CSME 206A Natural Language & Speech Processing **Spring Semester**

Lecture 3: Speech Recognition Models

Lecturer: K.R. Chowdhary *: Professor of CS*

3.1 Introduction

At the simplest level, speech-driven programs are characterized by the words or phrases you can say to a given application and how that application interprets them. An application's active vocabulary-what it listens for – determines what it understands. A speech recognition system requires a *speech engine*, which is language-independent, where the data it recognizes can include several domains of one or more languages. A domain consists of a vocabulary set, pronunciation models, and word usage models associated with a specific speech application. It also has an acoustic component reflected in the voice models, which, the speech engine uses during recognition. These voice models can be either standard (speaker independent) or unique per speaker.

Providing the computer with a natural interface, including the ability to understand human speech, has been a research goal for almost last 40 years. Speech recognition research started with an attempt to decode isolated words from a small vocabulary. As time progressed, the research community began working on large-vocabulary and continuous speech tasks. However, the practical versions of such systems have become moderately usable and commercially successful only in last few years. Even now, these commercial applications either restrict the vocabulary to a few thousand words, in the case of banking or airline reservation systems, or require high-bandwidth, high-feedback situations such as dictation, which requires modifying the user's speech to minimize recognition errors. For example, when some one pronounces "iland", the feedback system may suggest "Are you telling Island?" Based on your input, it will understand current word as well the future words, spoken similarly.

Early attempts at speech recognition tried to apply expert knowledge about speech production and perception processes, but researchers found that such knowledge was inadequate for capturing the complexities of continuous speech. To date, statistical modelling techniques trained from hundreds of hours of speech have provided most speech recognition advancements. Speech researchers have combined these modelling techniques with the massive increase in available computing power over the past several years to explore complex models with hundreds of thousands of parameters.

It is possible to provide free service of speech translation to people on-line, and in turn improve on the speech system model as well as the corpus of speech data-base.

Many organizations' multi-site speech recognition, research cooperation and competition, supported through government agencies such as DARPA, have also fuelled advancements in this field. In addition to participating in government-sponsored competitions, industrial labs, universities, and other companies have fostered rapid advances in speech recognition technology by sharing data and algorithms

A by-product of these cooperative efforts has been that most successful systems share roughly the same architecture and algorithms because each site immediately copies other sites successful algorithms. To enable next-generation applications such as speech recognition over cellular phones, transcription of call center interactions, and recognition of broadcast news, researchers continue to work on the automatic speech

recognition (ASR) system. An understanding of today's ASR systems architecture provides a basis for exploring the recent advances motivated by next-generation applications.

3.2 Resources for speech recognition system

Fig. 3.1 illustrates the resources for typical speech engine used during the recognition process. The domain-specific resources (for example medical domain), such as the vocabulary, can vary dynamically during a given recognition session. A dictation application can transcribe spoken input directly into the document's text content, a transaction application can facilitate a dialog leading to a transaction, and a multimedia indexing application can generate words as index terms [4].

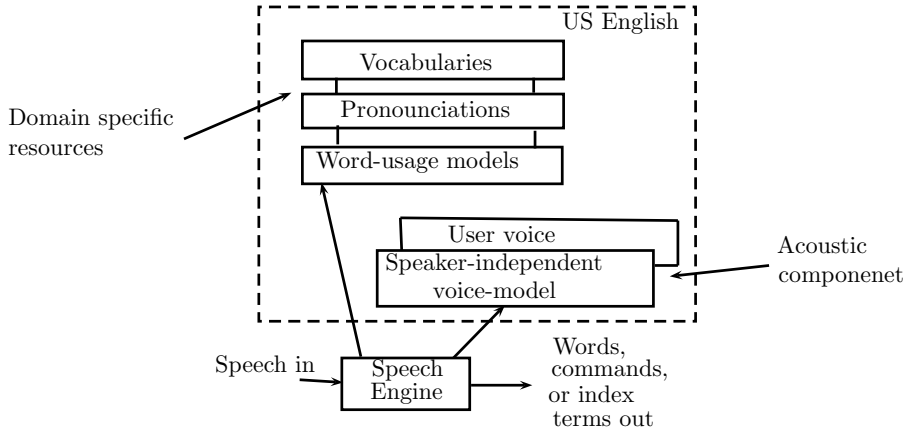

Figure 3.1: Resources for speech recognition system.

