Sound and Human Speech

Developing and understanding Automatic Speech Recognition systems is an inter-disciplinary activity, taking expertise in linguistics, computer science, and electrical engineering.

This course will focus on the structure of American English speech. Other languages may differ in more or less significant ways, from the use of tone to convey meaning, to the sets of meaningful distinctions in the sound inventory of the language.

Phonetics

Phonetics is the part of linguistics that focuses on the study of the sounds produced by human speech. It encompasses their production (through the human vocal apparatus), their acoustic properties, and perception. There are three basic branches of phonetics, all of which are relevant to automatic speech recognition.

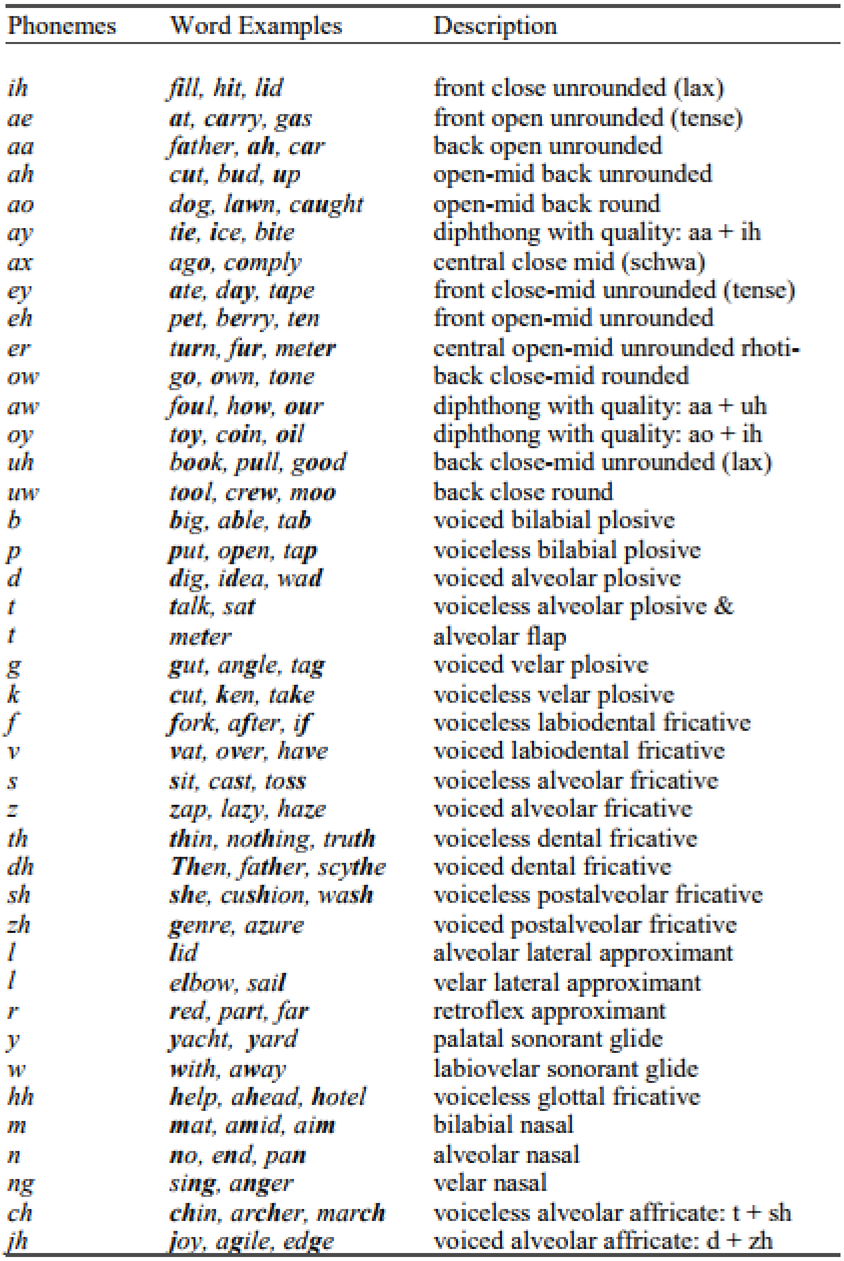
- Articulatory phonetics focuses on the production of speech sounds via the vocal tract, and various articulators
- Acoustic phonetics focuses on the transmission of speech sounds from a speaker to a listener
- Auditory phonetics focuses on the reception and perception of speech sounds by the listener.

