Automatic Speech Recognition: From the Beginning to the Portuguese Language

André Gustavo Adami

Universidade de Caxias do Sul, Centro de Computação e Tecnologia da Informação
Rua Francisco Getúlio Vargas, 1130, Caxias do Sul, RS 95070-560, Brasil
andre.adami@ucs.br

Abstract. This tutorial presents an overview of automatic speech recognition systems. First, a mathematical formulation and related aspects are described. Then, some background on speech production/perception is presented. An historical review of the efforts in developing automatic recognition systems is presented. The main algorithms of each component of a speech recognizer and current techniques for improving speech recognition performance are explained. The current development of speech recognizers for Portuguese and English languages is discussed. Some campaigns to evaluate and assess speech recognition systems are described. Finally, this tutorial concludes by discussing some research trends in automatic speech recognition.

Keywords: Automatic Speech Recognition, speech processing, pattern recognition

1 Introduction

Speech is a versatile mean of communication. It conveys linguistic (e.g., message and language), speaker (e.g., emotional, regional, and physiological characteristics of the vocal apparatus), and environmental (e.g., where the speech was produced and transmitted) information. Even though such information is encoded in a complex form, humans can relatively decode most of it.

This human ability has inspired researchers to develop systems that would emulate such ability. From phoneticians to engineers, researchers have been working on several fronts to decode most of the information from the speech signal. Some of these fronts include tasks like identifying speakers by the voice, detecting the language being spoken, transcribing speech, translating speech, and understanding speech.

Among all speech tasks, automatic speech recognition (ASR) has been the focus of many researchers for several decades. In this task, the linguistic message is the information of interest. Speech recognition applications range from dictating a text to generating subtitles in real-time for a television broadcast.

Despite the human ability, researchers learned that extracting information from speech is not a straightforward process. The variability in speech due to linguistic, physiologic, and environmental factors challenges researchers to reliably extract

relevant information from the speech signal. In spite of all the challenges, researchers have made significant advances in the technology so that it is possible to develop speech-enabled applications.

This tutorial provides an overview of automatic speech recognition. From the phonetics to pattern recognition methods, we show the methods and strategies used to develop speech recognition systems.

This tutorial is organized as follows. Section 2 provides a mathematical formulation of the speech recognition problem and some aspects about the development such systems. Section 3 provides some background on speech production/perception. Section 4 presents an historical review of the efforts in developing ASR systems. Section 5 through 8 describes each of the components of a speech recognizer. Section 9 describes some campaigns to evaluate speech recognition systems. Section 10 presents the development of speech recognition. Finally, Section 11 discusses the future directions for speech recognition.

2 The Speech Recognition Problem

In this section the speech recognition problem is mathematically defined and some aspects (structure, classification, and performance evaluation) are addressed.

2.1 Mathematical Formulation

The speech recognition problem can be described as a function that defines a mapping from the acoustic evidence to a single or a sequence of words. Let $X = (x_1, x_2, x_3, \ldots, x_t)$ represent the acoustic evidence that is generated in time (indicated by the index t) from a given speech signal and belong to the complete set of acoustic sequences, χ. Let $W = (w_1, w_2, w_3, \ldots, w_n)$ denote a sequence of n words, each belonging to a fixed and known set of possible words, ω. There are two frameworks to describe the speech recognition function: template and statistic.

2.1.1 Template Framework

In the template framework, the recognition is performed by finding the possible sequence of words W that minimizes a distance function between the acoustic evidence X and a sequence of word reference patterns (templates) [1]. So the problem is to find the optimum sequence of template patterns, R^*, that best matches X, as follows

$$R^* = \underset{R^S}{\operatorname{argmin}}\, d(R^S, X)$$

where R^S is a concatenated sequence of template patterns from some admissible sequence of words. Note that the complexity of this approach grows exponentially with the length of the sequence of words W. In addition, the sequence of template patterns does not take into account the silence or the coarticulation between words. Restricting the number of words in a sequence [1], performing incremental processing

[2], or adding a grammar (language model) [3] were some of the approaches used to reduce the complexity of the recognizer.

This framework was widely used in speech recognition until the 1980s. The most known methods were the dynamic time warping (DTW) [3-6] and vector quantization (VQ) [4, 5]. The DTW method derives the overall distortion between the acoustic evidences (speech templates) from a word reference (reference template) and a speech utterance (test template). Rather than just computing a distance between the speech templates, the method searches the space of mappings from the test template to that of the reference template by maximizing the local match between the templates, so that the overall distance is minimized. The search space is constrained to maintain the temporal order of the speech templates. Fig. 1 illustrates the DTW alignment of two templates.

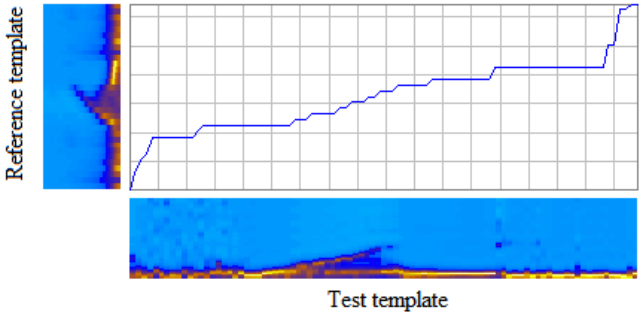

Fig. 1. Example of dynamic time warping of two renditions of the word "one".

The VQ method encodes the speech patterns from the set of possible words into a smaller set of vectors to perform pattern matching. The training data from each word $w_i \in \omega$ is partitioned into M clusters so that it minimizes some distortion measure [1]. The cluster centroids (codewords) are used to represent the word w_i, and the set of them is referred to as codebook. During recognition, the acoustic evidence of the test utterance is matched against every codebook using the same distortion measure. The test utterance is recognized as the word whose codebook match resulted in the smallest average distortion. Fig. 2 illustrates an example of VQ-based isolated word recognizer, where the index of the codebook with smallest average distortion defines the recognized word. Given the variability in the speech signal due to environmental, speaker, and channel effects, the size of the codebooks can become nontrivial for storage. Another problem is to select the distortion measure and the number of codewords that is sufficient to discriminate different speech patterns.

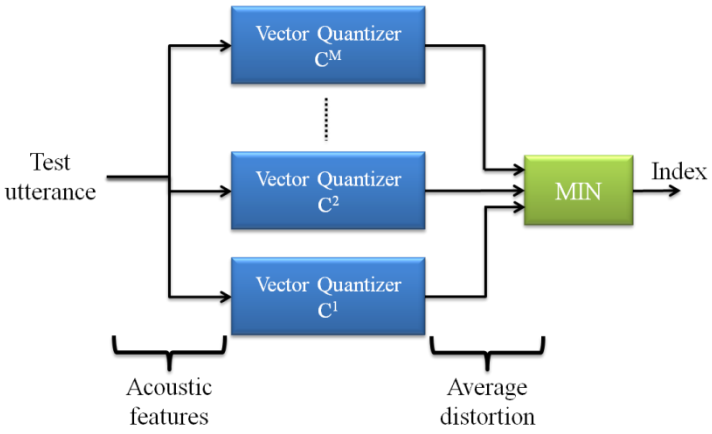

Fig. 2. Example of VQ-based isolated word recognizer.

2.1.2 Statistical Framework

In the statistical framework, the recognizer selects the sequence of words that is more likely to be produced given the observed acoustic evidence. Let $P(W|X)$ denote the probability that the words W were spoken given that the acoustic evidence X was observed. The recognizer should select the sequence of words $\widetilde{W}$ satisfying

$$\widetilde{W} = \underset{W \in \omega}{\operatorname{argmax}}\, P(W|X).$$

However, since $P(W|X)$ is difficult to model directly, Bayes' rule allows us to rewrite such probability as

$$P(W|X) = \frac{P(W)P(X|W)}{P(X)}$$

where $\mathrm{P}(W)$ is the probability that the sequence of words W will be uttered, $\mathrm{P}(X|W)$ is the probability of observing the acoustic evidence X when the speaker utters W, and $P(X)$ is the probability that the acoustic evidence X will be observed. The term $P(X)$ can be dropped because it is a constant under the max operation. Then, the recognizer should select the sequence of words $\widetilde{W}$ that maximizes the product $P(W)P(X|W)$, i.e.,

$$\widetilde{W} = \underset{W \in \omega}{\operatorname{argmax}}\, P(W)P(X|W). \quad (1)$$

This framework has dominated the development of speech recognition systems since the 1980s.

2.2 Speech Recognition Architecture

Most successful speech recognition systems are based on the statistical framework described in the previous section. Equation (1) establishes the components of a speech recognizer. The prior probability $\mathrm{P}(W)$ is determined by a language model, and the

likelihood $P(X|W)$ is determined by a set of acoustic models, and the process of searching over all possible sequence of words W that maximizes the product is performed by the decoder. Fig. 3 shows the main components of an ASR system.

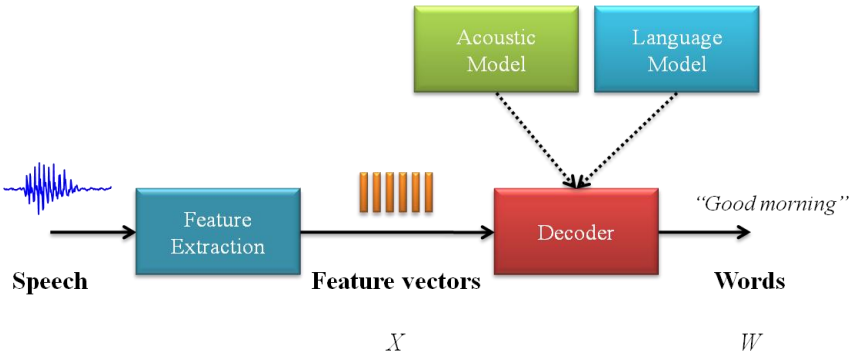

Fig. 3. Architecture of an ASR system.

The statistical framework for speech recognition brings four problems that must be addressed:

1. The acoustic processing problem, i.e., to decide what acoustic data X is going to be estimated. The goal is to find a representation that reduces the model complexity (low dimensionality) while keeping the linguistic information (discriminability), despite the effects from the speaker, channel or environmental characteristics (robustness). In general, the speech waveform is transformed into a sequence of acoustic feature vectors, and this process is commonly referred to as feature extraction. Some of the most used methods for signal processing and feature extraction are described in Section 5.
2. The acoustic modeling problem, i.e., to decide on how $P(X|W)$ should be computed. Thus several acoustic models are necessary to characterize how speakers pronounce the words of W given the acoustic evidence X. The acoustic models are highly dependent of the type of application (e.g., fluent speech, dictation, commands). In general, several constraints are made so that the acoustic models are computationally feasible. The acoustic models are usually estimated using Hidden Markov Models (HMMs) [1], described in Section 6.
3. The language modeling problem, i.e., to decide on how to compute the a priori probability $P(W)$ for a sequence of words. The most popular model is based on a Markovian assumption that a word in sentence is conditioned on only the previous N-1 words. Such statistical modeling method is called N-gram and it is described in Section 7.
4. The search problem, i.e., to find the best word transcription $\widetilde{W}$ for the acoustic evidence X, given the acoustic and language models. Since it is impractical to exhaustively search all possible sequence of words, some methods have been developed to reduce the computational requirements. Section 8 describes some of the methods used to perform such search.

2.3 Automatic Speech Recognition Classification

ASR systems can be classified according to some parameters that are related to the task. Some of the parameters are:

- **Vocabulary size**: speech recognition is easier when the vocabulary to recognize is smaller. For example, the task of recognizing digits (10 words) is relatively easier when compared to tasks like transcribing broadcast news or telephone conversations that involve vocabularies of thousands of words. There are no established definitions, but small vocabulary is measure in tens of words, medium in hundreds of words, large in thousands of words and up [6]. However, the vocabulary size is not a reliable measure of task complexity [7]. The grammar constraints of the task can also affect the complexity of the system. That is, tasks with no grammar constraints are usually more complex because all words can follow any word.
- **Speaking style**: this defines whether the task is to recognize isolated words or continuous speech. In isolated word (e.g., digit recognition) or connected word (e.g., sequence of digits that form a credit card number) recognition, the words are surrounded by pauses (silence). This type of recognition is easier than continuous speech recognition because, in the latter, the word boundaries are not so evident. In addition, the level of difficulty varies among the continuous speech recognition due to the type of interaction. That is, recognizing speech from human-human interactions (recognition of conversational telephone speech, broadcast news) is more difficult than human-machine interactions (dictation software) [8]. In read speech or when humans interact with machines, the produced speech is simplified (slow speaking rate and well articulated) so that it is easy to understand it [7].
- **Speaker mode**: the recognition system can be used by a specific speaker (speaker dependent) or by any speaker (speaker independent). Despite the fact that speaker dependent systems require to be trained on the user, they generally achieve better recognition results (there is no much variability caused by the different speakers). Given that speaker independent systems are more appealing than speaker dependent ones (no training required for the user), some speaker-independent ASR systems are performing some type of adaptation to the individual user's voice to improve their recognition performance.
- **Channel type**: the characteristics of the channel can affect the speech signal. It may range from telephone channels (with a bandwidth about 3.4 kHz) to wireless channels with fading and with a sophisticated voice [6].
- **Transducer type**: defines the type of device used to record the speech. The recording may range from high-quality microphones to telephones (landline) to cell phones to array microphones (used in applications that track the speaker location).

Fig. 4 shows the progress of spoken language systems along the dimensions of speaking style and vocabulary size. Note that the complexity of the system grows from the bottom left corner up to the top right corner. The bars separate the applications that can and cannot be supported by speech technology for viable deployment in the corresponding time frame.

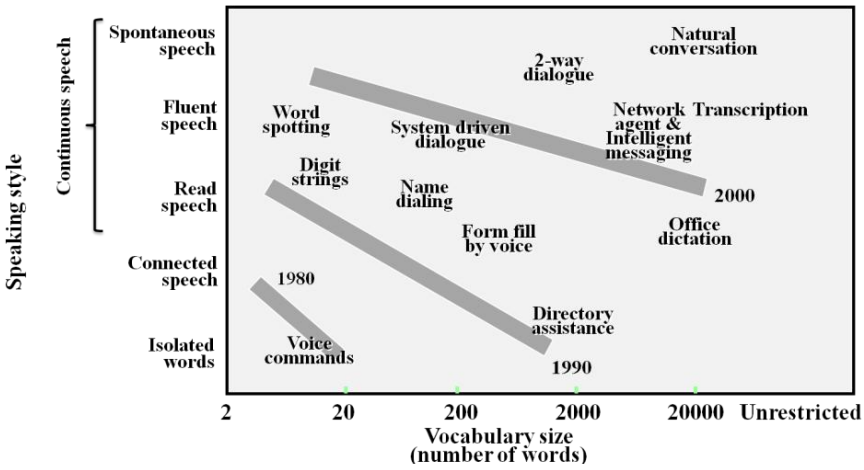

Fig. 4. Progress of spoken language system along the dimension of speaking style and vocabulary size (adapted from [9]).