In the area of application development, the speech engines typically offer a combination of programmable APIs (Application Programming Interfaces) and tools to create and define vocabularies and pronunciations for the words they contain. A dictation or multimedia indexing application may use a predefined large vocabulary of $100,000$ words or so, while a transactional application may use a smaller, task-specific vocabulary of a few hundred words.

Although adequate for some applications, smaller vocabularies pose usability limitations by requiring strict enumeration of the phrases the system can recognize at any given state in the application. For example, pronounce the PIN (Personal Identification Number) number after having inserted the ATM (Automatic Teller Machine) card, in four separate words one for each digit in PIN. To overcome this limitation, transactional applications define speech grammars for specific tasks. These grammars provide an extension of the single words or simple phrases a vocabulary supports. They form a structured collection of words and phrases bound together by rules that define the set of speech streams the speech engine can recognize at a given time. For example, developers can define a grammar that permits flexible ways of speaking a date, a currency amount, or a number. Prompts that cue users on what they can say next are an important aspect of defining and using grammars. It turns out that speech grammars are a critical component of enabling the Voice over Web.

3.3 Probabilistic Model for Speech Recognition

The process of speech recognition starts with a sampled speech signal. This signal has a good deal of redundancy because the physical constraints on the articulators that produce speech, i.e., the glottis, tongue, lips, and so on, prevent them from moving quickly. Consequently, the ASR system can compress information by extracting a sequence of acoustic feature vectors from the signal (refer previous chapter for more details). Due to this compression, it is possible to perform continuous speech recognition.

Typically, the system extracts a single multidimensional feature vector every 10 milli sec that consists of 39 parameters. These are called feature vectors, which contain information about the local frequency content in the speech signal, as *acoustic observations* because they represent the quantities the ASR system actually observes. The system seeks to infer the spoken word sequence that could have produced the observed acoustic sequence. That means, you need to find the *cause* on observing the acoustic sequence, i.e., *effect*, see figure 3.2. The acoustic sequences have been decoded as *feature vector*.

Figure 3.2: Cause-effect relation.

To simplify the design, it is assumed that the ASR system knows the speaker's vocabulary. This approach restricts the search for possible word sequences with in the words listed in the *lexicon*, which lists the vocabulary and provides *phonemes* - a set of basic units, which are usually individual speech word sounds for the pronunciation of each word.

Commercial lexicons (database of collection of words) typically include tens of thousands of words; however, length of the word sequence uttered by the speaker is unknown. Let us assume that the length of the word sequence uttered is N. If V (vocabulary) represents the size of the lexicon, the ASR system can hypothesize V^N possible word sequences. Language constraints dictate that these word sequences are not equally likely to occur. For example, the word sequence "give me a call" is much more likely than "give call a me." Further, the acoustic feature vectors extracted from the speech signal provide significant clues about the phoneme that produced them.

The sequence of phonemes that corresponds to the acoustic observations implies the word sequence that could have produced the sequence of sounds. Consequently, the acoustic observations provide an important source of information that can help further narrow the space of possible word sequences. The ASR system uses this information to assign a probability that the observed acoustic feature vectors were produced when the speaker uttered a particular word sequence. Essentially, the system efficiently computes these probabilities and outputs the most probable sequence as the decoded hypothesis.

