Part II: Speech

Notes on Natural Language Processing

Chia-Ping Chen

Department of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan ROC

Introduction

- In this part of natural language processing, we study **speech** from a computational perspective.
- We will touch upon the following subjects
  - **phonetics**
  - **speech recognition**
  - **computational phonology**
  - **speech synthesis**

Phonetics

- "*Phonetics is the study of linguistic sounds, how they are produced by the articulators of human vocal tract, how they are realized acoustically, and how this acoustic realization can be digitized and processed.*"
- Phonetics includes the following studies
  - **phonetic alphabets**
  - **articulatory phonetics**
  - **acoustic phonetics**

Phonetic Alphabet

- A *spoken* word can be decomposed into a string of "basic" **units of speech**.
- A set consisting of such basic units is called a **phonetic alphabet**.
- An element in a phonetic alphabet is called a **phone**.
- Using a symbol for each distinct phone, a spoken word can be **transcribed** as a string of such symbols.
- An **utterance** is a string of spoken words. It can be transcribed by an phonetic alphabet.

Common Phonetic Alphabets

- **The international phonetic alphabet** (IPA): originally developed by International Phonetic Association in 1888, with the goal of transcribing the speech of all human languages!
- **ARPAbet**: a phonetic alphabet designed for American English *only*. It uses *ASCII* symbols for phones.
- Examples are given in Figure 7.1.

Opaque vs. Transparent

- A **word** is represented by a string of **letters**. It is called the **orthography** of the word.
- A **spoken word** is represented by a string of **phones**. It is called the **pronunciation** of the word.
- The mapping between a letter and a phone may be **transparent** or **opaque**, depending on the languages.

Articulatory Phonetics

- **Articulatory phonetics** is the study of how the sound for each phone in the phonetic alphabet is produced, as the motion of organs in the **vocal tract**, including
  - **lung**
  - **trachea (windpipe)**
  - **larynx (Adam's apple)**
  - **vocal folds (vocal cords)**
  - **glottis (space between folds)**
  - **mouth (tongue, teeth, lips, palate, velum)**
  - **nose**
- Figure 7.3.

Consonants: Place of Articulation

- Airflow is restricted during the production of phones.
- Where this restriction takes place is called the **place of articulation**. Shown in Figure 7.4.
- For consonants, we have the following attributes for the place of articulation
  - **labial:** `[p][b]`
  - **labiodental:** `[f][v]`
  - **dental:** [θ]
  - **alveolar:** `[t][d][s][z]`
  - **palatal:** [$\int$]
  - **velar:** `[k][g]`
  - **glottal**

Consonants: Manner of Articulation

- **How** the airflow is restricted during the production of a phone is called the **manner of articulation**.
- For consonants, we have the following attributes for the manner of articulation
  - **stop (plosive)**: airflow completely blocked, `[t][d][p][b][k][g]`
  - **fricative**: airflow restricted but not completely blockes, `[s][z]`
  - **affricate**: stop + fricative, `[t∫]`
  - **tap (flap)**: quick motion of tongue against alveolar ridge, lo<u>t</u>us, ki<u>tt</u>en
  - **nasal**: airflow passing through nasal tract, `[m][n]`

Vowels

- Vowels can also be characterized by the position and motion of articulators.
- Commonly used parameters are
  - **height**
  - **frontness (backness)**
  - **rounded**
- According to these parameters, we have
  - *front* vowels `[iy]`, `[ey]`, **back** vowels `[uw]`
  - **high** vowels `[iy]`, `[uw]`, **low** vowels [a]
  - **rounded** vowels, `[u]`.
- “vowel space”, Figure 7.6

Syllables

- A **syllable** must contain a central **vowel**, and optinal surrounding consonants.
- Specifically, a syllable consists of
  - **nucleus**: the central vowel (mandatory)
  - **onset**: a sequence of consonants in front of the nucleus (optional)
  - **coda**: a sequence of consonants following the nucleus (optional)
- The **rime** of a syllable is the nucleus plus coda.

Accentuation and Lexical Stress

- In a natural *sentence* of English, some words are more **prominent** than others, due to their grammatical roles.
- They are said to be **accented**.
- This is also known as **pitch accent** in the sentence level.
- When an accented word is multi-syllabic, the syllable that is accented is said to has the **lexical stress**.
- Lexical stress are often labeled in a pronunciation dictionary.

Pronunciation Variation

- The production of a phone does vary from one instance to another.
- This is called **pronunciation variation**.
- The **context** of a phone differs.
- For example, a vowel can be accented or reduced depending on the context. Consonants have similar behavior.
- We will look at **local** effects induced by neighboring phones. But note that semantic and syntactical factors are also important.