Some other parameters specific to the methods employed in the development of an ASR system are going to be analyzed throughout the text.

2.4 Evaluating the Performance of ASR

A commonly metric used to evaluate the performance of ASR systems is the word error rate (WER). For simple recognition systems (e.g., isolated words), the performance is simply the percentage of misrecognized words. However, in continuous speech recognition systems, such measure is not efficient because the sequence of recognized words can contain three types of errors. Similar to the error in the digit recognition, the first error, known as word substitution, happens when an incorrect word is recognized in place of the correctly spoken word. The second error, known as word deletion, happens when a spoken word is not recognized (i.e., the recognized sentence does not have the spoken word). Finally, the third error, known as word insertion, happens when extra words are estimated by the recognizer (i.e., the recognized sentence contains more words than what actually was spoken). In the following example, the substitutions are bold, insertions are underlined, and deletions are denoted as *.

Correct sentence: "Can you bring me a glass of water, please?"
Recognized sentence: "Can you bring * a glass of <u>cold</u> water, police?"

To estimate the word error rate (WER), the correct and the recognized sentence must be first aligned. Then the number of substitutions (S), deletions (D), and insertions (I) can be estimated. The WER is defined as

$$WER = 100\% \times \left(\frac{S + D + I}{|W|} \right)$$

where $|W|$ is the number of words in the sequence of word W. Table 1 shows the WER for a range of ASR systems. Note that for a connected digit recognition task, the WER goes from 0.3% in a very clean environment (TIDIGIT database) [10] to 5% (AT&T HMIHY) in a conversation context from a speech understanding system [11]. The WER increases together with the vocabulary size, when the performance of ATIS [12] is compared to Switchboard [13] and Call-home [14]. In contrast, the performance of NAB & WSJ [15] is lower than the Switchboard and Call-home. The difference is that in the NAB & WSJ task the speech is carefully uttered (read speech) as opposed to the spontaneous speech in the telephone conversations.

Table 1. Word error rates for a range of speech recognition systems (adapted from [16]).

Task	Type of speech	Vocabulary size	WER
Connected digit string (TIDIGIT database)	Spontaneous	11 (0-9, oh)	0.3%
Connected digit string (AT&T mall recordings)	Spontaneous	11 (0-9, oh)	2.0%
Connected digit string (AT&T HMIHY)	Conversational	11 (0-9, oh)	5.0%
Resource Management (RM)	Read speech	1,000	2.0%
Airline travel information system (ATIS)	Spontaneous	2,500	2.5%
North American business (NAB & WSJ)	Read Text	64,000	6.6%
Broadcast News	Narrated news	210,000	~15.0%
Switchboard	Telephone conversation	45,000	~27.0%
Callhome	Telephone conversation	28,000	~35.0%

3 Speech

In this section, we review human speech production and perception. A better understanding of both processes can result in better algorithms for processing speech.

3.1 Speech Production

The anatomy of the human speech production system is shown in Fig. 5. The vocal apparatus comprises three cavities: nasal, oral, and pharyngeal. The pharyngeal and oral cavities are usually grouped into one unit referred to as the vocal tract, and the nasal cavity is often called the nasal tract [1]. The vocal tract extends from the opening of the vocal folds, or glottis, through the pharynx and mouth to the lips (shaded area in Fig. 5). The nasal tract extends from the velum (a trapdoor-like mechanism at the back of the oral cavity) to the nostrils.

The speech process starts when air is expelled from the lungs by muscular force providing the source of energy (excitation signal). Then the airflow is modulated in various ways to produce different speech sounds. The modulation is mainly performed in the vocal tract (the main resonant structure), through movements of several articulators, such as the velum, teeth, lips, and tongue. The movements of the articulators modify the shape of the vocal tract, which creates different resonant frequencies and, consequently, different speech sounds. The resonant frequencies of the vocal tract are known as formants, and conventionally they are numbered from the low- to the high-frequency: F_1 (first formant), F_2 (second formant), F_3 (third formant),

and so on. The resonant frequencies can also be influenced when the nasal tract is coupled to the vocal tract by lowering the velum. The coupling of both vocal and nasal tracts produces the "nasal" sounds of speech, like /*n*/ sound of the word "nine".

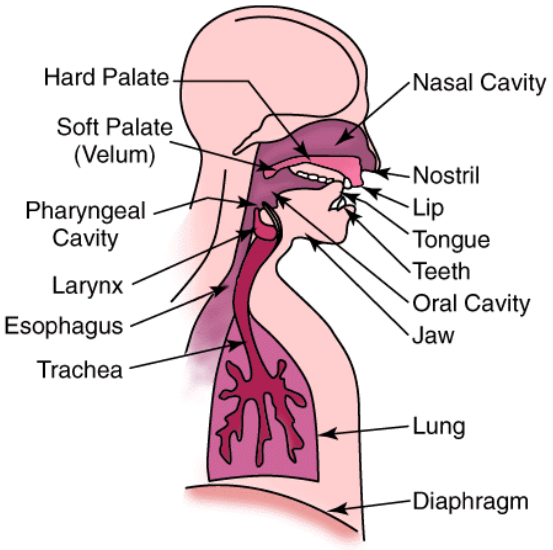

Fig. 5. The human speech production system [17].

The airflow from the lungs can produce three different types of sound source to excite the acoustic resonant system [18]:

- For voiced sounds, such as vowels, air is forced from the lungs through trachea and into the larynx, where it must pass between two small muscular folds, the vocal folds. The tension of the vocal folds is adjusted so that they vibrate in oscillatory fashion. This vibration periodically interrupts the airflow creating a stream of quasi-periodic pulses of air that excites the vocal tract. The modulation of the airflow by the vibrating vocal folds is known as phonation. The frequency of vocal fold oscillation, also referred to as fundamental frequency (F_0), is determined by the mass and tension of the vocal folds, but is also affected by the air pressure from the lungs.
- For unvoiced sounds, the air from the lungs is forced through some constriction in the vocal tract, thereby producing turbulence. This turbulence creates a noise-like source to excite the vocal tract. An example is the /s/ sound in the word "six".
- For plosive sounds, pressure is built up behind a complete closure at some point in the vocal tract (usually toward the front of the vocal tract). The subsequent abrupt release of this pressure produces a brief excitation of the vocal tract. An example is the /t/ sound in the word "put".

Note that these sound sources can be mixed together to create another particular speech sound. For example, the voiced and turbulent excitation occurs simultaneously for sounds like /*v*/ (from the word "victory") and /*z*/ (from the word "zebra").

Despite the inherent variance in producing speech sounds, linguists categorize speech sounds (or phones) in a language into units that are linguistically distinct, known as phonemes. There are about 45 phonemes in English, 50 for German and Italian, 35 for French and Mandarin, 38 for Brazilian Portuguese (BP), and 25 for Spanish [19]. The different realizations in different contexts of such phonemes are called allophones. For example, in English, the aspirated t [t^h] (as in the word 'tap') and unaspirated [t] (as in the word 'star') correspond to the same phoneme /t/, but they are pronounced slightly different. In Portuguese, the phoneme /t/ is pronounced differently in words that end with '*te'* due to regional differences: *leite* ('*milk'*) is pronounced as either /lejt͡ʃi/ (southeast of Brazil) or /lejte/ (south of Brazil). The set of phonemes can be classified into vowels, semi-vowels and consonants.

The sounds of a language (phonemes and phonetic variations) are represented by symbols from an alphabet. The most known and long-standing alphabet is the International Phonetic Alphabet or IPA[1]. However, other alphabets were developed to represent phonemes and allophonic variations among phonemes not presents in the IPA: Speech Assessment Methods Phonetic Alphabet (SAMPA)[20] and Worldbet.

Vowels

The BP language has eleven oral vowels: /a ɐ e ẹ ɛ i ɪ ɔ o u ʊ/. Some examples of oral vowels are presented in Table 2.

Table 2. Oral vowels examples (adapted from [21]).

Oral Vowel	Phonetic Transcription	Portuguese Word	English Translation
i	sikʊ	sico	chigoe
e	sekʊ	seco	dry
ɛ	sɛkʊ	seco	(I) dry
a	sakʊ	saco	bag
ɔ	sɔkʊ	soco	(I) hit
o	sokʊ	soco	hit (noun)
u	sukʊ	suco	juice
ɪ	sakɪ	saque	withdrawal
ẹ	nũmẹɾʊ	número	number
ɐ	sakɐ	saca	sack
ʊ	sakʊ	saco	bag

It also has five nasalized vowels: /ɐ̃ ẽ ĩ õ ũ/. Such vowels are also produced when they precede nasal consonants (e.g., /ɲ/ and /m/). Some examples of oral vowels are presented in Table 3.

[1] http://www.langsci.ucl.ac.uk/ipa/

Table 3. Nasal vowels examples (adapted from [21]).

Nasal Vowel	Phonetic Transcription	Portuguese Word	English Translation
ĩ	sĩ^{n}tʊ	cinto	'belt'
ĩ	sĩmɐ	cima	'above'
ẽ	se^{n}tʊ	sento	'(I) sit'
ẽ	te^{m}porɑ̯	temporal'	'storm'
ɐ̃	sɐ̃^{n}tʊ	santo	'saint'
ɐ̃	gɐ̃ɲar	ganhar	'(to) win'
ɐ̃	imɐ̃	imã	magnet
õ	sõ^{n}dʊ	sondo	'(I) probe'
ũ	sũ^{n}tʊ	sunto	'summed up'

The position of the tongue's surface and the lip shape are used to describe vowels in terms of the common features *height* (vertical dimension, i.e., high, mid, low), *backness* (horizontal dimension, i.e., front, mid, and back) and *roundedness* (lip position, i.e., round and tense). Fig. 6 illustrates the height and backness features of vowels. According to the backness features, /e ẹ ẽ ɛ i ɪ ĩ/ are front vowels, /ɐ̃ a ɐ/ are mid vowels, and /ɔ o õ ũ u/ are back vowels.

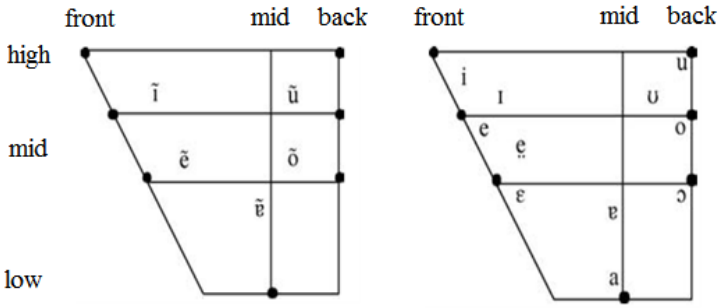

Fig. 6. Relative tongue positions in the nasal (left) and oral (right) vowels for BP, as they are pronounced in São Paulo [21].

The variations in the tongue placement with the vocal tract shape and length determine the resonances frequencies of each vowel sound. Fig. 7 shows the average frequencies of the first three formants for some BP vowels. Vowels are usually long in duration and are spectrally well defined [1], what make the task of vowel recognition easier for humans and machines.

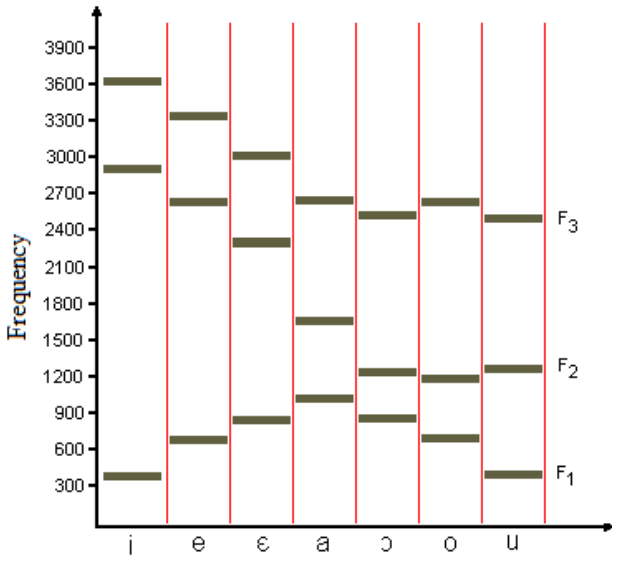

Fig. 7. F_1, F_2 and F_3 of BP oral vowels estimated over 90 speakers [22].

Semivowels

Semivowels are a class of speech sounds that have a vowel-like characteristic. Sometimes they are also classified as approximant because the tongue approaches the top of the oral cavity without obstructing the air flow [23]. They occur at the beginning or end of a syllable and they can be characterized by a gliding speech sound between adjacent vowel-like phonemes within a single syllable [1]. Such gliding speech sound is also known as diphthong (for two phonemes) or triphthong (for three phonemes). Usually, the sounds produced by semivowels are weak (because of the gliding of the vocal tract) and influenced by the neighboring phonemes.

In the BP language, semivowels occur with oral vowels (represented by the phonemes /w/ and /j/) or nasal vowels (represented by the phonemes /w̃/ and /j̃/), as illustrated in Table 4. Semivowels also occur in words that end in nasal diphthongs (i.e., word with endings: -am, -em/-ém, -ens/-éns, -êm, -õem).

Table 4. Examples of semivowels in the BP language.

Semivowel	Phonetic Transcription	Portuguese Word	English Translation
j	lejt͡ʃi	leite	‘milk’
w	sɛw	céu	‘sky’
j̃	sẽj̃	cem	‘(a) hundred’
j̃	mɐ̃j̃	mãe	‘mother’
w̃	sagwɐ̃w̃	saguão	‘lobby’
w̃	mɐ̃w̃	mão	‘hand’

Consonants

Consonants are characterized by momentary interruption or obstruction of the airstream through the vocal tract. Therefore, consonants can be classified according to the place and manner of this obstruction. The obstruction can be caused by the lips, the tongue tip and blade, and the back of the tongue. Some of the terms used to specify the place of articulation, as illustrated in Fig. 8, are the following:

- Bilabial: made by constricting both lips as in the phoneme /p/ as in *pata* /patɐ/ (‘paw’). The BP consonants that belong to this class are /p/, /b/, and /m/.
- Labiodentals: the lower lip contacts the upper front teeth as in the phoneme /f/ as in *faca* /fakɐ/ (‘knife’). The BP consonants that belong to this class are /f/ and /v/.
- Dental: the tongue tip or the tongue blade protrudes between the upper and lower front teeth (most speakers of American English, also known as interdental [24]) or have it close behind the lower front teeth (most speakers of BP). The BP consonants that belong to this class are /t/, /d/, and /n/. The allophones /t͡ʃ/ and /d͡ʒ/ occur in syllables that start with ‘ti’ (as in the proper name Tita /t͡ʃitɐ/) or ‘di’ (as in the word dita /d͡ʒitɐ/, ‘said (fem.)’), respectively, and in words that end with ‘te’ and ‘de’.
- Alveolar: the tongue tip or blade approaches or touches the alveolar ridge as in the phoneme /s/ as in *saca* /sakɐ/ (‘sack’);
- Retroflex: the tongue tip is curled up and back. However, such phoneme does not occur in BP.
- Postalveolar: the tongue tip or (usually) the tongue blade approaches or touches the back of the alveolar ridge as in the phoneme /ʃ/ as in *chaga* /ʃagɐ/ (‘open sore’). Sometimes it is called palato-alveolar since it is the area between the alveolar ridge and the hard palate.
- Palatal: the tongue blade constricts with the hard palate (“roof” of the mouth) as in the phoneme /ɲ/ as in *ganhar* /gɐ̃ɲar/ (‘(to) win’).
- Velar: the dorsum of the tongue approaches the soft as in the phoneme /g/ as in *gata* /gatɐ/ (‘(female) cat’).