Given a trained speech recognition model and a test speech signal, the goal is to hypothesize the best sentence – a word sequence. If $\mathbf{A}$ represents the acoustic feature sequence extracted from the test data, the speech recognition system should yield *optimal word sequence*, $\hat{\mathbf{W}}$, that matches the $\mathbf{A}$ best, expressed by a probability expression,

$$\hat{\mathbf{W}} = argmax_W P(W|\mathbf{A}) \tag{3.1}$$

Here w is one of all the word sequences, and one of them is optimal. By rearranging the terms in equation 3.2, yields the Bayes's rule,

$$P(W|A) = \frac{P(\mathbf{A}|W)P(W)}{P(\mathbf{A})}. \tag{3.2}$$

Theorem 3.1 ***Bayes Theorem:*** *Given that an even B exists, what is probability that the event A_i has caused it, can be represented by Bayes conditional probability, expressed as,*

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{3.3}$$

where, $P(A_i)$ is called prior probability. □

In fact B might have been caused by any of the A_i (A_1, A_2, ..., A_i, ..., A_n). In that we need to choose the correct A_i. Obviously, the correct one is that which has got maximum probability. Thus,

$$argmax_{A_i}\ P(A_i|B) = argmax_{A_i}\ \frac{P(B|A_i)P(A_i)}{P(B)} \tag{3.4}$$

Since the denominator $p(B)$ is common for all the expressions, it would not effect the order of probability of $p(A_i|B)$, hence the denominator can be dropped, without effecting the result. Thus, we get,

$$argmax_{A_i}\ p(A_i|B) = argmax_{A_i}\ p(B|A_i)p(A_i) \tag{3.5}$$

This new formula is called *Naive Bayes theorem.*

3.3.1 Feature extraction

The input speech signal needs to be processed to extract features relevant for recognition. This stage is common for both *training phase* and *test phase*. The features should help is discriminating similar sounds, and its count should be small to minimize the computation. The speech waveform is broken into segments called *frames*, each of about 10 - 25 msec., and a set of features, equivalent to multi-dimensional feature-vector, are extracted from each frame.

These feature vectors provide clues about the phonemes that produced them. Consequently, the procedure for extracting them is modelled on the workings of the human auditory system.

The sounds are characterized by resonances of the oral tract. Hence, the features extracted from speech signal should represent the gross shape of the spectrum while ignoring the fine features of spectrum such as pitch peaks.

The human auditory system simulates a constant Q-filter bank, in which the sensitivity to the energy in each channel follows a logarithmic relationship.

In a constant Q-filter bank, the ratio of the center frequencies of adjacent filters is constant, and the filter bandwidth is proportional to the center frequencies. Most feature-extraction schemes mimic these steps, with some variations in modelling perceptual aspects, such as male-frequency versus perceptual linear prediction spectral parameters. Because the temporal variation of the speech signal's local spectral content also contains information that helps infer spoken phonemes, the system often computes and appends the temporal derivatives and second derivatives of these features to form the final feature vector.

Table 3.1: Typical lexicon, with the word *the* with two pronunciations.

Phonetic representation	Word
Dhah	The
Thiy	The
Kaet	Cat
Pihg	Pig
Tuw	Two

3.3.2 Generative Probabilistic Model

All of today's most successful speech recognition systems use a "generative probabilistic model" that encapsulates the sequence of steps. The equation 3.6 for this model shows that the recognizer seeks to find the word sequence $\hat{w}_1^N$ that maximizes the word sequence's probability, given some observed acoustic sequence y_1^T (called feature vector of size T). This approach applies the Bayes' law and ignores the denominator term to maximize the product of two terms: 1. the probability of the acoustic observations given the word sequence (first part in equation 3.6) and 2. Probability of the word sequence itself (Second part in equation 3.6).

$$\begin{aligned} \hat{w}_1^N &= argmax_{w_1^N} \; P(w_1^N|y_1^T) \\ &= argmax_{w_1^N} \; P(y_1^T|w_1^N)P(w_1^N) \end{aligned} \tag{3.6}$$

The search for the most likely word sequence w_1^N in Equation (3.6) requires the computation of two terms, $P(y_1^T|w_1^N)$ and $P(w_1^N)$. The first part, $P(y_1^T|w_1^N)$, is called *acoustic model*, i.e., probability of having acoustic observations as y_1^T, if the word spoken was w_1^N. The second $P(w_1^N)$, is called *language model*. Its function is to assign a probability to a sequence of words w_1^N [1].