Phonetic Features

- The pronunciation variation can be described by **phonetic features**.
- A phonetic feature corresponds to a property. For a given feature, a phone has value 1 or 0. E.g.,
  - voice
  - labial
  - nasal
  - $\ldots$
- Equivalently, a feature value can be non-binary.

Phonological Rules

- A **phonological rule** is denoted by

$$/A/ \longrightarrow B \ / \ C \_\_ D,$$

where A, B, C, D are phone classes.

- It characterizes the relation between a phone class (A) and its realization (B) for a specific context (C and D).

Acoustic Phonetics

- Physically, speech is a wave of air **pressure**.
- Basic notions of the **acoustic waveform** of speech
  - **frequency** of a waveform
  - **spectrum**
  - **sampling** of speech waveforms
  - **Nyquist** frequency, rate

Time-domain Quantities

- The **power** P of a segment of N samples is defined by

$$P = \frac{1}{N} \sum_{i=1}^{N} |x_i|^2$$

- The **root-mean-square** amplitude is

$$\text{RMS} = \sqrt{P}$$

- The **intensity** is defined by

$$\text{Intensity} = 10 \log_{10} \frac{P}{P_0}$$

Spectral Analysis

- A signal is often described as a function of **time**.
- It can also be described as a function of **frequency**.
- The transformation from a function of time to a function of frequency is called **spectral analysis**.
- The **spectrum** of a signal is a function of frequency. It is obtained through the **Fourier transform**, and indicates the distribution of energy over frequency.

Basic Terms

- **fundamental frequency**: the frequency of vocal fold vibrations, also called **F0**
- **pitch track (contour)**: the plot of F0 over time (Fig. 7.15)
- The term **pitch** refers to a perceptual measure correlated to the fundamental frequency.
- The perceptual measure of **loudness** is related to the signal intensity.

Spectrogram

- A **spectrogram** shows how the spectrum of a speech waveform changes over time.
- Refer to Fig. 7.23. It is a time-frequency plot.
- In order to obtain spectrogram, it is essential to **window** the waveform and apply **short-time Fourier transform**.
- **Dark** points have high amplitudes.

Source-Filter Model

- A dark strip (spectral peak) in a spectrogram is called a **formant**.
- Formant frequencies can be used to identify vowels.
- Imagine a **source-filter** model where the source is the pulses by vocal folds and the filter is the vocal tract.
- The formants are the resonant frequencies.
- Different vocal-tract configurations of different vowels lead to different formant frequencies.

Computational Resources

- **Pronunciation dictionaries**
  - *CELEX, CMUdict, PRONLEX*
  - CMUdict: 125k wordforms, ARPAbet, stress marked for vowels
- **Phonetically annotated corpus**
  - *TIMIT, Switchboard*
  - TIMIT: 6300 utterances from 2300+ sentences by 630 speakers, time-aligned transcription (at the phone-level)
- **Phonetic software tools**

Automatic Speech Recognition

- *“The task of speech recognition is to take as input an acoustic waveform and produce as output a string of words.”*
- Noisy-channel model
  - The acoustic waveform representation (channel output) is a noisy version of a string of words (channel input).
- HMM model (of generation)
  - The model from a string of words to its acoustic waveform representation is a random process described by hidden Markov models.

Decoding Problem

- From waveform representation to word string is also called *decoding*.
- The acoustic input is often processed to be a sequence of “observations”, say

$$O = o_1, \ldots, o_t.$$

- The decoding problem can be written as

$$\begin{aligned}\hat{W} &= \arg\max_W P(W|O) = \arg\max_W P(W,O) \\ &= \arg\max_W P(O|W)P(W)\end{aligned}$$

- O is often not the waveform samples, but *features*.

Feature Extraction

- **sampling**
- **pre-emphasis**
- **windowing**
- **discrete Fourier transform**
- **Mel-frequency filter bank**
- **taking logarithm**
- **cepstrum**
- **dynamic features**
- **log energy**

Sampling

- We sample from speech waveform with period T

$$x[n] = x_c(nT), n = 1, 2, \ldots, T.$$

- If the sampling period T is sufficiently short, then $x[n]$ is a fair representation of $x_c(t)$.
- In fact, if $x_c(t)$ is bandlimited, $x[n]$ is *exact*, meaning it contains all information about $x_c(t)$.