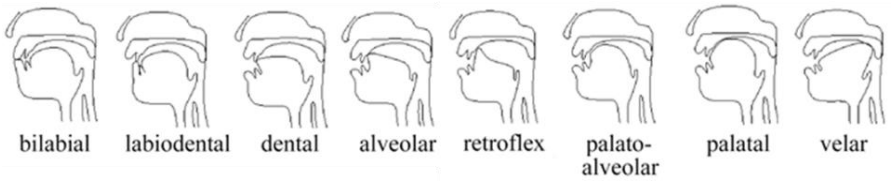

Fig. 8. Places of articulation.

The manner of articulation describes the type of closure made by the articulators and the degree of the obstruction of the airstream by those articulators, for any place of articulation. The major distinctions in manner of articulation are:

- Plosive (or **oral stop**): a complete obstruction of the oral cavity (no air flow) followed by a release of air. Examples of BP phonemes include /p t k/ (unvoiced)

and /b d g/ (voiced). In the voiced consonants, the voicing is the only sound made during the obstruction.

- Fricative: the airstream is partially obstructed by the close approximation of two articulators at the place of articulation creating a narrow stream of turbulent air. Examples of BP phonemes include /f s ʃ/ (unvoiced) and / v z ʒ ɣ/ (voiced).
- Affricate: begins with a complete obstruction of the oral cavity (similar to a plosive) but it ends as a fricative. Examples of BP allophones include /t͡ʃ/ (unvoiced) and /d͡ʒ/ (voiced).
- Nasal (or **nasal stop**): it also begins with a complete obstruction of the oral cavity, but with the velum open so that air passes freely through the nasal cavity. The shape and position of the tongue determine the resonant cavity that gives different nasal stops their characteristic sounds. Examples of BP phonemes include /m b ɲ/, all voiced.
- Tap: a single tap is made by one articulator against another resulting in an instantaneous closure and reopening of the vocal tract. Example of BP phoneme is /ɾ/ in the word *caro* /kaɾʊ/ ('expensive').
- Approximant: one articulator is close to another without causing a complete obstruction or narrowing of the vocal tract. The consonants that produce an incomplete closure between one or both sides of the tongue and the roof of the mount are classified as lateral approximant. Examples of lateral approximant in BP include /l/ of /galʊ/ *galo* ('rooster') and /ʎ/ of /gaʎʊ/ *galho* ('branch'). Semivowels, sometimes called a **glide**, are also a type of approximant because it is pronounced with the tongue closer to the roof of the mouth without causing a complete obstruction of the airstream.

The BP consonants can be arranged by manner of articulation (rows), place of articulation (columns), and voiceless/voiced (pairs in cells) as illustrated in Table 5.

Table 5. The consonants of BP arranged by place (columns) and manner (rows) of articulation [21].

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Palatal	Velar
Plosive	p b		t d				k g
Affricates			(t͡ʃ) (d͡ʒ)				
Nasal	m		n			ɲ	
Tap				ɾ			
Fricative		f v		s z	ʃ ʒ		ɣ
Lateral approximant				l		ʎ	

The place and manner of articulation are often used in automatic speech recognition as a useful way of grouping phones together or as features [25, 26].

Despite all these different descriptions on how these sounds are produced, we have to understand that speech production is characterized by a continuous sequence of articulatory movements. Since every phoneme has an articulatory configuration, physiological constraints limit the articulatory movements between adjacent phonemes. Thus, the realization of phonemes is affected by the phonetic context. This

phenomenon between adjacent phonemes is called coarticulation [24]. For example, a noticeable change in the place of articulation can be observed in the realization of /k/ before a front vowel as in 'key' /ki/ as compared with a back vowel as in 'caw' /kɔ/.

3.2 Speech Perception

The process of how the brain interprets the complex acoustical patterns of speech as linguistics units is not well understood [18, 27]. Given the variations in the speech signal produced by different speakers in different environments, it has become clear that speech perception does not rely on invariant acoustic patterns available in the waveform to decode the message. It is possible to argue that the linguistic context is also very important for the perception of speech, given that we are able to identify nonsense syllables spoken (clearly articulated) in isolation [27].

It is out of the scope of this tutorial to give more than a brief overview of the speech perception. We are going to focus on the physical aspects of the speech perception used for speech recognition.

3.2.1 The auditory System

The auditory system can be divided anatomically and functionally into three regions: the outer ear, the middle ear, and the inner ear. Fig. 9 shows the structure of the human ear. The outer ear is composed of the pinna (external ear, the part we can see) and the external canal (or meatus). The function of pinna is to modify the incoming sound (in particular, at high frequencies) and direct it to the external canal. The filtering effect of the human pinna preferentially selects sounds in the frequency range of human speech. It also adds directional information to the sound.

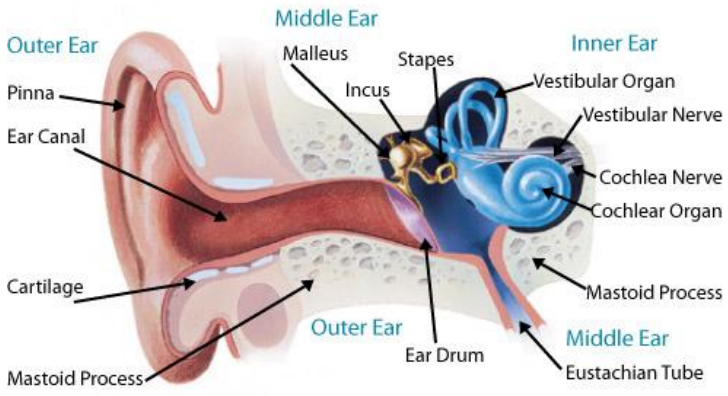

Fig. 9. Structure of the human ear[2].

The sound waves conducted by the pinna go through the external canal until they hit the eardrum (or tympanic membrane), causing it to vibrate. These vibrations are

[2] http://www.hearingclinic.net.au/mhc/content/the_ear.php

transmitted through middle ear by three small bones, the ossicles, to a membrane-covered opening (called oval window) in the bony wall of the spiral shaped structured of the inner ear – the cochlea. The middle ear is an air-filled cavity (tympanic cavity) that couples sound from the air to the fluids via oval window in the cochlea. It connects to the throat/nasopharynx via the Eustachian tube. The smallest bones in the human body, the ossicles are named for their shape. The hammer (malleus) joins the inside of the eardrum. The anvil (incus), the middle bone, connects to the hammer and to the stirrup (stapes). The base of the stirrup, the footplate, fills the oval window which leads to the inner ear. Because of the resistance of the oval window, the middle ear converts, through the lever action of the ossicles, low-pressure vibration of the eardrum into high-pressure vibration at the oval window. It is interesting to note that the middle ear is most efficient at middle frequencies (500-4000Hz), which mostly characterizes speech sounds.

The inner ear consists of a bony labyrinth filled with fluid that has two main functional parts: the vestibular system (the rear part, responsible for the balance) and the cochlea (frontal part, responsible for hearing). The cochlea is divided along its length by two membranes: Reissner's membrane and the basilar membrane (BM), as shown in Fig. 10. The BM has its base situated at the start of the cochlea, the oval window. At the end of the BM, known as the apex, there is a small opening (the helicotrema), which connects the two outer chambers of the cochlea, the scala vestibuli and the scala tympani. The oscillation of the oval window due to an

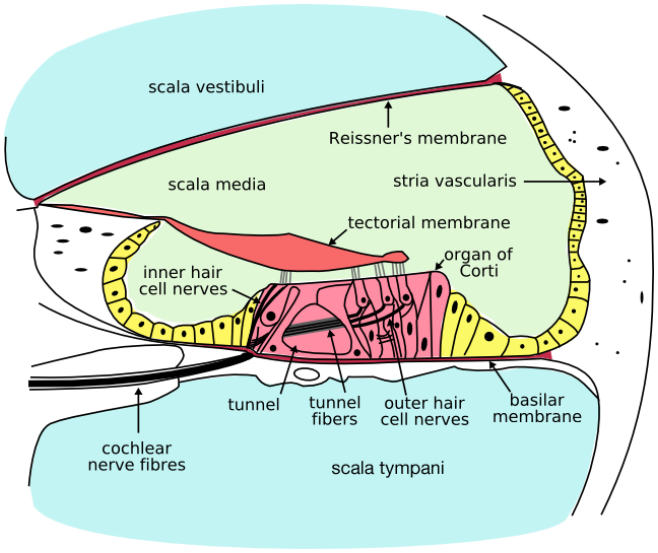

Fig. 10. Cross section of the cochlea[3].

frequency and varies along the BM because the BM is stiff and narrow at the base and it is wider and much less stiff at the apex. This means that different frequencies

[3] http://en.wikipedia.org/wiki/Cochlea

resonate particularly strongly at different points along the BM. High-frequency sounds cause the greatest motion of the BM near the oval window and low-frequency sounds cause the greatest motion of the BM farthest from the oval window. This suggests that the BM can be modeled as a bank of overlapping bandpass filters [28] (also known as 'auditory filters'). Consequently, each location on the BM responds to a limited range of frequencies, so each different point correspond to an auditory filter with a different center frequency (the frequency that gives maximum response). These auditory filters are nonlinear, level-dependent and the bandwidth increases from the apex to base of the cochlea (from low to high frequency). The bandwidth of the auditory filter is called the critical bandwidth [28].

Finally, the motion of the BM is converted into neural signal in the auditory nervous system for final processing resulting in sound perception through hair cell nerves. The hair cell nerves are between the BM and the tectorial membrane, which form part of a structure called organ of Corti (Fig. 10). The tunnel of Corti divides the hair cell nerves into two groups. Closest to the outside of the cochlea, the outer hair cells are arranged in three rows in the cat and up to five rows in humans and make contact with tectorial membrane [27]. On the other side of the arch, the inner hair cells form a single row. There are about 25 000 outer hair cells (each with about 140 hairs, or stereocilia, protruding from it) and 3 500 inner hair cells (each with about 40 hairs). The up and down motion of the BM causes the fine stereocilia to shear back and forth under the tectorial membrane. The displacement of the stereocilia leads to excitation of the inner hair cells generating action potentials in the neurons of the auditory nerve. The great majority of neurons that carry information to the auditory system connect to inner hair cells (each hair cell is contacted by about 20 neurons) [27]. The main role of the outer hair cells may be to produce high sensitivity and sharp tuning.

4 Historical Review of Automatic Speech Recognition

The first machine to recognize speech, in some level, was a commercial toy named Radio Rex produced in 1922 by Elmwood Button Company [7]. Rex was a brown bulldog made of celluloid and metal that jumps out the house when its name was spoken. The dog was held within its house by an electromagnet arrangement against the force of a spring. The electromagnet arrangement could be interrupted by a vibration caused by an acoustic energy of 500 Hz that released the dog. Such energy is present in the vowel of the word Rex. Despite its ingenious way of responding when the dog's name was called, the toy suffered the same problem of many current ASR systems: it rejected out-of-vocabulary words. This problem happens because several words can carry sounds that have 500Hz acoustic energy and the toy could not distinguish them.

The first true speech recognizer was a system built in 1952 by David et. al [29] at Bell Laboratories. The system was able to recognize digits from a single speaker. The system used the spectral energy over time of two wide bands that cover the first two formant frequencies of the vocal tract. Such approach was quite successful (achieved a 2% error) for a single speaker because it averaged out the speech variability by

performing a histogram of the energy (therefore, the time information was lost). Besides, the digits were separated by pauses.

In 1959, Denes and Fry [30] introduced a simple bigram language model for phonemes to improve the recognition of speech sounds (4 vowels and 9 consonants), consequently, words. The hypothesis was that the probability of uttering a linguistic unit is conditional to the probability of the previous unit. Their system used derivatives of spectral energies as the acoustic information.

Given that computers were not fast enough in the 1960s for signal processing, several Japanese researchers built special-purpose devices to perform speech recognition tasks. Suzuki and Nakata at the Radio Research Lab in Tokyo [31] built a hardware for vowel recognizer (Fig. 11). This system was based on a filter bank spectrum analyzer whose output from each of the channels was fed to a vowel decision circuit, and a majority decision logic scheme was used to choose the spoken vowel. Nagata et al. [32] at NEC Laboratories built a hardware for digit recognition (results of 99.7% for 1000 utterances of 20 male speakers were obtained for a set of formant-related features). Sakai and Doshita at Kyoto University [33] developed a hardware for phoneme recognition (one hundred Japanese monosyllables). This last hardware is considered significant because it was the first report of a system that performed speech segmentation along with zero-crossing analysis on different sections of the speech to recognize phonemes. Up to that date, recognizers were built assuming that the unknown utterance contained only the token to be recognized and no other speech sound.

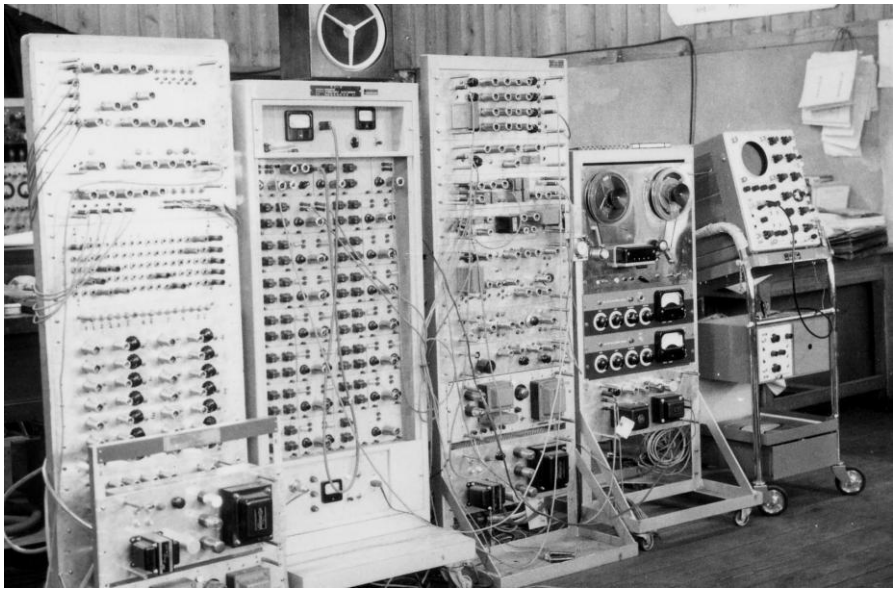

Fig. 11. Front view of the spoken vowel recognizer built by Suzuki and Nakata at the Radio Research Lab in Tokyo [31].

