speech recognition = find the most probable word. Bayes' theorem allows us to compute this using a computationally efficient model.

Key Equation

The probability of the observation sequence $O = [O_1, \ldots, O_T]$ given the word λ can be efficiently computed as $p(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j)$, where

$$\alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}), \quad 1 \leq i, j \leq N, \ 1 \leq t \leq T-1$$

where

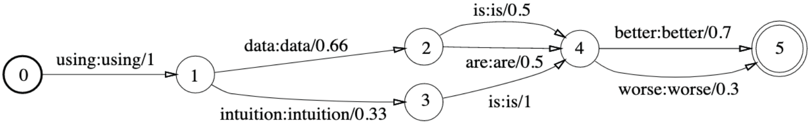
- N is the number of states in the word model, T is the number of frames in the spectrogram,
- $b_j(O_{t+1})$ is the probability of generating the observation O_{t+1} from state j,
- a_{ij} is the probability of a transition from state i to state j,
- $\alpha_t(i)$ is the probability of seeing all observations until time t, and reaching state i at time t.

Key Results

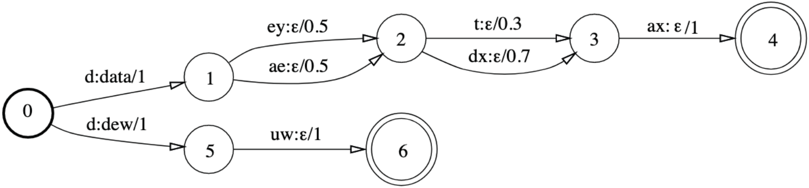
Demonstrates 3% WER for digit recognition. Reviews remarkable variety of theoretical results including:

- Mixture Gaussian models
- State-dependent linear dependence between consecutive spectra
- Explicit state duration densities
- KL Divergence between two HMMs
- Multiple observation sequences
- Sparse data issues

Weighted finite-state transducers in speech recognition, Mohri, Pereira and Riley, 2002



(a)



(b)

Key Idea: WFSTs can combine many different types of knowledge into a single probability distribution.

Key Equation

A WFST is a set of states Q, an initial state $i \in Q$, a set of final states $F \subseteq Q$, and a list of transitions T such that

$$t = (p[t], \ell_i[t], \ell_o[t], w[t], n[t]) \; \forall t \in T$$

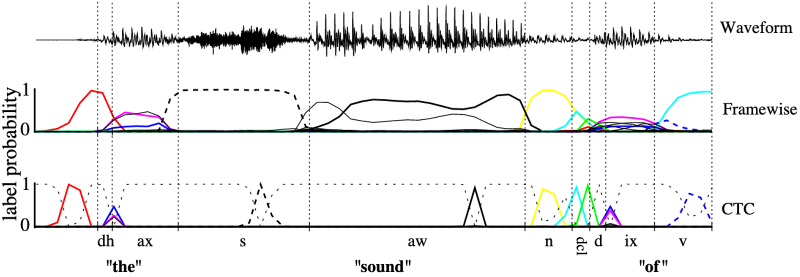
- $p[t] \in Q$ is the preceding state,
- $n[t] \in Q$ is the next state,
- $\ell_i[t]$ is the input label,
- $\ell_o[t]$ is the output label, and
- $w[t]$ is the weight, which is usually expressed either as a probability, or as a negative log probability.

Key Results

The four most important types of information for speech recognition are the four transducers H, C, L, and G:

- H maps from observation probability IDs (e.g., elements in the softmax output of a neural net) to triphone state IDs (e.g., third state in the triphone model of /k-æ+t/).
- C maps from triphone states to monophones (e.g., /ae/).
- L maps from monophones to words.
- G calculates word sequence probabilities.

Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, Graves, Fernandez, Gomez and Schmidhuber, 2006



Key Idea: Train a neural net to output the right sequence of labels, regardless of whether or not the labels occur at the right times.

Key Equation

The probability of observing character sequence $\mathbf{l}$ given acoustic sequence $\mathbf{x}$ is

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} \prod_{t=1}^{T} y^t_{\pi_t},$$

where

- S is the length of the character sequence ($\mathbf{l}$), T is the length of the acoustic sequence ($\mathbf{x}$), and $T \geq S$,
- $\mathcal{B}^{-1}(\mathbf{l})$ is a set of time-aligned label sequences π, each of length T (time-aligned to the audio), but containing exactly the same labels as $\mathbf{l}$, separated as necessary by arbitrarily placed "blanks," and
- $y^t_{\pi_t}$ is the neural net's estimated probability of label π_t at time t.

Key Results

- CTC can be computed efficiently using math that's very similar to an HMM.
- A network trained using CTC converges reasonably quickly.
- CTC outperforms a hybrid DNN-HMM on the TIMIT database.

Listen, Attend and Spell, Chan, Jaitly, Le and Vinyals, 2015

Key Idea: Characters are produced by a decoder ("speller"), each of whose inputs is a weighted sum ("attend") of the encoder states ("listener").