The simplest way to determine such a probability would be to compute the relative frequencies of different word sequences. However, the number of different sequences grows exponentially with the length of the sequence, making this approach infeasible.

A typical approximation assumes that the probability of the current word depends on the previous two words only, so that the computation can approximate the probability of the word sequence as:

$$P(w_1^N) \approx P(w_1)P(w_2)|w_1) \prod_{i=3}^{i=N} P(w_i|w_{i-1}, w_{i-2}) \tag{3.7}$$

The computation can estimate $P(w_i|w_{i-1}, w_{i-2})$ by counting the relative frequencies of word trigrams, or triplets:

$$P(w_i|w_{i-1}, w_{i-2}) \approx N(w_i, w_{i-1}, w_{i-2})/N(w_{i-1}, w_{i-2}) \tag{3.8}$$

where N refers to the associated event's relative frequency. Typically, training such a language model requires using hundreds of millions of words to estimate $P(w_i w_{i-1}, w_{i-2})$. Even then, many trigrams do not occur in the training text, so the computation must smooth the probability estimates to avoid zeros in the probability assignment.

3.4 Hypothesis search

Three basic components that comprise the hypothesis search are: a lexicon, an acoustic model and a language model.

Lexicon The typical lexicon shown in Table 3.1 lists each word's possible pronunciations, constructed from phonemes, of which English uses approximately 50 pronunciations per word. An individual word can have multiple pronunciations, which, complicates recognition tasks. The system chooses the lexicon on a task-dependent basis, trading off vocabulary size with word coverage. Although a search can easily find phonetic representations for commonly used words in various sources, task-dependent jargon often requires writing out pronunciations by hand.

Acoustic model Hidden Markov Models (HMMs) are the most popular models used continuous speech recognition. The HMMs are capable of modeling and matching sequences that have inherent variability in length as well as acoustic characteristics. An acoustic model computes the probability of feature vector sequences under the assumption that a particular word sequence produced the vectors. Given speech's inherently stochastic nature, speakers usually do not utter a word the same way twice. The variation in a word's or phoneme's pronunciation manifests itself in two ways: duration and spectral content, also known as acoustic observations. Further, phonemes in the surrounding context can cause variations in a particular phoneme's spectral content, a phenomenon called co-articulation.

Language model Given a sequence of test feature vectors, one can compute the likelihood of each phoneme model generating each frame of speech. Subsequently, one can generate a most likely phone sequence or a lattice of phone hypotheses, using Viterbi algorithm. The role of language model is to derive the best sentence hypothesis subject to the constraints of the language.

The language model incorporates the database of various type of linguistic information. The lexicon specifies specifies the sequence of phonemes which form valid words of the language. The syntax describes the rules of combining words to form valid sentences [3].

Apart from the the methods discussed above, the other method used is ANN (Artificial Neural networks). There are benchmarking systems, for performance evaluation of different types of speech recognition systems.

3.5 Basic Architecture of Speech Recognition system

The goal of speech recognition is to generate the optimal word sequence subject to linguistic constraints. A sentence is composed of linguistic units such as words, syllables, and phonemes. The acoustic evidence provided by the acoustic model of such units is combined with, the rules of constructing valid and meaningful sentences in the language to hypothesize the sentence. Therefore, the pattern matching stage can be viewed as taking place in two domains: acoustic and symbolic. In the acoustic domain, a feature vector corresponding to a small segment of test speech (called a frame of speech of about 10 msec) is matched with with the acoustic model of each and every class. The segment is assigned a set of well matching class labels along with their matching scores. This process of label assignment is repeated for every feature vector in the feature vector sequence computed from the test data. The resultant lattice of label hypothesis is processed in conjunction with the language model to yield the recognized sentence [3].

Fig. 3.3 shows the processing stages involved in speech recognition, according to Equation (3.6). In Block 1, the search extracts multidimensional features from the sampled speech signal. In Block 6, the search hypothesizes a probable word sequence through matching. For acoustic model (Block 3), the hypothesis search matches in acoustic domain, while in language model (Block 4), the hypothesis search matching is performed in symbolic domain [2].