Besides segmenting speech, another approach that was used to deal with the nonuniformity of the time scales in speech events was time normalization. Same

speech event can have different durations for the same speaker or different contexts, and this will cause a probable mismatch with the training material. Martin et al. [34] at RCA Laboratories proposed, among several solutions, the use of detection of utterance endpoints to perform time alignment of speech events. Such time normalization method improved the recognition performance by reducing the time scale variability between training and testing material. In the Soviet Union, Vintsyuk [35] proposed the use of dynamic programming, also known as dynamic time warping (DTW), for time alignment of a pair of speech utterances to derive a meaningful measurement of their similarity. He also applied it to continuous speech recognition [36]. Even though his work was unknown in the research community around 1970, Sakoe and Chiba at NEC Laboratories proposed a more formal method of dynamic time warping for speech pattern matching but they only published it in an English-language journal in 1978 [37]. After such publication, several other researchers follow the method [38, 39], making it one of the main methods for speech recognition at that time [40].

The mathematical foundation of another statistical approach was in development in the 1960s. Baum and Petrie developed several concepts for hidden Markov modeling, such as the forward-backward algorithm for estimating the model parameters iteratively [41].

Until the 1960s, the main method for estimating the short-term spectrum was a filterbank. In 1965, Cooley and Tukey [42] introduced a computationally efficient form of the discrete Fourier transform: the fast Fourier transform (FFT). It is an equivalent to filterbank but much more efficient. In 1968, Oppenheim et. al. [43] proposed the cepstral analysis for speech processing that essentially estimates a smooth spectral envelope. In the late 1960's, the fundamental concepts of Linear Predictive Coding (LPC), to estimate the vocal tract response from speech waveforms, were formulated by Atal [44, 45] and Itakura [46].

In the late 1960s, John Pierce published a letter [47] that examines the motivations and progress of the speech recognition area. First, he argued that the only motivation for working on speech recognition was the money that was supporting it, not a real need in that time. He continued the letter saying that any signal processing experiment was a waste of time and money because people did not perform speech recognition, but speech understanding. Another issue that he raised was the lack of science in the speech research:

> ***"We all believe that a science of speech is possible despite the scarcity in the field of people who behave like scientists and of results that look like science. Most recognizers behave, not like scientists, but like mad inventors or untrustworthy engineers…***

Despite Pierce's criticism, it is not possible to deny that the 1960s was one of the decades with more breakthroughs (e.g., LPC, FFT, Cepstral analysis, HMM, DTW) that were important for the speech processing technology for the years to come.

In the 1970s, the Advanced Research Project Agency (ARPA) funded a five-year program of research and development on speech understanding [48]: the ARPA SUR (Speech Understanding Research). The goal of such program was to develop several

speech understanding systems that accept continuous speech from cooperative speakers. The system should recognize 1,000 words with constrained grammar yielding less than 10% semantic error. The $15 million dollar project was mainly done at three sites: System Development Corporation (SDC), Carnegie Mellon University (CMU) and Bold, Beranek & Newman (BBN). The ARPA project also included the effort from other sites to support the main work: Lincoln Laboratory, SRI International, and University of California at Berkeley. Among the systems built by the sites, CMU's Harpy [49] was the only one to deliver the requirements of the program. Harpy was able to recognize 1,011 with a reasonable accuracy (95% of sentences understood). In the Harpy system (Fig. 12), the speech was parameterized using LPC and followed by a phone template matching that was used to segment and label the speech input. Then, a graph search, based on a beam search algorithm, built the most likely sequence of words according to constraints extracted from the language. The Harpy system was the first to take advantage of a finite state network (FSN) to reduce computation and efficiently determine the closest matching string [50].

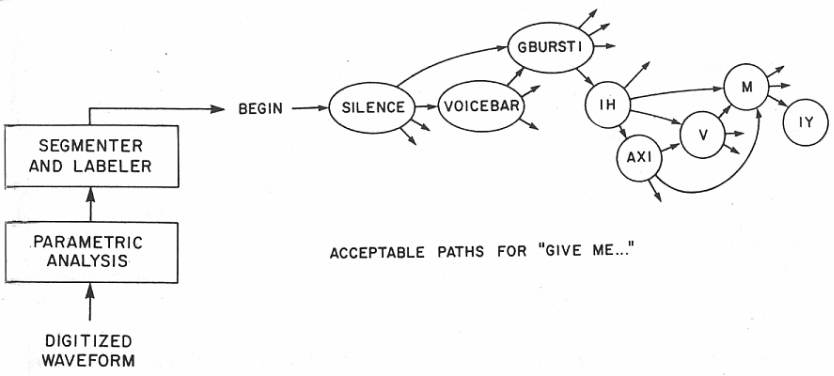

Fig. 12. A block diagram of the CMU Harpy system. It is also shown a small fragment of the state transition network for sentence beginning with "Give me" [49].

Although the other systems did not meet ARPA program goals, they also collaborate in the advance of the speech recognition technology. The CMU's Hearsay-II [51] pioneered the use of parallel asynchronous processes that simulate the component knowledge sources in a speech system. The knowledge sources performed several functions, such as extracting acoustic parameters, classifying acoustic segments into phonetic classes, recognizing words, parsing phrases, and generating and evaluating predictions for undetected words or syllables. All knowledge sources are integrated through a global "blackboard" to produce the next level of hypothesis from some type of information or evidence (in a lower level). The BBN's HWIM (Hear What I Mean) [52] incorporated phonological rules to improve phoneme recognition, handled segmentation ambiguity by a lattice of alternative hypotheses, and introduced the concept of word verification at the parametric level.

The most significant progress on the speech recognition area was the introduction of the statistical approach Hidden Markov Models (HMMs). The theory of HMM was developed in the late 1960s and early 1970s by Baum, Eagon, Petrie, Soules and

Weiss [41, 53]. The HMM was introduced into speech recognition by the researchers at IBM [54, 55], Carnegie Mellon University [54, 56], AT&T Bell Laboratories, and Institute for Defense Analyses. The main idea was that instead of storing the whole speech pattern in the memory, the units to be recognized are stored as statistical models represented by a finite state automata made of states and links among states. This approach allowed the introduction of different pronunciations for the same word and the modeling of smaller speech units like phonemes. The parameters of the model are the probability density of observing a speech feature in a given state and the probability of transitioning among states. Algorithms were proposed in the late 1960s to estimate such parameters [41] and to find an optimal path between states that matches the signal [57], similarly to DTW. The Expectation-Maximization (EM) algorithm [58] was incorporated into the modeling to allow estimating the parameters from real data.

The goal of AT&T Bell Laboratories was to provide automated telecommunication services to the public (e.g., voice dialing, and command and control for routing of phone calls) that worked well for a large number of costumers. That is, any speech recognition system should be speaker independent and could deal with different accents or pronunciations [59]. These needs led to the creation of speech clustering algorithms for creating word and sound patterns that were representative of a large population.

In the 1980s, the speech recognition systems moved from a template framework to a more elaborated statistical framework, from simple tasks (digits, phonemes) to more complex tasks (connected digits and continuous speech recognition). The complexity of speech recognition demanded a framework that integrated knowledge and allowed to decompose the problem into sub-problems (acoustic and language modeling) easier to solve [60]. The statistical framework developed in the 1980s (and all the improvements along the years) is used in most current speech recognition systems.

Despite the use of HMM in speech applications in the 1970s, such approach was really disseminated in the 1980s [61, 62]. HMM became the dominant speech recognition paradigm [63-66]. More than 30 years later, this methodology is still dominant due to the improvement efficiently incorporated.

The lack of a standard research database was a problem for many speech researchers because it made comparisons between speech recognition systems a very difficult task. Besides, the evaluation of speech recognition systems was compromised by the lack of large speech corpora. To solve these problems, a group of scientists (a joint effort among the Massachusetts Institute of Technology (MIT), Stanford Research Institute (SRI), and Texas Instruments (TI)) worked with NIST (National Institute of Standard and Technology) to develop a large corpus. The collection of the TIMIT corpus began in 1986. Such corpus is a collection of read sentences (10 sentences) that are phonetically balanced from 630 speakers. The speech was recorded at TI and transcribed at MIT (that is the origin of the name TIMIT).

Advances in speech signal representation included the perceptually motivated mel-frequency cepstral coefficients [67] and the integration of dynamic features (time derivatives of temporal trajectories)[68]. The dynamic features, known as delta cepstrum features (or just delta features), were first proposed for speaker recognition

[69], but later they were applied to speech recognition. Both representations are widely used in almost all speech recognition systems.

Artificial neural networks (ANN) re-emerged in the 1980s after a decade in the obscurity because of the book Perceptrons by Minsky and Papert [70]. Such book proved that perceptrons could not represent non-linearly separable problems. The main reason for re-emerging was the advent of the training technique (backpropagation) for multilayer perceptron (MLP) that avoided such problem. ANNs were developed to perform different types of classification in speech. For example, a time-delay neural network (this network is similar to MLP and the continuous input data is delayed and sent as an input to the neural network to consider the context information) was used for recognizing consonants [71] and phonemes [72]. Despite considerable number of work on phoneme or digit recognition, few researches applied ANN to complex tasks such as large-vocabulary continuous-speech problems [73].

In 1984, ARPA began a second program to develop a large-vocabulary, continuous-speech recognition system that yielded high word accuracy for a 1000-word database management task. The program included speaker-independent recognition. This program produced a new (read) speech corpus called Resource Management [74] with 21,000 utterances from 160 speakers, several speech recognition systems [63-66, 75, 76], and several improvements and refinements in the HMM approach for speech recognition.

In the 1990s, the development of software tools for speech recognition helped to increase the speech research community. A speech recognition tool named HTK Hidden Markov Model Toolkit was made available by the Speech Vision and Robotics Group (lead by Steve Young) of the Cambridge University Engineering Department [77]. HTK is a tool for developing large-vocabulary, speaker-independent continuous speech recognition systems (but it has also been used for other application that can benefit from the hidden Markov modeling approach). The constant improvements have made one of the most used toolkits for speech recognition research.

In another development of HMMs, Morgan and Boulard [78] demonstrated that artificial neural networks (more specifically multi-layer perceptrons) can be used to estimate the HMM state-dependent observation.

Several feature transformation methods were introduced in the 1990s. Hermansky introduces the Perceptual Linear Prediction (PLP) method [79] that modifies the speech spectrum by applying several psychophysically based spectral transformations. Several methods were proposed to alleviate channel distortion and speaker variations like RASTA filtering [80, 81] and Vocal Tract Length Normalization (VTLN) [82, 83], respectively. Kumar [84] proposed the heteroscedastic linear discriminant analysis (HLDA) that projects the feature space into a smaller space and maximally discriminative similar to the LDA, but without the assumption that the classes have equal variances.

The DARPA (Defense Advanced Research Projects Agency) program continued in the 1990s with the read speech program. After the Resource Management task, the program moved to another task: the Wall Street Journal [85]. The goal was to recognize read speech from the Wall Street Journal, with a vocabulary size as large as 60,000 words. In parallel, a speech-understanding task, called Air Travel Information

System (ATIS) [12], was developed. The goal of the ATIS task was to perform continuous speech recognition and understanding in the airline-reservation domain.

Since the early 1990s, methods for adapting the acoustic models to a specific speaker data (speaker adaptation) have been introduced. Two commonly used methods are the maximum a posteriori probability (MAP) [86, 87] and the maximum likelihood linear regression (MLLR) [88]. Other methods focused on the HMM training by shifting the paradigm of fitting the HMM to the data distribution to minimizing the recognition error, such as the minimum error discriminative training [89].

In 2000, the Sphinx group at Carnegie Mellon made available the CMU Sphinx [90], an open-source toolkit for speech recognition.

Hermansky proposed a new speech feature that is estimated from an artificial neural net [91]. The features are the posterior probabilities of each possible speech unit estimated from a multi-layer perceptron. Another feature transformation method is feature-space minimum phone error (fMPE) [92]. The fMPE transform operates by projecting from a very high-dimensional, sparse feature space derived from Gaussian posterior probability estimates to the normal recognition feature space, and adding the projected posteriors to the standard features.

In summary, a huge progress has been made in speech recognition over nearly 60 years.

Fig. **13** outlines the progress made in speech recognition and natural language understanding. Applications went from recognition of a few isolated words to recognition of continuous speech with vocabularies of tens of thousands of words. The continuous development of methods for speech processing that integrate knowledge from several areas and increasing computer power has enabled the application of speech technology in several areas. Despite all the progress, there is still the challenge of enabling machines to recognize and, more importantly, understand fluent speech in any environment or condition.

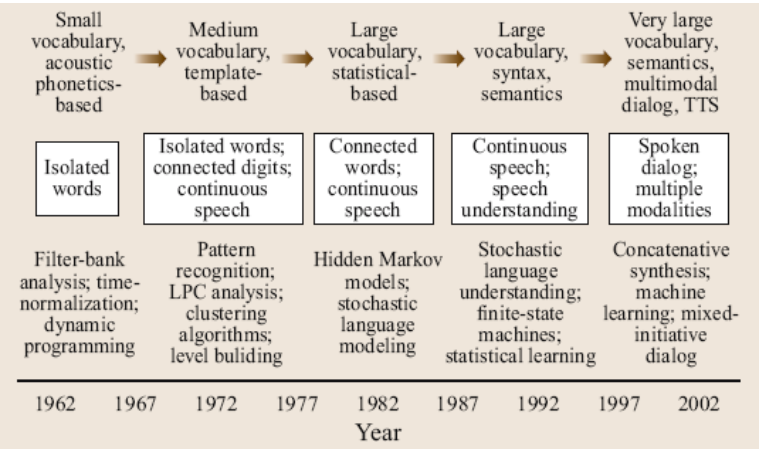

Fig. 13. Milestones in speech recognition and understanding technology over the past 40 years (from [93]).

5 Signal Processing and Feature Extraction

Every other component in a speech recognition system depends on two basic sub-systems: signal processing and feature extraction. The signal processing sub-system works on the speech signal to reduce the effects of the environment (e.g., clean versus noisy speech), the effects of the channel (e.g., cellular/land-line phone versus microphone). The feature extraction sub-system parameterizes the speech waveform so that the relevant information (in this type of application, the information about the speech units) is enhanced and the non-relevant information (age-related effects, speaker information, and so on) is mitigated. There are methods that attempt to extract parameters of a speech production model (production-based analysis), or to simulate the effect that the speech signal has on the speech perception system (perception-based analysis), or just to use a signal-based method to describe the signal in terms of its fundamental components [94].

Regardless the method employed to extract features from the speech signal, the features are usually extracted from short segments of the speech signal. This approach comes from the fact that most signal processing techniques assume stationarity of the vocal tract, but speech is nonstationary due to constant movement of the articulators during speech production. However, due to the physical limitations on the movement rate, a segment of speech sufficiently short can be considered equivalent to a stationary process. It is like if the segment is a picture taken of the speech sound during its production. In practical terms, a sliding window (with a fixed length and shape) is used to isolate each segment from the speech signal. Typically, the segments have between 20 ms and 30 ms and they are overlapped by 10 ms [7]. This approach is commonly referred to short-time analysis.