The atomic unit of speech sound is called a phoneme. Words are comprised of one or more phonemes in sequence. The acoustic realization of a phoneme is called a phone. Below is a table of phonemes of U.S. English and common realizations.

One major way to categorize phonemes is into vowels and consonants.

Vowels can be distinguished by two attributes. First, they are voiced sounds, meaning that the airflow from the vocal chords into the mouth cavity is created by the vibration of the vocal chords at a particular fundamental frequency (or pitch). Second, the tongue does not in any way form a constriction of air flow during production. The placement of the tongue, lips, and jaw distinguishes different vowel sounds from each other. These different positions form different resonances inside the vocal tract called formants and the resonant frequencies of these formants characterizes the different vowel sounds.

Consonants are characterized by significant constriction of air flow in the airway or mouth. Like vowels, some consonants can be voiced, while others are unvoiced. Unvoiced phonemes do not engage the vocal cords and therefore do not have a fundamental frequency or pitch. Some consonant phonemes occur in pairs that differ only in wether they are voiced or unvoiced but are otherwise identical. For example, the sounds /b/ and /p/ are have identical articulatory characteristics



Phonemes	Word Examples	Description
ih	*fill, hit, lid*	front close unrounded (lax)
ae	*at, carry, gas*	front open unrounded (tense)
aa	*father, ah, car*	back open unrounded
ah	*cut, bud, up*	open-mid back unrounded
ao	*dog, lawn, caught*	open-mid back round
ay	*tie, ice, bite*	diphthong with quality: aa + ih
ax	*ago, comply*	central close mid (schwa)
ey	*ate, day, tape*	front close-mid unrounded (tense)
eh	*pet, berry, ten*	front open-mid unrounded
er	*turn, fur, meter*	central open-mid unrounded rhoti-
ow	*go, own, tone*	back close-mid rounded
aw	*foul, how, our*	diphthong with quality: aa + uh
oy	*toy, coin, oil*	diphthong with quality: ao + ih
uh	*book, pull, good*	back close-mid unrounded (lax)
uw	*tool, crew, moo*	back close round
b	*big, able, tab*	voiced bilabial plosive
p	*put, open, tap*	voiceless bilabial plosive
d	*dig, idea, wad*	voiced alveolar plosive
t	*talk, sat*	voiceless alveolar plosive &
t	*meter*	alveolar flap
g	*gut, angle, tag*	voiced velar plosive
k	*cut, ken, take*	voiceless velar plosive
f	*fork, after, if*	voiceless labiodental fricative
v	*vat, over, have*	voiced labiodental fricative
s	*sit, cast, toss*	voiceless alveolar fricative
z	*zap, lazy, haze*	voiced alveolar fricative
th	*thin, nothing, truth*	voiceless dental fricative
dh	*Then, father, scythe*	voiced dental fricative
sh	*she, cushion, wash*	voiceless postalveolar fricative
zh	*genre, azure*	voiced postalveolar fricative
l	*lid*	alveolar lateral approximant
l	*elbow, sail*	velar lateral approximant
r	*red, part, far*	retroflex approximant
y	*yacht, yard*	palatal sonorant glide
w	*with, away*	labiovelar sonorant glide
hh	*help, ahead, hotel*	voiceless glottal fricative
m	*mat, amid, aim*	bilabial nasal
n	*no, end, pan*	alveolar nasal
ng	*sing, anger*	velar nasal
ch	*chin, archer, march*	voiceless alveolar affricate: t + sh
jh	*joy, agile, edge*	voiced alveolar affricate: d + zh

Figure 1: Phonemes

(your mouth, tongue, jaw are in the same position for both) but the former is voiced and the latter is unvoiced. The sounds /d/ and /t/ are another such pair.