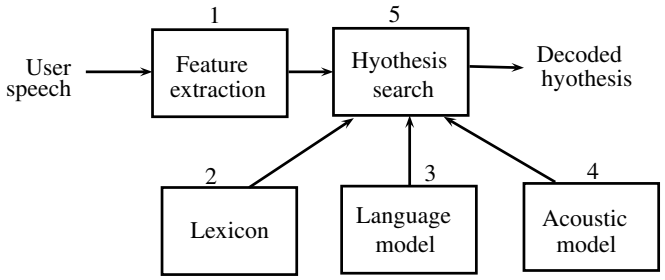

Figure 3.3: Basic Architecture of Speech Recognition system

There are two phases in supervised pattern recognition, viz., *training* and *testing*. The process of extraction of features relevant for classification is common to both phases. During the training phase, the parameters of the classification model are estimated using a large number of training data. During the testing or recognition phase, the features of a test pattern (test speech data) are matched with the trained model of each and every class. The test pattern is declared to belong to that class whose model matches the test pattern best[2].

Several components that drive the recognition steps:

- the lexicon in Block 2 defines the possible words that the search in Block 6 can hypothesize, representing each word as a linear sequence of phonemes;
- the Syntax rules of the language in Block 5 defines the possible words' order (i.e., sentence) that the search can hypothesize, representing each sentence / phrase as a linear sequence of words;
- the acoustic model in Block 3 models the relationship between the feature vectors and the phonemes, $(P(y_1^T|w_1^N))$, which means, given phonemes (word), you can find out the probability of feature vectors, and then maximize it. The probability of feature vector for a given phoneme is available in the acoustic model is database.
- the language model in Block 4 models the linguistic structure but does not contain any knowledge about the relationship between the feature vectors (y_1^T) and the words (w_1^N), that is, the probability $(P(w_1^N|y_1^T))$.

Exercises

1. Is a *Speech Engine* languages dependent or language independent, of the language it recognizes? Justify.
2. What is a speech grammar? Give an example of some speech grammar for, 1. *transaction* application, and for 2. *command* based application. Example of first is drawing money from ATM and ticket book, while of second is Voice-based dialling.
3. What is voice-over-web? Explain the protocols, and language of voice-over-web. What are the applications of voice-over-web?

4. Explain the following two application modes of automatic speech recognition?
   - Command mode.
   - Transaction mode.
5. Give your own version of algorithm to produce indexing of a large data-base of sound collection, e.g., of in music or speech?
6. Explain the significance of following equation, discussed in this chapter.

$$\hat{w}_1^N = argmax_{w_1^N} \; P(y_1^T | w_1^N) P(w_1^N)$$

7. Explain the probabilistic model of ASR, and its following components:
   - Acoustic model.
   - Language model.
8. What are the functions of of speech engine? What are its inputs/outputs, and resources required for it? Describe with suitable diagram.
9. Perform literature survey and find out the "feature" of speech which are extracted and stored in the feature vectors?
10. Explain the Bayes theorem for modelling of conditional probability. How the Naive Bayes models differs from this? Explain the naive Bayes model also.
11. Draw a block diagram and explain the basic architecture of speech recognition in detail.

References

[1] Jurafsky D and Martin J, *Speech and Language Processing, 3rd Ed.*, Pearson India, isbn: 3257227892, Nov. 2005.

[2] Padmanabhan M and Picheny M, *Large-vocabulary speech recognition algorithms*, *Computer*, Vol. 35, No. 4, pp. 42-50, 2002, doi=10.1109/MC.2002.993770

[3] Samudravijaya K, *Automatic Speech Recognition*, `http://www.internationalphoneticalphabet.org/ipa-sounds/ipa-chart-with-sounds/`, urldate = 2018-04-30.

[4] Srinwasan, S and Brown, Eric, *Is Speech Recognition Becoming Mainstream?*, *Computer*, May 2002, pp. 38-41