5.1 Signal-based Analysis

The methods in this type of analysis disregard how the speech was produced or perceived. The only assumption is that the signal is stationary. Two methods commonly used are filterbanks and wavelet transform.

Filterbanks estimate the frequency content of a signal using a bank of bandpass filters, whose coverage spans the frequency range of interest in the signal (e.g., 100-3000Hz for telephone speech signals, 100-8000 Hz for broadband signals). The most commonly technique for implementing a filterbank is the short-time Fourier transform (STFT). It uses a series of harmonically related basis functions (sinusoids) to describe a signal. The discrete STFT is estimated using the following equation

$$X(n,k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\frac{2\pi}{N}km} \quad (2)$$

where $w[n]$ is assumed to be non-zero only in the interval [0; N-1] and it is known as the analysis window, N is the number of sinusoidal components (which defines the frequency resolution of the analysis). The bandpass filters have center frequencies equal to the frequencies of the basis functions of the Fourier Analysis, i.e., $\omega_k = \frac{2\pi}{N}k$ [95]. The shape of the bandpass filters are frequency-shifted copies of the transfer

function of the analysis function w[n]. The drawbacks of the STFT are that all filters have the same shape, the center frequencies of the filters are evenly spaced and the properties of the function limit the resolution of the analysis [94]. Another drawback is the time-frequency resolution trade-off. A wide window produces better frequency resolution (frequency components close together can be separated) but poor time resolution. A narrower window gives good time resolution (the time at which frequencies change) but poor frequency resolution. In speech applications, the fast Fourier transform (FFT) is used to efficiently compute $X(n,k)$.

Given the STFT-based filterbank drawbacks, wavelets were introduced to allow signal analysis with different levels of resolution. This method uses sliding analysis window function that can dilate or contract, and that enables the details of the signal to be resolved depending on its temporal properties. This allows to analyze signals with discontinuities and sharp spikes. Similar to the STFT analysis, the wavelet analysis multiplies the signal of interest with a wavelet function (like the analysis window), and then the transform is computed for each segment generated. Unlike STFT, the width of the wavelet function changes with each spectral component, so that, at high frequencies, it produces good time resolution and poor frequency resolution, whereas at low frequencies, it produces gives good frequency resolution and poor time resolution. The discrete wavelet transform is estimated using the following equation

$$c_{n,m} = \int x[t] h^*_{n,m}[t] dt$$

$$h_{n,m}[t] = \frac{1}{\sqrt{a_m}} h\left(\frac{t - \tau_n}{a_m}\right)$$

where $c_{n,m}$ are the wavelet coefficients (result of the inner product between the signal $x[t]$ with the discretized wavelet basis $h_{n,m}[t]$, which are the original wavelets sampled in scale and in shift. The wavelet coefficients are analogous the coefficients of the discrete STFT, $X(n,k)$. Fig. 14 shows the time-frequency "tiles" for the STFT and the wavelet transform, respectively, that represent the essential concentration of the basis in the time-frequency plane.

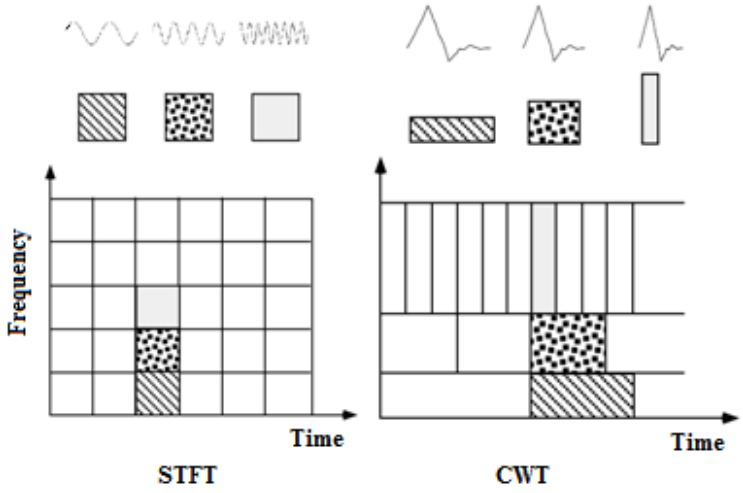

Fig. 14. Comparison of the time-frequency resolution for the STFT and the wavelet transform.

5.2 Production-based Analysis

The speech production process can be described by a combination of a source of sound energy modulated by a transfer (filter) function. This theory of the speech production process is usually referred to as the source-filter theory of speech production [94, 96]. The transfer function is determined by the shape of the vocal tract, and it can be modeled as a linear filter. However, the transfer function is changing over time to produce different sounds. The source can be classified into two types. The first one is quasi-periodic that occurs at the glottal opening. It is responsible for the production of voiced sounds (e.g., vowels, semivowels, and voiced consonants). This source can be modeled as a train of pulses. The second one is related to unvoiced excitation. In this type, the vocal folds are apart but some constriction(s) is (are) made (tongue-tip-teeth constriction for /s/, or teeth-lower-lip constriction for /f/), making difficult to the airstream pass through as easily. This source can be modeled as a random signal. Fig. 15 illustrates this speech production model, where u(t) is the source, h(t) is the filter, and s(t) is the segment of produced speech. The magnitude spectrum of each component for a voiced segment is also shown. Note that the amplitude of the harmonics of the quasi-periodic signal combines the effects of both the source spectrum (glottal pulse shape) and radiation (lip), and decreases by approximately 6dB per octave. The spectrum of the produced speech segment is shown on the right, and is the result from filtering the source spectrum with the filter function.

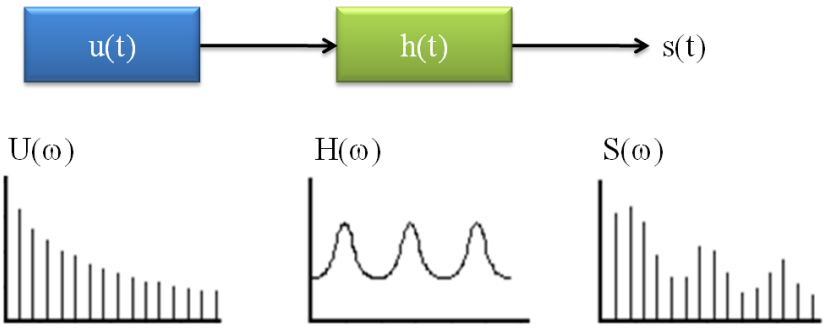

Fig. 15. Source-filter model of speech production. The source u(t) is passed through an acoustic filter h(t) resulting in speech s(t). The spectra of the source U(ω), filter H(ω), and speech output U(ω) are shown at bottom.

Despite this model is a decent approximation of the speech production, it fails on explaining the production of voiced fricatives. Voiced fricatives are produced using a mix of excitation sources: a periodic component and an aspirated component. Such mix of sources is not taken into account by the source-filter model.

Several methods take advantage of the described linear model to derive the state of the speech production system by estimating the shape of the filter function. In this section, we describe three production-based: spectral envelope, linear predictive analysis and cepstral analysis.

5.2.1 Spectral Envelope

According to the source-filter model, the spectral envelope of the transfer function would reflect the vocal tract shape to produce a given speech sound. Thus, the spectral envelope could be used to discriminate speech units that are linguistically distinct in a given language. Consequently, the goal of many speech analysis techniques is to separate the spectral envelope (filter shape) from the source. Fig. 16 shows the spectral envelope of a vowel produced by male and female speakers. The peaks in the spectral envelope correspond to the resonance frequencies (formants) of the vocal tract, which characterize a speech sound of a language. Note that both spectra have certain similarity in the overall shape of the envelope. However, the location of the peaks is different for both speakers. Differences in the dimensions of the articulators affect the formants for the same speech sound [97, 98]. In Fig. 16, the female speaker has higher formant frequencies than the male speaker due to a shorter vocal tract (this is also true for children).

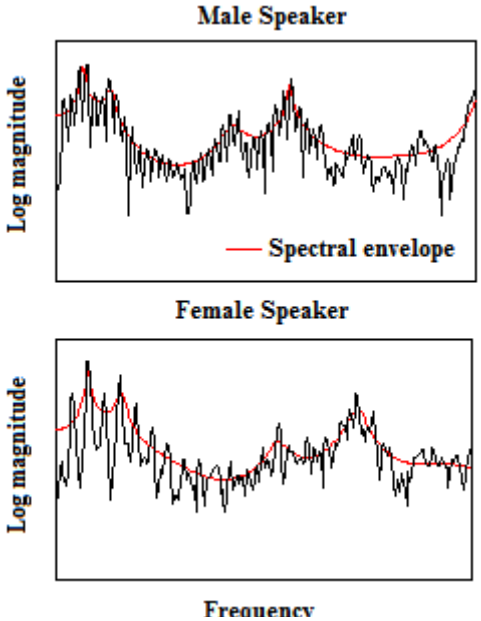

Fig. 16. Short-time spectra of the same vowel (voiced sound) produced by a male and female speaker.

5.2.2 Cepstral Analysis

According to the source-filter theory, the speech signal is the result of convolving an excitation source with the vocal tract response. Therefore, a useful speech analysis approach would be to separate (deconvolve) the two components. Usually this operation is not possible for signals in general, but it works for speech because both signals have different spectral characteristics [99]. This transformation is described by a mathematical theory called homomorphic (i.e., cepstral) processing [43, 100].

The source filter model of the speech production can be represented by the spectral magnitude of the speech signal (most speech applications require only the amplitude spectra)

$$|S(\omega)| = |U(\omega)||H(\omega)|. \quad (3)$$

The multiplication in the frequency domain of the excitation and vocal tract spectra means that the components are convolved in the time domain. Taking the logarithm of Equation (3) yields

$$log|S(\omega)| = log|U(\omega)| + log|H(\omega)|. \quad (4)$$

Equation (4) is a linear function that can be deconvolved by operations like filtering. The slowly varying components of $log|S(\omega)|$ (filter component) are represented by the low frequencies and the fine details (source component) by the high frequencies.

Hence another Fourier transform is used to separate the components of $U(\omega)$ and $H(\omega)$ and produce the cepstrum of the speech signal

$$c(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} log|S(\omega)|e^{j\omega n}\, d\omega \quad (5)$$

where $c(n)$ is called the n^{th} cepstral coefficient. The cepstral analysis estimates the spectral envelope of the filter component by truncating the cepstrum below a certain threshold [101], which is assumed to cover the filter impulse response. Fig. 17 shows a voiced speech segment, its spectrum, and the estimated cepstrum. The x-axis of the cepstrum plot is quefrency because the variable being analyzed is frequency rather than time. The transfer function usually appears as a steep slope at the beginning of the plot. The excitation appears as periodic peaks occurring after around 5ms. Note that there is a peak around 0.091 seconds, which represents the periodic excitation source of a male speaker (110 Hz). The spectral envelope is estimated using a small number of cepstral coefficients (to capture only the filter impulse response), resulting in a smooth spectral envelope. The only problem is that the smoothing performed by the cepstral analysis can remove the spectral differences between different sounds.

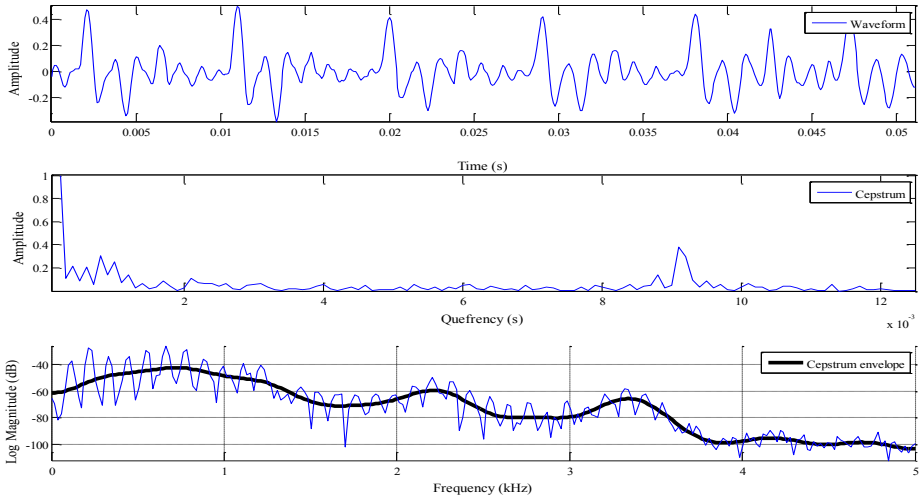

Fig. 17. Cepstral Analysis of a voiced speech segment (male speaker). The spectral envelope estimated by cepstral analysis (20 cepstral coefficients) is shown in the bottom plot.

5.2.3 Linear Predictive Analysis

The idea behind the linear predictive (LP) analysis is to represent the speech signal by time-varying parameters that are related to the vocal tract and the source [44, 102, 103]. Based on the source-filter model of speech production, the LP analysis defines that the output $s[n]$ of the acoustic filter can be approximated by a linear combination of the past p speech samples and some input excitation $u[n]$

$$s[n] = -\sum_{k=1}^{p} a_k s[n-k] + Gu[n] \quad (6)$$

where a_k for k = 1, 2, ..., p, are the predictor coefficients (also known as autoregressive coefficients because the output can be thought of as regressing itself), and G is the gain of the excitation. Since the excitation input is unknown during analysis, we can disregard the estimation of such variable and rewrite equation as:

$$\tilde{s}[n] = -\sum_{k=1}^{p} a_k s[n-k] \quad (7)$$

where $\tilde{s}[n]$ is the prediction of $s[n]$. The predictor coefficients a_k account for the filtering action of the vocal tract, the radiation and the glottal flow [45]. The transfer function of the linear filter is defined as

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \quad (8)$$

and it is also known as all-pole system function (the roots of the denominator polynomial). This function can also be used to describe another widely used model for speech production: lossless tube concatenation [95, 104]. This model is based on the assumption that the vocal tract can be represented by a concatenation of lossless tubes.

The basic problem of LP analysis is to determine the predictor coefficients a_k from the speech. The basic approach is to find the set of predictor coefficients that minimize the mean-squared prediction error of a speech segment. Given that the spectral characteristics of the vocal tract filter changes over time, the predictor coefficients are estimated over a short segment (short-time analysis).

According to Atal [45], the number of coefficients required to adequately represent any speech segment is determined by the number of resonances and anti-resonances of the vocal tract in the frequency range of interest, the nature of the glottal volume flow function, and the radiation. Fant [105] showed that, on average, the speech spectrum contains one resonant per kHz. Since such filter requires at least two coefficients (poles) for every resonant in the spectrum [94], a speech signal sampled at 10kHz would require, at least, a 10th order model. Given that LPC is an all-pole model, a couple of extra poles may be required to take care of some anti-resonances (zeros, the roots of the numerator polynomial) [23]. Gold and Morgan [7] suggested that the speech spectrum can be specified by a filter with p = 2 * (BW + 1) coefficients, where BW is the speech bandwidth in kHz. So, for our example above, the number of coefficients would be 12. Fig. 18 shows several spectra of different LPC model orders for a voiced sound. Note that a 4th order LPC model (Fig. 18a) does not efficiently represent the spectral envelope of the speech sound. The 12th order LPC model (Fig. 18b) fits efficiently the three resonances (which is a very compacted representation of the spectrum). However, as p increases (Fig. 18c and Fig. 18d), the harmonics of the spectrum are more fitted by the LPC filter. Consequently, the separation between the source and filter is reduced, which does not provide a better discrimination between different sounds.