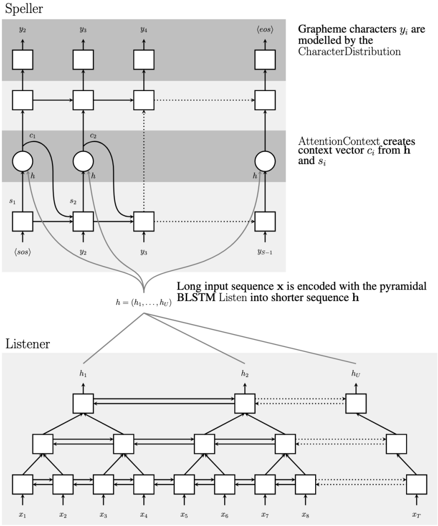


Figure 1: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence **x** into high level features **h**, the speller is an attention-based decoder generating the **y** characters from **h**.

Key Equation

The i^{th} output character, y_i, is produced by a neural network dependent on state vector s_i and context vector c_i, where

$$c_i = \sum_u \alpha_{i,u} h_u, \quad \alpha_{i,u} = \frac{\exp(e_{i,u})}{\sum_u \exp(e_{i,u})}, \quad e_{i,u} = \phi(s_i)\dot{\psi}(h_u),$$

where

- h_u is the "listener" state vector at the u^{th} encoder frame,
- s_i is the "speller" state vector at the i^{th} output character,
- $\phi(s_i)$ and $\psi(h_u)$ are multilayer perceptrons that transform s_i and h_u so that their dot product is a useful measure of their relevance to one another.

Key Result

With 2 million training utterances, LAS + Sampling + language model rescoring achieves 10.3% word error rate, compared to 8.0% for a state of the art hybrid system (convolutional-LSTM-deep neural network-HMM, or CLDNN-HMM).

Outline

1 Synthesis
- Formant-Based

2 Recognition
- Dynamic Time Warping
- Hidden Markov Models
- Weighted Finite State Transducers
- Connectionist Temporal Classification
- Listen, Attend and Spell

3 Emotion
- OpenSmile

OpenSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor, Eyben, Wöllmer, and Schuller, 2010

Key Idea: In many audio classification problems (e.g., emotion), we don't really know which acoustic features are useful. In that case, it's better to generate a huge variety of possibly-useful features, and then use feature-selection algorithms or sparse learners to sort through them.

Key Decomposition: LLDs × Functionals

Feature Group	Description
Waveform	Zero-Crossings, Extremes, DC
Signal energy	Root Mean-Square & logarithmic
Loudness	Intensity & approx. loudness
FFT spectrum	Phase, magnitude (lin, dB, dBA)
ACF, Cepstrum	Autocorrelation and Cepstrum
Mel/Bark spectr.	Bands 0-N_{mel}
Semitone spectr.	FFT based and filter based
Cepstral	Cepstral features, e.g. MFCC, PLP-CC
Pitch	F_0 via ACF and SHS methods Probability of Voicing
Voice Quality	HNR, Jitter, Shimmer
LPC	LPC coeff., reflect. coeff., residual Line spectral pairs (LSP)
Auditory	Auditory spectra and PLP coeff.
Formants	Centre frequencies and bandwidths
Spectral	Energy in N user-defined bands, multiple roll-off points, centroid, entropy, flux, and rel. pos. of max./min.
Tonal	CHROMA, CENS, CHROMA-based features

Table 1: openSMILE's low-Level descriptors.

Category	Description
Extremes	Extreme values, positions, and ranges
Means	Arithmetic, quadratic, geometric
Moments	Std. dev., variance, kurtosis, skewness
Percentiles	Percentiles and percentile ranges
Regression	Linear and quad. approximation coefficients, regression err., and centroid
Peaks	Number of peaks, mean peak distance, mean peak amplitude
Segments	Number of segments based on delta thresholding, mean segment length
Sample values	Values of the contour at configurable relative positions
Times/durations	Up- and down-level times, rise/fall times, duration
Onsets	Number of onsets, relative position of first/last on-/offset
DCT	Coefficients of the Discrete Cosine Transformation (DCT)
Zero-Crossings	Zero-crossing rate, Mean-crossing rate

Table 2: Functionals (statistical, polynomial regression, and transformations) available in openSMILE.

Key Results

- **Key Benefits:**
  - OpenSmile is used in the baseline system for every Interspeech Parlinguistic challenge, every year. Competing submissions usually use OpenSmile plus some modification, often getting only a few percent improvement.
  - An MLP applied to OpenSmile features is often an excellent and hard-to-beat classifier, for any task that involves classifying audio segments of a few seconds in duration.
- **Key Problems:**
  - Overgeneration means it's hard to interpret: the "best" feature in their 6000-feature set might be very little better than the "second-best" feature, and it's not clear why.
  - Designed for classification; not clear how to use it for sequence recognition.