One important aspect of phonemes is that their realization can change dependent on the surrounding phones. This is called phonetic context and it caused by a phenomenon called coarticulation. The process of producing these sounds in succession changes their characteristics. Modified versions of a phoneme caused by coarticulation are called allophones.

All state of the art speech recognition systems use this context-dependent nature of phonemes to create a detailed model of phonemes in their various phonetic contexts.

Words and Syntax

Syllables and words

A syllable is a sequence of speech sounds, composed of a nucleus phone and optional initial and final phones. The nucleus is typically a vowel or syllabic consonant, and is the voiced sound that can be shouted or sung.

As an example, the English word "bottle" contains two syllables. The first syllable has three phones, which are "b aa t" in the Arpabet phonetic transcription code. The "aa" is the nucleus, the "b" is a voiced consonant initial phone, and the "t" is an unvoiced consonant final phone. The second syllable consists only of the syllabic cosonant "l".

A word can also be composed of a single syllable which itself is a single phoneme, e.g. "Eye," "uh," or "eau."

In speech recognition, syllable units are rarely considered and words are commonly tokenized into constituent phonemes for modeling.

Syntax and Semantics

Syntax describes how sentences can be put together given words and rules that define allowable grammatical constructs. Semantics generally refers to the way that meaning is attributed to the words or phrases in a sentence. Both syntax and semantics are a major part of natural language processing but neither plays a major role in speech recognition.

Measuring Performance

When we build and experiment with speech recognition systems is it obviously very important to measure performance. Because speech recognition is a sequence classification task (in constrast to image labeling where samples are independent), we must consider the entire sequence when we measure error.

The most common metric for speech recognition accuracy is the Word Error Rate (WER). There are three types of errors a system can make: a substitution, where one word is incorrectly recognized as a different word, a deletion, where no word is hypothesized when the reference transcription has one, and an insertion where the hypothesized transcription inserts extra words not present in the reference. The overall WER can be computed as

$$WER = \frac{N_{\text{sub}} + N_{\text{ins}} + N_{\text{del}}}{N_{\text{ref}}}$$

where N_{sub}, N_{ins}, and N_{del} are the number of substitutions, insertions, and deletions, respectively, and N_{ref} is the number of words in the reference transcription.

The WER is computed using a string edit distance between the reference transcription and the hypothesized transcription. String edit distance can be efficiently computed using dynamic programming. Because string edit distance can be unreliable over a long body of text, we typically accumulate the error counts on a sentence-by-sentence basis and these counts are aggregated overall sentences in the test set to compute the overall WER.

In the example below, the hypothesis "how never a little later he had comfortable chat" is measured against the reference "however a little later we had a comfortable chat" to reveal two substitution errors, one insertion error, and one deletion error.

Reference	Hypothesis	Error
however	how	Substitution
	never	Insertion
a	a	
little	little	
later	later	
we	he	Substitution
had	had	
a		Deletion
comfortable	comfortable	
chat	chat	

The WER for this example is 4/7 = 0.4444 or 44.44%. It can be calculated as follows:

$$WER = \frac{2+1+1}{9} = 0.4444$$

In some cases, the cost of the three different types of errors may not be equivalent. In this case the edit distance computation can be adjusted accordingly.

Sentence error rate (SER) is a less commonly used evaluation metric which treats each sentence as a single sample that is either correct or incorrect. If any word in the sentence is hypothesized incorrectly, the sentence is judged incorrect. SER is computed simply as the proportion of incorrect sentences to total sentences.

Significance testing

Statistical significance testing involves measuring to what degree the difference between two experiments (or algorithms) can be attributed to actual differences in the two algorithms or are merely the result inherent variability in the data, experimental setup or other factors. The idea of statistical significance underlies all pattern classifications tasks. However, the way statistical significance is measure is task-dependent. At the center of most approaches is the notion of a "hypothesis test" in which there is a "null" hypothesis. The question then becomes with what confidence you can argue that the null hypothesis can be rejected.