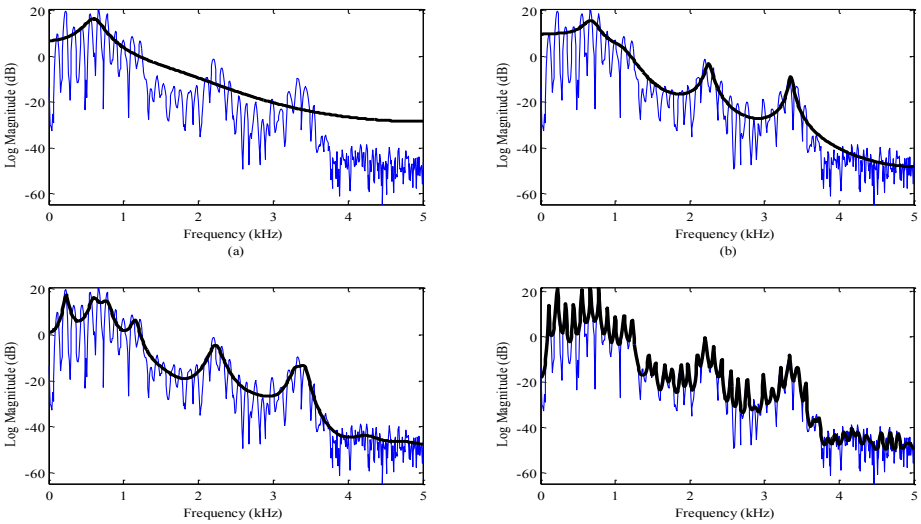

Fig. 18. Spectra of different LPC models with different model orders of a segment from /ah/ phoneme: (a) 4th order, (b) 12th order, (c) 24th order, and (d) 128th order. The spectra of the LPC analysis (thick line) are superimposed on the spectrum of the phoneme (thin line).

Despite the good fit of resonances, the LP analysis does not provide an adequate representation of all types of speech sounds. For example, nasalized sounds are poorly modeled by LPC because the production of such sounds is better modeled by a pole-zero system (the nasal cavity)[94]. Unvoiced sounds are usually over-estimated by a model with order for voiced sounds because such sounds tend to have a simpler spectral shape [7].

Different representations can be estimated from the LPC coefficients that characterize uniquely the vocal tract filter H(z) [102]. One reason for using other representations is that the LPC coefficients are not orthogonal or normalized [7]. Among the several representations [102, 106], the most common are:

- Complex poles of the filter describe the position and bandwidth of the resonance peaks of the model.
- Reflection coefficients represent the fraction of energy reflected at each section of a nonuniform tube (with as many sections as the order of the model).
- Area functions describe the shape of the hypothetical tube.
- Line spectral pairs relate to the positions and shapes of the peaks of the LP model.
- Cepstrum coefficients form a Fourier pair with the logarithmic spectrum of the model (they can be estimated through a recursion from the prediction coefficients). These parameters are orthogonal and well behaved numerically.

Besides speech recognition, the theory of LP analysis has been applied to several other speech technologies, such as, speech coding, speech synthesis, speech enhancement, and speaker recognition.

5.3 Perception-based Analysis

Perception-based analysis uses some aspects and behavior of the human auditory system to represent the speech signal. Given the human capability of decoding speech, the processing performed by the auditory system can tell us the type of information and how it should be extracted to decode the message in the signal. Two methods that have been successfully used in speech recognition are Mel-Frequency Cepstrum Coefficients (MFCC) and Perceptual Linear Prediction (PLP).

5.3.1 Mel-Frequency Cepstrum Coefficients

The Mel-Frequency Cepstrum Coefficients is a speech representation that exploits the nonlinear frequency scaling property of the auditory system [67]. This method warps the linear spectrum into a nonlinear frequency scale, called Mel. The Mel-scale attempts to model the sensitivity of the human ear and it can be approximated by

$$B(f) = 1125ln\left(1 + \frac{f}{700}\right),$$

The scale is close to linear for frequencies below 1 kHz and is close to logarithmic for frequencies above 1 kHz. The MFCC estimation is depicted in Fig. 19.

Fig. 19. Diagram of the Mel-Frequency Cepstrum Coefficients estimation.

The first step is to estimate the magnitude spectrum of the speech segment. First, the speech signal is windowed with w[n], and the discrete STFT, X(n, k), is computed according to Equation (2). Then, the magnitude of X(n, k) is weighted by a series of triangular-shaped filter frequency responses, $H_m(k)$, (whose center frequencies and bandwidths match the Mel scale) as follows

$$\Theta(m) = \sum_{k=0}^{N-1} |X(n,k)|^2 H_m(k)\,, \qquad 0 < m \leq M$$

where M is the number of filters, and $H_m(k)$ is the m^{th} filter. Fig. 20 shows an example of a mel-scale filterbank with 24 triangular-shaped frequency responses.

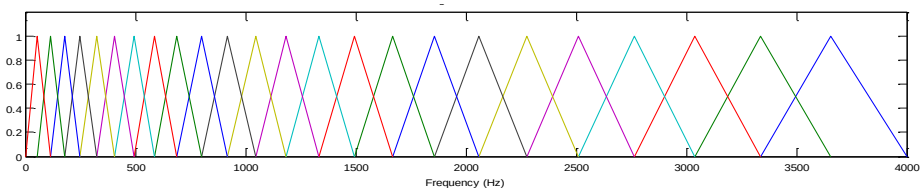

Fig. 20. Mel-scale filter bank with 24 triangular-shaped filters.

The weighting operation, $\Theta(m)$, performs two operations on the magnitude spectrum: frequency warping and critical band integration. The log-energy is computed at the output of each filter

$$S(m) = ln[\Theta(m)].$$

The mel-frequency cepstrum is then the discrete cosine transform (DCT) of the M filter outputs

$$c[n] = \sum_{m=0}^{M-1} S(m) cos\left(n\left(m - \frac{1}{2}\right)\frac{\pi}{M}\right), \qquad n = 1, 2, \ldots, L$$

where L is the desired length of the cepstrum. For speech recognition, typically only the first 13 cepstrum coefficients are used [23]. The advantage of computing the DCT is that it decorrelates the original me-scale filter log-energies [104]. One of the advantages of MFCC is that it is more robust to convolutional channel distortion [104].

5.3.2 Perceptual Linear Prediction

Conventional LP analysis approximates the areas of high-energy concentration of the spectrum (formants) in the spectrum, while smoothing out the fine harmonic structure and other less relevant spectral details. Such approximation is performed equally well at all frequencies of the analysis band, which is inconsistent with human hearing. For example, frequency resolution decreases in frequency above 800 Hz and hearing is most sensitive at middle frequency range of the audible spectrum. In order to alleviate such inconsistency, Hermansky [79] proposed a technique, called Perceptual Linear prediction, that modifies the short-term spectrum of speech by several psychophysically-based spectral transformations prior to LP analysis. The estimation of the PLP coefficients is illustrated in Fig. 21.

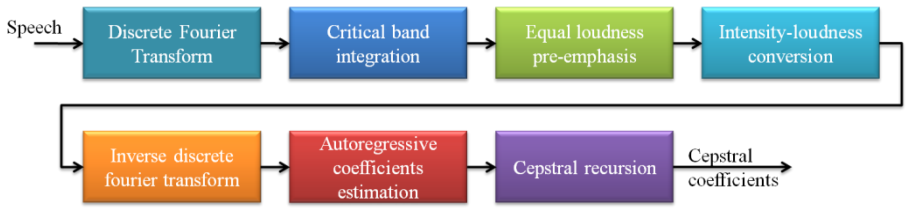

Fig. 21. Perceptual Linear Prediction estimation.

The first steps in computing the PLP and MFCC coefficients are very similar. The speech signal is windowed (e.g., Hamming window) and the discrete STFT, $X(n,k)$, is computed. Typically the FFT is used to estimate the discrete STFT. Then, the magnitude of the spectrum is computed.

The magnitude of $X(n,k)$ is integrated within overlapping critical band filter responses. Unlike the mel cepstral analysis, the integration is performed by applying trapezoid-shaped filters (an approximation of what is known about the shape of

auditory filters) to the magnitude spectrum at roughly 1-Bark intervals. Fig. 22 shows an example of a bark-scale filterbank with 14 trapezoid-shaped frequency responses. The Bark frequency Ω is derived from the frequency axis ω (radians/second) by the warping function from Schroeder [107]

$$\Omega(\omega) = 6ln\left(\frac{\omega}{1200\pi} + \sqrt{\left(\frac{\omega}{1200\pi}\right)^2 + 1}\right).$$

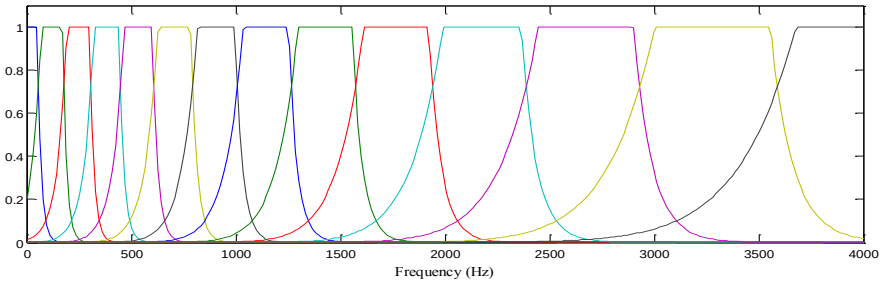

Fig. 22. Bark-scale filter bank with 14 trapezoid-shaped filters.

Some researchers suggest to use the Mel-frequency scale instead of the Bark scale to improve the system robustness to mismatched environments [108].

To compensate the unequal sensitivity of human hearing at different frequencies, the output of each filter, $\Theta(m)$, is pre-emphasized by a simulated equal-loudness curve, $E(\omega_m)$, as follows

$$\Xi(m) = E(\omega_m)\,\Theta(m)$$

where $\omega_m = 1200\pi \sinh\left(\frac{\Omega(2\pi)}{M} m/6\right)$, and $E(\omega)$ is given by

$$E(\omega) = \frac{(\omega^2 + 56.8 \times 10^6)\omega^4}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)}.$$

In MFCC analysis, pre-emphasis is applied in the time-domain.

The spectral amplitudes are compressed by taking the cubic root, as follows

$$\Phi(m) = \Xi(m)^{1/3}.$$

Typically, the compression is performed using the logarithm, but the cube root is an operation that approximates the power law of hearing and simulates the nonlinear. This operation together with the equal loudness pre-emphasis reduce the spectral-amplitude variation of the critical band spectrum so that the LP analysis can be performed using a low order model.

Finally, $\Phi(m)$ is approximated by the spectrum of an all-pole model using the autocorrelation method. Since the logarithm has not been computed, the inverse DFT of $\Phi(m)$ yields a result more like autocorrelation coefficients (since the power spectral are real and even, only the cosine components of the inverse DFT is computed). An autoregressive model is used to smooth the compressed critical band

spectrum. The prediction coefficients can be further transformed into the cepstral coefficients using the cepstral recursion.

5.4 Methods for Robustness

Although the described speech representations provide smooth estimates of the short-term spectrum, other methods are applied to such parameters to provide robustness in ASR applications. For example, the assumption of a stationary model in the short-term analysis does not take into account the dynamics of the vocal tract. In addition, any short-term spectrum based method is susceptible to convolutive effects in the speech signal introduced by the frequency response of the communication channel. Three methods that increased the robustness of ASR systems are described: delta features, RASTA filtering and Cepstral Means Subtraction.

A method widely used to model the dynamics of the speech signal is the temporal derivatives of acoustic parameters [109, 110]. Typically, feature vectors are augmented with the first and second temporal derivatives of the short-term spectrum or cepstrum, which corresponds to the velocity and the acceleration of the temporal trajectory, respectively. The velocity component is usually referred to delta features and the acceleration is referred to delta-delta features [111]. The delta and delta-delta features are estimated by fitting a straight line and a parabola, respectively, over a finite length window (in time) of the temporal trajectory. Typically, the delta features are estimated over a time interval between 50ms and 100ms. This processing can be seen as a filtering of the temporal trajectories. Another method that performs filtering of temporal trajectories is the RASTA processing.

Any other short-term spectrum based method is susceptible to convolutive effects in the speech signal introduced by the frequency response of the communication channel. The frequency characteristic of a communication channel is often fixed or slowly varying in time, and it shows as an additive component in the logarithmic spectrum of speech (convolutional effect). In addition, the rate of change of these components in speech often lies outside the typical rate of change of the vocal tract shape. The RASTA (RelAtive SpecTRAl) filtering exploits these differences to reduce the effects of changes in the communication channel, by suppressing the spectral components that change more slowly and faster than speech [112]. This is accomplished by applying a bandpass filter to each frequency channel, which preserves much of the phonetically important information in the feature representation. In a modification of the PLP, called RASTA-PLP, the filtering is applied on the log of each critical band trajectory and then followed by an exponentiation [113, 114]. RASTA approaches are discussed in much greater detail in [112].

Another method that performs some filtering in the logarithmic spectral domain is the cepstral mean normalization or subtraction (CMS) [69]. The CMS removes the mean of the cepstral coefficient feature vectors over some interval. This operation reduces the impact of stationary and slowly time-varying distortion. Another normalization applied to the cepstral coefficients to improve the system robustness to adverse conditions is the cepstral variance normalization (CVN) [115]. This normalization scales and limits the range of deviation in cepstral features to unity. Usually, the normalization is applied together with the mean normalization to the

sequence of feature vector. Thus, the cepstral features has zero mean and unity variance.

6 Acoustic Modeling

Acoustic models, $P(X|W)$, are used to compute the probability of observing the acoustic evidence X when the speaker utters W. One of the challenges in speech recognition is to estimate accurately such model. The variability in the speech signal due to factors like environment, pronunciation, phonetic context, physiological characteristics of the speaker make the estimation a very complex task. The most effective acoustic modeling is based on a structure referred to as Hidden Markov Models (HMM), which is discussed in this section.

6.1 Hidden Markov Models

A hidden Markov model is a stochastic finite-state automaton, which generates a sequence of observable symbols. The sequence of states is a Markov chain, i.e., the transitions between states has an associated probability called transition probability. Each state has an associated probability function to generate an observable symbol. Only the sequence of observations is visible and the sequence of states is not observable and therefore hidden; hence the name hidden Markov model. A hidden Markov model, as illustrated in Fig. 23, can be defined by

- An output observation alphabet $O = \{o_1, o_2, \dots, o_M\}$, where M is the number of observation symbols. When the observations are continuous, M is infinite.
- A state space $\Omega = \{1, 2, \dots, N\}$.
- A probability distribution of transitions between states. Typically, it is assumed that next state is dependent only upon the current state (first-order Markov assumption). This assumption makes the learning computationally feasible and efficient. Therefore, the transition probability can be defined as the matrix $\mathrm{A} = \{a_{ij}\}$, where a_{ij} is the probability of a transition from state i to state j, i.e.,

$$a_{ij} = P(s_t = j | s_{t-1} = i), \qquad 1 \leq i, j \leq N$$

  where, s_t is denoted as the state at time t.