Pre-Emphasis

- There is more energy at the low-frequency range than at the high-frequency range in voiced segments, such as vowels.
- **Pre-emphasis** boosts the high-frequency energy.
- This is often done by a filter

$$y[n] = x[n] - \alpha x[n-1], \quad 0.9 < \alpha < 1.$$

Windowing

- Use a **window function** to extract a segment of $x[n]$ for processing, then moves the window forward.
- A simple window function is the **rectangular window**

$$w[n] = \text{rect}_L[n] = \begin{cases} 1, & 0 \leq n \leq L-1 \\ 0, & \text{otherwise.} \end{cases}$$

- A window whose coverage starts at sample m is

$$w_m[n] = w[n-m].$$

Clearly it is non-zero only for $m \leq n \leq m + L - 1$.

Hamming Window

- Instead of the simple rectangular window, the **Hamming window** is often used as the window function

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L}\right), & 0 \le n \le L-1 \\ 0, & \text{otherwise.} \end{cases}$$

- A Hamming window has the same span, but is smoothier.

Frame

- Each window of signal is called a **frame**.
- In the previous example, a frame has a **size** of L.
- The difference of the starting positions of a frame and the next frame is called **frame shift**.
- Suppose the frame shift is S. The lth frame uses the window function of

$$w[n - lS].$$

- Frame size and frame shift need not be equal. Often the frame shift is smaller to have *overlapping* frames.

Discrete Fourier Transform

- For each frame, the **discrete Fourier transform** (DFT) is applied,

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}},\ 0 \leq k \leq N-1.$$

- DFT produces samples of the Fourier transform of $x[n]$, equally spaced at points $\omega_k = \frac{2\pi k}{N}$.
- Note that the Fourier transform of $x[n]$ is related to the Fourier transform of $x_c(t)$.
- So $X[k]$ is related to the short-time spectrum of that window of signal.

Spectrogram

- The DFT of a frame represents the spectral information of the frame.
- Plotting frame-DFT over time, we have a 2-dimensional (time-frequency) representation of spectral information. This is called **spectrogram**.
  - wide-band spectrogram: a small frame size (≤ 10 ms) has a fine time resolution and coarse frequency resolution
  - narrow-band spectrogram: a large frame size (> 20 ms)

Mel-Frequency Filter Bank

- The **mel-scale** of a frequency f is defined by

$$\text{mel}(f) = 1127 \log(1 + \frac{f}{700}).$$

- To decide the mel-frequency filters
  - decide the number of filters, say M
  - decide the overlapping ratio
  - decide the mel-scales of the lowest and highest frequencies
  - decide the central frequencies such that
    - the filters are equally spaced in the mel-scale;
    - the entire frequency range is covered.

Binning and Logarithm

- Using a mel-scale filter as **bin**, the DFTs with frequencies falling within that filter are integrated

$$B[m] = \sum_{k \in S_m} |X[k]|.$$

  - The squared magnitudes can also be used.
- The sum in each bin is taken logarithm

$$\hat{X}[m] = \log B[m].$$

MFCC

- The **cepstrum** of a frame is the **inverse discrete Fourier transform** of the logarithm of mel-bin outputs

$$c[p] = \frac{1}{M} \sum_{m=0}^{M-1} \hat{X}[m] e^{j2\pi mp/M}.$$

- **liftering**
- **truncation**
- $c[p]$'s are called **MFCC**, the mel-frequency cepstral coefficients.

Log Energy

- The **log energy** of the lth frame is simply

$$\xi[l] = \log\left(\sum_{0 \le n \le L-1} x^2[lS+n]\right).$$

Dynamic Features

- The MFCC or the log energy of a frame is called the **static features**.
- We can enhance the feature vectors by including the **dynamic features**.
- The simplest are the **delta features**, defined by

$$\Delta f[n] = \frac{f[n+1] - f[n-1]}{2}.$$

  - Other formulas for delta feature exist.
  - The delta features of delta features, called the **acceleration features**, are often used as well.

Gaussian Mixture Models

- Uni-variate Gaussian

$$N(x;\mu,\sigma^2):\; p(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multi-variate Gaussian

$$N(\mathbf{x};\mu,\Sigma):\; p(\mathbf{x})=\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}{2}}$$

- Gaussian Mixture Model

$$p(\mathbf{x})=\sum_k c_k N(\mathbf{x};\mu_k,\Sigma_k).$$

Embedded Training

- **Training** means the learning the parameters by the recognizer.
- Each basic HMM corresponds to a linguistic unit, such as a phone or a word.
- With labeled data to the level of basic HMM units, each HMM can be trained by the corresponding data.
- However, this is often a *luxury*. What we have is a label for the entire utterance.
- The **embedded training** is an method in this scenario: to use higher-level label to train lower-level HMMs.