For speech recognition the most commonly used measure to compare two experiments is called the Matched Pairs Sentence-Segment Word Error (MAPSSWE) Test, commonly shortened to just the Matched Pairs Test. It was suggested for speech recognition evalutions by Gillick et al..

In this approach, the test set is divided up into segments with the assumption that errors in one segment are statistically independent from each other. This assumption is well-matched with typical speech recognition experiments where many test utterances are run through the recognizer one-by-one. Given the utterance-level error count from the WER computation described above, constructing a matched pairs test is straightforward. More details of the algorithm can be found in Pallet et al..

Real-time Factor

Besides accuracy, there may be computational requirements that impact performance, such as processing speed or latency. Decoding speed is usually measured with respect to a real-time factor (RTF). A RTF of 1.0 means that the system processes the data in real-time, takes ten seconds to process ten seconds of audio.

$$RTF = \frac{\text{Total processing time}}{\text{Total audio time}}$$

Factors above 1.0 indicate that the system needs more time to process the data. For some applications, this may be acceptable. For instance, when creating a transcription of a meeting or lecture, it may be more important to take more time and produce accurate transcriptions than to get the transcriptions quickly.

When the RTF is below 1.0, the system processes the data more quickly than it arrives. This can be useful when more than one system runs on the same machine. In that case, multithreading can effectively use one machine to process multiple audio sources in parallel. RTF below 1.0 also indicates that the system can “catch up” to real-time in online streaming applications. For instance, when performing a remote voice query on the phone, network congestion can cause gaps and delays in receiving the audio at the server. If the ASR system can process data in faster than real-time, it can catch up after the data arrives, hiding the latency behind the speed of the recognition system.

In general, any ASR system can be tuned to tradeoff speed for accuracy. But, there is a limit. For a given model and test set, the speed-accuracy graph has an asymptote that is impossible to cross, even with unlimited computing power. The remaining errors can be entirely ascribed to modeling errors. Once the search finds the best result according to the model, further processing will not improve the accuracy.

The Fundamental Equation

Speech recognition is cast as a statistical optimization problem. Specifically, for a given sequence of observations $\mathbf{O} = \{O_1, \ldots, O_N\}$, we seek the most likely word sequence $\mathbf{W} = \{W_1, \ldots, W_M\}$. That is, we are looking for the word sequence which maximizes the posterior probability $P\left(\mathbf{W}|\mathbf{O}\right)$. Mathematically, this can be expressed as:

$$\hat{W} = argmax_W P(W|O)$$

To solve this expression, we employ Bayes rule

$$P\left(W|O\right) = \frac{P\left(O|W\right) P\left(W\right)}{P(O)}$$

Because the word sequence does not depend on the marginal probability of the observation $P(O)$, this term can be ignored. This, we can rewrite this expression as

$$\hat{W} = argmax_W P\left(O|W\right) P(W)$$

This is known as the fundamental equation of speech recognition. The speech recognition problem can be cast as a search over this joint model for the best word sequence.

The equation has a component $P(O|W)$ known as an acoustic model, that describes the distribution over acoustic observations O given the word sequence W. The acoustic model is responsible for modeling how sequences of words are converted into acoustic realizations, and then into the acoustic observations presented to the ASR system. Acoustics and acoustic modeling are covered in Modules 2 and 3 of this course.

The equation has a component $P(W)$ called a language model based solely on the word sequence W. The language model assigns a probability to every possible word sequence. It is trained on sequences of words that are expected to be like those the final system will encounter in everyday use. A language model trained on English text will probably assign a high value to the word sequence "I like turtles" and a low value to "Turtles sing table." The language model steers the search towards word sequences that follow the same patterns as in the training data. Language models can also be seen in purely text-based applications, such as the autocomplete field in modern web browsers. Module 4 of this course is dedicated to language modeling.

For a variety of reasons, building a speech recognition engine is much more complicated that this simple equation implies. In this course, we will describe how these models are constructed and used together in modern speech recognition systems.