- An output probability distribution $B = \{b_i(k)\}$ associated with each state. Also known as emission probability, $b_i(k)$ is the probability of generating symbol o_k while in state i, defined as

$$b_i(k) = P(v_t = o_k | s_t = i)$$

  where v_t is the observed symbol at time t. It is assumed that current output (observation) is statistically independent of the previous outputs (output independence assumption).

- A initial state distribution $\pi = \{\pi_i\}$, where π_i is the probability that state i is the first state in the state sequence (Markov chain),

$$\pi_i = P(s_0 = i), \qquad 1 \leq i \leq N$$

Since a_{ij}, $b_i(k)$, and π_i are all probabilities, the following constraints must be satisfied

$$a_{ij} \geq 0, \qquad \sum_{j=1}^{N} a_{ij} = 1,$$

$$b_i(k) \geq 0, \qquad \sum_{k=1}^{M} b_i(k) = 1,$$

$$\pi_i \geq 0, \qquad \sum_{j=1}^{N} \pi_i = 1, \ \forall \text{ all i, j, k.}$$

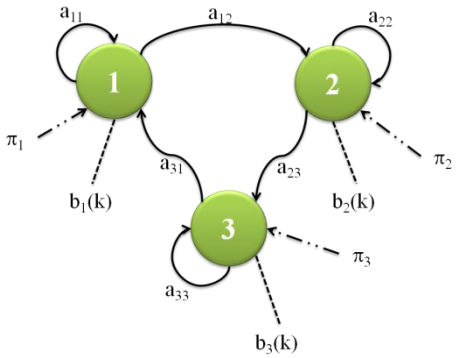

Fig. 23. A hidden Markov model with three states.

The compact notation $\lambda = (A, B, \pi)$ is used to represent an HMM. The design of an HMM includes choosing the number of states, N, as well as the number of discrete symbols, M, and estimate the three probability densities, A, B, and π.

Three problems must be solved before HMMs can be applied to real-words applications [1, 23]:

1. **Evaluation problem**: given an observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ and a model λ, how the probability of the observation sequence given the model, $P(O|\lambda)$, is efficiently computed?
2. **Decoding problem**: given an observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ and a model λ, how to choose the corresponding state sequence $S = \{s_1, s_2, \ldots, s_T\}$ that is optimal in some sense?
3. **Learning problem**: given a model λ, how to estimate the model parameters to maximize $P(O|\lambda)$?

The next sections presented formal mathematical solutions to each problem of HMM.

6.1.1 Evaluation Problem

The simplest way to compute the probability the observation sequence, $O=\{o_1, o_2, \ldots, o_T\}$, given the model λ, $P(O|\lambda)$, is summing the probabilities of all possible state sequences *S* of length *T*. That is, to sum the joint probability of *O* and *S* occur simultaneously over all possible state sequences *S*, giving

$$P(O|\lambda) = \sum_{all\ S} P(O,S|\lambda)$$
$$= \sum_{all\ S} P(O|S,\lambda)\ P(S|\lambda)$$

where $P(O|S,\lambda)$ is the probability of observing the sequence O given a particular state sequence S and $P(S|\lambda)$ is the probability of occurring such a state sequence *S*. Given the output independence assumption, $P(O|S,\lambda)$ can be written as

$$P(O|S,\lambda) = \prod_{t=1}^{T} P(o_t|s_t,\lambda)$$
$$= b_{s_1}(o_1) \cdot b_{s_2}(o_2) \ldots b_{s_T}(o_T).$$

By applying the first order Markov assumption, $P(S|\lambda)$ can be written by as

$$P(S|\lambda) = \pi_{s_1} \cdot a_{s_1 s_2} \cdot a_{s_2 s_3} \ldots a_{s_{T-1} s_T}.$$

Therefore the $P(O|\lambda)$ can be rewritten as

$$P(O|\lambda) = \sum_{all\ S} \pi_{s_1} \cdot b_{s_1}(o_1) \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2) \ldots a_{s_{T-1} s_T} \cdot b_{s_T}(o_T).$$

Note that this approach is computationally infeasible because the equation above requires $(2T - 1)N^T$ multiplications and $N^T - 1$ additions [1]. Fortunately, a more efficient algorithm, called forward algorithm, can be used to compute $P(O|\lambda)$.

The forward algorithm is a type of dynamic programming algorithm that stores intermediate values as it builds up the probability of the observation sequence. The algorithm evaluates state by state the probability of being at that state given the partial observation sequence, that is,

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, s_t = \mathrm{i}|\lambda)$$

where $\alpha_t(i)$ is the probability of the partial observation sequence in state *i* at time *t*, given the model λ. The variable $\alpha_t(i)$ can be solved inductively, as follows

1. Initialization

$$\alpha_1(i) = \pi_i \cdot b_i(o_1), \qquad 1 \le i \le N.$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}\right], \qquad 1 \le t \le T-1, 1 \le j \le N.$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

The forward algorithm has a complexity of $O(N^2T)$, which is much better than an exponential complexity. Typically, temporal constraint is assumed in speech recognition

systems, that is, the state transitions have some temporal order, usually left to right. Thus, HMMs for speech applications have a final state (s_F), altering the termination step of the forward algorithm to $P(O|\lambda) = \alpha_T(s_F)$.

6.1.2 Decoding Problem

An approach to find the optimal state sequence for a given observation sequence is to choose the states s_t that are individually most likely at each time t. Even though this approach maximizes the expected number of correct states, the estimated state sequence can have transitions that are not likely or impossible to occur (i.e., a_{ij}=0). The problem is that the approach does not take into account the transition probabilities. A modified version of the forward algorithm, known as the Viterbi algorithm, can be used to estimate the optimal state sequence

The Viterbi algorithm estimates the probability that the HMM is in state j after seeing the first t observations, like in the forward algorithm, but only over the most likely state sequence $s_1, s_2, \dots, s_{t-1}$, given the model λ, that is,

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} P\left(s_1, s_2, \dots, s_{t-1}, s_t = \mathrm{i}, o_1, o_2, \dots, o_t \middle| \lambda\right)$$

where $\delta_t(i)$ is the probability of the most likely state sequence in state i at time t after seeing the t observations. An array $\psi_t(t)$ is used to keep track of the previous state with highest probability so the state sequence can be retrieved at the end of the algorithm. The Viterbi algorithm can be defined as follows:

1. Initialization

$$\begin{aligned} \delta_1(i) &= \pi_i \cdot b_i(o_1), \qquad 1 \le i \le N. \\ \psi_t(t) &= 0. \end{aligned}$$

2. Recursion

$$\begin{aligned} \delta_t(j) &= \max_{1 \le i \le N} \left[\delta_{t-1}(i) \cdot a_{ij}\right] \cdot b_j(o_t), \\ \psi_t(j) &= \operatorname*{argmax}_{1 \le i \le N} \left[\delta_{t-1}(i) \cdot a_{ij}\right], \qquad 2 \le t \le T, 1 \le j \le N. \end{aligned}$$

3. Termination

$$\begin{aligned} P^* &= \max_{1 \le i \le N} [\delta_T(i)], \\ s_t{}^* &= \operatorname*{argmax}_{1 \le i \le N} [\delta_T(i)]. \end{aligned}$$

4. Path backtracking

$$s_t{}^* = \psi_{t+1}(s_{t+1}{}^*), \qquad t = T-1, T-2, \dots, 1.$$

6.1.3 Learning Problem

The estimation of the model parameters $\lambda = (\mathrm{A}, \mathrm{B}, \pi)$ is the most difficult of the three problems, because there is no known analytical method to maximize the probability of the observation sequence in a closed form. However, the parameters can be estimated

by maximizing $P(O|\lambda)$ locally using an iterative algorithm, such as the Baum-Welch algorithm (also known as the forward-backward algorithm).

The Baum-Welch algorithm starts with an initial estimate of the transition and observation probabilities, and then use these estimated better probabilities that maximizes $P(O|\lambda)$. The algorithm uses the forward probability $\alpha_t(i)$ (in Section 6.1.1) and the complementary backward probability β. The backward probability β is defined as

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | s_t = \mathrm{i}, \lambda)$$

where $\beta_t(i)$ is the probability of seeing the partial observation sequence from time *t+1* to the end in state *i* at time *t*, given the model λ. The variable $\beta_t(i)$ can be solved inductively, as follows

1. Initialization

$$\beta_T(i) = 1/N, \qquad 1 \le i \le N.$$

2. Induction

$$\beta_t(j) = \left[\sum_{i=1}^{N} a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)\right], \qquad \begin{matrix} t = T-1, T-2, \dots, T \\ 1 \le i \le N \end{matrix}$$

Before the reestimation procedure is described, two auxiliary variables need to be defined. The first variable, $\xi_t(i,j)$, is the probability of being in state *i* at time *t*, and state *j* at time *t+1*, given the model and the observation sequence, i.e.

$$\xi_t(i,j) = P(s_t = \mathrm{i}, s_{t+1} = \mathrm{j} | \mathrm{O}, \lambda).$$

Using the definitions of the forward and backward variables, $\xi_t(i,j)$ can be rewritten as

$$\begin{aligned} \xi_t(i,j) &= \frac{P(s_t = \mathrm{i}, s_{t+1} = \mathrm{j}, \mathrm{O}|\lambda)}{P(\mathrm{O}|\lambda)} \\ &= \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=0}^{N} \alpha_t(i) \cdot a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)}. \end{aligned}$$

The second variable, $\gamma_t(i)$, defines the probability of being in state *i* at time *t*, given the model and the observation sequence. This variable can be estimated from $\xi_t(i,j)$, by summing all the probabilities of being in state *i* at time *t* and every state at time *t+1*, i.e.,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Using the above formulas, the method for reestimation of the HMM parameters can be defined as

$$\begin{aligned} \tilde{\pi}_j &= \text{expected frequency in state i at time t=1} = \gamma_1(i) \\ \tilde{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \end{aligned}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\tilde{b}_j(k) = \frac{\text{expected number of times in state } j \text{ observing symbol } v_k}{\text{expected number of transitions from state } i}$$

$$= \frac{\sum_{t=1, o_t = v_k}^{T} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)}.$$

The reestimated model is $\tilde{\lambda} = (\tilde{\mathrm{A}}, \tilde{\mathrm{B}}, \tilde{\pi})$, and it is more likely than the model λ (i.e., $P(O|\tilde{\lambda}) > P(O|\lambda)$). Based on the above method, the model λ is replaced by $\tilde{\lambda}$ and the reestimation is repeated. This process can iterate until some limiting point is reached (usually is local maxima).

One issue in the HMM reestimation is that the forward and backward probabilities tend exponentially to zero for sufficiently large sequences. Thus, such probabilities will exceed the precision range of any machine (underflow). An approach to avoid such problem is to incorporate a scaling procedure or to perform the computation in the logarithmic domain [1].

6.2 Hidden Markov Models for Speech Recognition

There are several aspects of the model that must be defined before applying HMMs for speech recognition. In this section, some of the aspects are reviewed: discriminative training, choice of speech unit, model topology, output distribution estimators, parameter initialization, and some adaptation techniques.

6.2.1 Discriminative Training

The standard maximum likelihood (ML) maximize the probability given the sequence of observations to derive the HMM model λ, as follows

$$\lambda_{ML} = \underset{\lambda}{\operatorname{argmax}}\, P(O|\lambda).$$

In a speech recognition problem, each acoustic class c from an inventory of C classes is represented by an HMM, with a parameter set λ_c, $c = 1, 2, \ldots, C$. The ML criterion to estimate the model parameters λ_c using the labeled training sequence O^c for the class c can be defined as

$$(\lambda_c)_{ML} = \underset{\lambda}{\operatorname{argmax}}\, P(O^c|\lambda).$$

Since each model is estimated separately, the ML criterion does not guarantee that the estimated methods are the optimal solution for minimizing the probability of recognition error. It does not take into account the discrimination ability of each model (i.e., the ability to distinguish the observations generated by the correct model from those generated by the other models). An alternative criterion that maximizes such discrimination is the maximum mutual information (MMI) criterion. The mutual information between an observation sequence O^c and the class c, parameterized by $\Lambda = \{\lambda_c\}$, $c = 1, 2, \ldots, C$, is

$$I_\Lambda(O^c, c) = \log \frac{P(O^c|\lambda_c)}{\sum_{w=1}^{C} P(O^c|\lambda_w, w)P(w)}$$
$$= \log P(O^c|\lambda_c) - \log \sum_{w=1}^{C} P(O^c|\lambda_w, w)P(w).$$

The MMI criterion is to find the entire model set Λ such that the mutual information is maximized,

$$(\Lambda)_{MMI} = \max_{\Lambda} \left\{ \sum_{c=1}^{C} I_\Lambda(O^c, c) \right\}. \quad (9)$$

Thus, the MMI criterion is maximized by making the correct model sequence likely and all the other model sequence unlikely. The implementation of the MMI is based on a variant of Baum-Welch training called Extended Baum-Welch that maximizes (**9**). Briefly, the algorithm computes the forward-backward counts for the training utterances like in the ML estimation. Then, another forward-backward pass is computed over all other possible utterances and subtract these from the counts. Note that the second step is extremely computing intensive. In practice, MMI algorithms estimate the probabilities of the second step only on the paths that occur in a word lattice (as an approximation to the full set of possible paths). MMI training can provide consistent performance improvements compared to similar systems trained with ML [116].

Rather than maximizing the mutual information, several authors have proposed the use of different criteria. The minimum classification error (MCE) criterion is designed to minimize these errors and have been shown to outperform MMI estimation on small tasks [117]. Other criterion includes to minimize the number of word level errors (minimum word error – MWE) or the number of phone level errors (minimum phone error - MPE) [92, 118].

6.2.2 Speech Unit Selection

A crucial issue for acoustic modeling is the selection of the speech units that represent the acoustic and linguistic information for the language. The speech units should at least derive the words in the vocabulary (or even new words) and be trainable (i.e., there is data enough to estimate the models). The amount of data is also related to the matter of getting the speech units. The more difficult is to extract the speech units from the speech signal, the fewer data is obtained from estimating the models.

The speech units can range from phones up to words. Whole words have been used for tasks like digit recognition. An advantage of this unit is that it captures the phonetic coarticulation within the word. However, this approach becomes prohibitive for tasks with large vocabularies (i.e., requirement of large amounts of training data, no generalizable for new words). Typically, phones or sub-phones (transition-based units such as diphone to circumvent the phonetic coarticulation problem) are used as speech units. Usually, these units are fewer than words, which present no training data problem. However the realization of a phoneme is strongly affected by the surrounding phonemes (phonetic coarticulation).