Flat Start

- Basic ideas of embedded training
  - assume parameter set
  - accumulates data statistics based on label
  - update parameter set
- In order to start this iterative process, we need to have an initial estimate.
- The **flat start** method is used in such initialization.
  - for initial Gaussian parameters, use global mean and variance
  - for initial Markov chain parameters, use uniform probability for allowed transitions

Viterbi Training

- In an **exact** embedded training, the complete data statistics of **all possible state sequences** are accumulated for parameter updates.
- In a **Viterbi** embedded training, the data statistics is only computed on the **Viterbi path**, which is the most probable one.
- The Viterbi path actually defines an **alignment** between linguistic units and speech segments.
- This alignment is called a **forced alignement** as the speech is forced to be aligned to a given label.

Evaluation

- word error rate

$$\text{WER} = \frac{I + S + D}{N} * 100\%,$$

  - I: the number of insertions
  - D: the number of deletions
  - S: the number of substitutions
  - N: the number of word tokens in the reference
- based on an optimal alignment

Advanced Topics in ASR

- N-Best List and Word Lattice
- Stack Decoding
- Context-Dependent Models
- Tree-Strutured Lexicon

Re-Scoring

- An ASR system can be designed to create **multiple hypotheses**, instead of just one.
- These hypotheses are subject to further mechanism to decide which is the best. The is called **rescoring**.
- More sophisticated models are used for rescoring purpose, to pick a better candidate.

N-Best List and Word Lattice

- A list of the top-N candidates is called an **N-best list**.
- An alternative way is to use a **word lattice**, which represents a set of word sequences in a **directed graph**.
  - an arc is labeled by a word, along with other useful information
  - a node is labeled by a time mark

Oracle Error Rate

- By definition, the **oracle error rate** of multiple hypotheses is the error rate when the best hypothesis is used.
  - Same definition for an entire test set, where the best hypothesis is used for each utterance.
- When using a word lattice to represent multiple hypotheses, it is also called **lattice error rate**.
- It represents the **lower bound** on error rate for any rescoring mechanism.

Word Graph

- A **word graph** is a simplified version of a word lattice.
  - removing the time information
  - merging overlapped copies
  - Fig. 10.3, 10.4
- *vastly* restricting the search space

Confusion Networks

- The **word posterior probability** represents the system's **confidence** of about a specific word.
- It can be computed by a normalized probability based on competing hypotheses.
- A **confusion network** has a word and the confusable words in a section of sectioned graph. Fig. 10.5.
  - nicknamed "sausage"
- An arc is labeled by word identity and its posterior probability.
- The word probability normalization is based on sentence probability.

Context-Dependent Models

- We will use **phone** models for illustration.
- The acoustic realization of a **phone** does depend on its **context**, that is, the adjacent phones have influence.
- Instead of a single model, we can use a different model for each different phone context.

$$\text{i} \;\longrightarrow\; \text{b-i+p},\;\; \text{k-i+k},\;\; \ldots$$

State-Tying

- The number of models will increase significantly, as well as the number of states.
- **State-tying** can be then applied to reduce the total number of states. Fig. 10.13.
  - Suppose there are three states per triphone HMM.
  - The first states of the triphone models with the same middle phone are tied.
  - Similarly for the other states.

Stack Decoding

- Imagine speech decoding as a graph search problem.
- A hypothesis prefixed by p has a score

$$f(p) = g(p) + h(p),$$

  where $g(p)$ is the best score from root to p, and $h(p)$ is an estimate for completing the best hypothesis prefixed by p.
- A **priority queue**, using $f(p)$ as the priority, is used to store active hypotheses.

Optimality

- In **stack decoding**, the current best hypothesis in the stack is extended to generate new hypotheses, which are then pushed back according to its priority.
- If $h(p)$ is an **upper bound** for the residual score, a completed hypothesis p^* at the top of stack is *optimal*.
- That is, no hypothesis in the queue can have a better score when it is finished

$$g(p^*) = f(p^*) \geq f(q) + h(q) \geq g(q+).$$

where q is a hypothesis in the queue and $q+$ is the best completed hypothesis prefixed by q.

Tree-Structured Lexicon

- The pronunciation lexicon can be arranged in a **tree** such that each edge is either an HMM, or NULL
  - HMM: acoustic unit
  - NULL: word end (with word ID)
- The acoustic model of a word w is the label of a path from root to the word-end node of w.
  - efficiency: less node/edges for the entire lexicon
  - deficiency: word id is not known until a word-end node is met
- During search, each **language model state** requires a lexicon tree copy.