One way to reduce such effects is to model the context where the phoneme occurs. This approach, known as context-dependent phonetic modeling, has been widely used by large-vocabulary speech recognition systems. The most common kind of context-dependent model is a triphone HMM [23]. A triphone model represents a phone in a particular left and right context. For example, in the word speech, pronounced /s p iy ch/, one triphone model for /p/ is [s-p+iy], that is, /p/ is preceded by /s/ and followed by /iy/. The specificity of the model increases the number of parameters to estimate and not all triphones will have enough examples to be used in the estimation. For example, there are about 40^3 or 64,000 triphones for a phoneset with 40 phones. Certainly, not all triphones occur in any language. The problem can become more complicated when the context is modeled between words. All the possible surrounding neighboring words can produce a large number of models. Some techniques are used to deal with this problem by parameter sharing.

Another speech unit that reduces the coarticulation effect is the syllable [119]. The advantage of syllables is that they contain most of the variable contextual effects, even though the beginning and ending portions of the syllable are susceptible to some contextual effect. Chinese has about 1200 tone-dependent syllables, 50 syllables in Japanese, 30,000 syllables in English. Syllable is not suitable for English given the large number of units. To reduce the considerable number of syllables for certain languages, another type of syllable-based unit was used for speech recognition: demisyllables. A demisyllable consists of either the initial (optional) consonant cluster and some part of the vowel nucleus, or the remaining part of the vowel nucleus and the final (optional) consonant cluster [1]. English has something on the order of 2,000 demisyllables, Spanish has less than 750, and German has about 344.

Speech recognition systems that use sub-word models (i.e., phones, sub-phones, or syllables) have a list that provides the transcription of all words of the task according to the set of sub-word units [16, 120]. This list is commonly referred to as lexicon or dictionary. Used by the language model, acoustic models, and the decoder, every entry of the lexicon (word) is described as a sequence of the sub-word units. When the sub-word units are phones, the lexicon is also referred to as pronunciation dictionary or phonetic dictionary. Some phonetic dictionaries freely available include

- CMU Dictionary[4]: contains over 125,000 lexical entries for North American English;
- UFPAdic[5]: contains over 64,000 lexical entries for BP
- PRONLEX[6] contains 90,988 lexical entries and includes coverage of the Wall Street Journal, and conversational telephone speech (Switchboard and CallHome English).

6.2.3 Model Topology

Some of the issues in implementing HMMs are the number of states and the choice of transitions between states. Again, there is no deterministic answer. Given that

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[5] http://www.laps.ufpa.br/falabrasil/downloads.php
[6] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97L20

speech is a nonstationary temporal signal, a left-to-right topology is used to capture the temporal dynamics of speech. Such topology has a self transition (to account for differences in duration) and there is one transition between two adjacent states, that configures the temporal evolution of speech (i.e., the transitions allow only a path that goes from left to right, $a_{ij} = 0, j < i$). Fig. 24 illustrates a left-to-right HMM with five states. This topology is one of the most popular HMM structures used in state-of-the-art speech recognition systems. In addition to states that have an output probability distribution, it is often used states without it called null states. The goal is to facilitate the composition of larger units (e.g., words) from the sub-units [121]. The inclusion of null states requires some changes to the computation of the forward and backward probabilities.

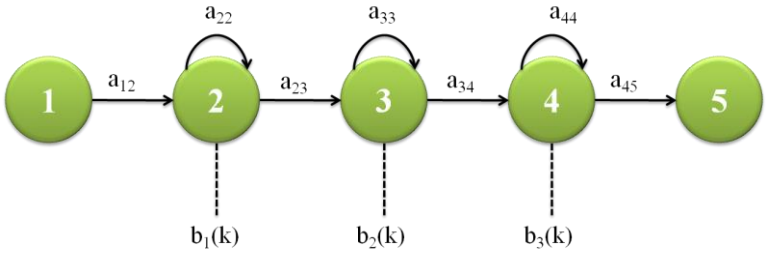

Fig. 24. Left-to-right HMM with two null states (without output probability distribution).

6.2.4 Output Probability Density Estimators

In the previous sections, the observations where characterized as discrete symbols that could be modeled at each state by a discrete probability density. The problem with this approach is that most of speech signal representations are continuous (and multi-dimensional), as seen in Section 5. Among the methods used to describe continuous observations, Gaussian densities and neural networks are commonly used.

Most speech recognition systems assume that the observations are generated by a multivariate Gaussian distribution (described by a mean vector and a covariance matrix). However, the number of parameters required to estimate the covariance matrix (dimension of each observation squared) for each state can be prohibitive. So, Gaussians with diagonal covariance (i.e., only variances) are combined to model the observation. In this density estimator, the output probability density of each state is represented by

$$b_i(o) = \sum_{k=1}^{M} c_{ik} \cdot \mathcal{N}(o, \mu_{ik} \Sigma_{\mathrm{ik}})$$

where o is the observation vector being modeled, $\mathcal{N}(x, \mu_{ik} \Sigma_{\mathrm{ik}})$ is a single Gaussian density function with mean vector μ_{ik} and covariance matrix Σ_{ik} for state i, and M denotes the number of components, and c_{ik} is the weight for the k^{th} component satisfying the stochastic and nonzero constraints. The parameter estimation for the density changes the reestimation procedure in the Baum-Welch algorithm, as following

$$\tilde{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}$$

$$\tilde{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot o_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\tilde{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (o_t - \mu_{jk})(o_t - \mu_{jk})^t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

where $\gamma_t(j,k)$ is the probability of being in state j at time t with the k[th] mixture component accounting for o_t defined as

$$\gamma_t(j,k) = \frac{c_{jk} \cdot \mathcal{N}(o_t, \mu_{jk}, \Sigma_{\text{jk}})}{\sum_{m=1}^{M} c_{jm} \cdot \mathcal{N}(o_t, \mu_{jm}, \Sigma_{\text{jm}})}.$$

Despite the reduction of parameters by using diagonal covariance matrix, there is still some considerable number of parameters. In addition, some states can share similar observation densities. To improve the parameter estimation, the distributions of different states can be tied (i.e., the same density for the tied states) according to some rule. The most common technique to select the states to tie is decision tree based on a triphone model [122]. Decision tree is a binary tree in which a question is attached to each node. The questions are related to the phonetic context to the immediate left or right. For example, in Fig. 25, the first question in the tree (root node) is: "Is the left-context phone a nasal?." A decision tree is built for each phone to cluster all the corresponding states of all triphones. Each state cluster (in the leaf nodes of the tree) will form a single state.

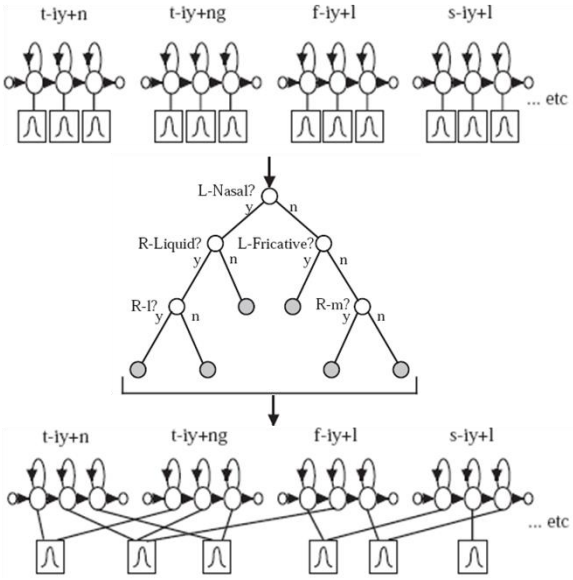

Fig. 25. Example of a tied-state HMM and a phonetic decision tree (adapted from [122]).

Artificial neural network (ANN) is another method to estimate probabilities. It has been shown that the outputs of ANNs used in classification can be interpreted as estimates of posterior probabilities of output classes conditioned on the input data [123]. The state output probability can be estimated by applying Bayes rule to the outputs [124]. The hybrid approach HMM/ANN has been used in a significant number of ASR systems [7].

6.2.5 Initial Estimates

The Baum-Welch algorithm uses an initial estimate of the transition and observation probabilities. Since the algorithm tends to a local maximum, it is important to select an initial estimate that is as close as possible to global maximum of the likelihood function.

Empirical work has shown that random (under the stochastic and nonzero value constraints) or uniform initial estimates can work reasonable well for speech applications, especially for discrete HMMs [1, 23]. However, when the observations are continuous, more sophisticated methods can be applied to produce an initial estimate. The segmented data from *k*-means clustering [125] can be used to derive the parameters (e.g., Gaussian mean and covariance) for the probability density function of each state. Another method is to equally divide the sequence of observations amongst the model states to estimate the parameters for the probability density function and then to perform a maximum likelihood segmentation of the sequence until some stopping criteria [121]. The flat-start approach sets all transitions probabilities to be equal and initializes the density parameters for each state with the parameters estimated over the data for that model [6]. Models with mixture of Gaussians densities can be estimated by incrementally splitting on each iteration the Gaussian densities.

6.2.6 Model Adaptation

Mismatches between the training and testing conditions may degrade performance of the speech recognition system. Some of the mismatches include new speakers, unseen environments or channels. The solution to these problems is to minimize the effect of such mismatches by modifying the acoustic models using some data from the unseen condition. For example, in a speaker-independent ASR system, the acoustic models can be adapted (modified) to the new speaker using training data from the new user, which could result in improved accuracy [126]. In addition to minimize the differences between the model and the new speaker, model adaptation can be used to estimate models on a limited amount of (unseen) training data. Among all the methods, the three main approaches to adaptation are:

1. **Maximum A Posteriori (MAP) adaptation** [127], the simplest form of acoustic adaptation, incorporates some prior knowledge into the estimation procedure. In ML estimation, the HMM parameter λ is assumed fixed but unknown. In the MAP estimation, λ is assumed random with a priori distribution $P_0(\lambda)$, and it can be estimated by

$$\lambda_{MAP} = \underset{\lambda}{\text{argmax}}\, P(O|\lambda)P_0(\lambda)$$

Note that the choice of the prior distribution $P_0(\lambda)$ is very important in the estimation process. The HMM parameters are still estimated with the expectation-maximization (EM) algorithm, but using the MAP formulas [127]. The MAP adaptation can be regarded as an interpolation of the original prior parameter with those that would be obtained from the adaptation data [6, 23]. One important property of MAP adaptation is that as the amount of adaptation data increases, more the estimated parameters tend to a model estimated only on sufficient adaptation data.

2. **Maximum Likelihood Linear Regression (MLLR) adaptation** [88] adjusts the Gaussian density parameters (mean vector and covariance matrix) using a set of linear regression transformation functions to increase the data likelihood of the adaptation data. Since the number of transformation parameters is small, it is possible to adapt large model with small amounts of data. It consists of finding a linear transformation (R) to adjust the Gaussian density parameters so that it maximizes the likelihood of the adaptation data, satisfying

$$\lambda_{MLLR} = \underset{\lambda}{argmax}\, P(O|\lambda, R)$$

Typically, the transformation R is applied to the model means. In the mixture Gaussian density functions, the k^{th} mean vector μ_{ik} for each state i can be transformed using the following equation

$$\tilde{\mu}_{ik} = A\mu_{ik} + b$$

where A is a regression matrix and b is an additive bias vector. The transformation parameters A and b are associated with some broad phonetic class or a set of tied Markov states, so that the number of free parameters is significantly less than the number of mean vectors. The number of transformations can be determined automatically using a regression class tree [6], where each node represents a regression class. The occupation count of each node is easily computed because the counts are known at the leaf nodes. Thus, given a set of adaptation data, the tree may be descended to an appropriate depth and a set of transformations for which there is sufficient data is selected. A modification was introduced to this method, called constrained MLLR (CMLLR) [6], so that the same linear transform is applied to the mean vector and covariance matrix.

3. **Vocal Tract Length Normalization (VTLN)** warps the frequency scale to compensate for vocal tract differences. The warping is usually applied in the acoustic processing state (similar to the Bark/Mel frequency warping of the perception-based analysis methods in Section 5.3). Typically, two issues need to be addressed: how to choose the scaling function and how to estimate the scaling function parameters (e.g., warping factor). Several approaches have been proposed [82, 83, 128]. The first VTLN methods used a simple linear mapping [128] but this approach does not take into account that, due to a shorter vocal tract, female speakers have higher formant frequencies than male speakers. This problem can be reduced by using a piecewise linear function [83]. The warping factor can be estimated by maximizing the model probability given some transcription (ML approach). It also can be derived from the signal (e.g., formant frequency [82]).

Generally, the findings are that piecewise linear models work as well as the more complex models, and that simple acoustic models can be used to estimate the warp factors.

7 Language Modeling

In the statistical framework, the sequence of words is selected by the recognizer so that it maximizes the product between the probabilities of observing the acoustic evidence X when the speaker utters W, $P(X|W)$, and the sequence of words W that will be utter, $P(W)$ in a given task. The first probability is estimated by the acoustic models, described in Section 6, and the second one is estimated by the language model. The goal of the language model is to model the sequence of words in the context of the task being performed by the speech recognition system. In continuous speech recognition, the incorporation of a language model is crucial to reduce the search space of sequence of words. In this section, algorithms for language modeling are described.

7.1 *N*-gram Language Models

The language model $P(W)$ can be decomposed as

$$\begin{aligned} P(W) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_n|w_1, w_2, \dots, w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i|w_1, w_2, \dots, w_{i-1}) \end{aligned}$$

where $P(w_i|w_1, w_2, \dots, w_{i-1})$ is the conditional probability that w_i will occur given the previous word sequence $w_1, w_2, \dots, w_{i-1}$. Unfortunately, it is impossible to compute the conditional word probabilities $P(w_i|w_1, w_2, \dots, w_{i-1})$ for all words and all sequence lengths in a given language. Even if the sequences are limited to moderate values of i, there would not be data enough to estimate reliably all the conditional probabilities. Thus, the conditional probability can be approximated by estimating the probability only on the preceding N-1 words (i.e., a Markov model of order N-1) defined by

$$P(W) \approx \prod_{i=1}^{n} P(w_i|w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1}).$$

This approximation is commonly referred to N-gram model [129]. If the model is estimated using only the two preceding words (N=3), the model is called trigram, $P(w_i|w_{i-2}, w_{i-1})$. Similarly, the model is bigram, $P(w_i|w_{i-1})$, for one preceding word (N=2), and unigram, $P(w_i)$ when no preceding word is used (N=1). The probability for the sequence "cats sleeps a lot" can be estimated as follows

Bigram model

P(cats sleep a lot) = P(cats| <START>) P(sleep|cat) P(a|sleep) P(lot|a) P(<END>|lot)

Unigram model

P(cats sleep a lot) = P(cats) P(sleep) P(a) P(lot)

Note that for the bigram model, some tokens were added to the sentence so that $P(w_i|w_{i-1})$ for i=1 is meaningful (<START> in the beginning of the sentence) and the sum of all probabilities of all strings is equal to 1 (<END > in the end of the sentence).

The conditional probabilities $P(w_i|w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1})$ can be estimated by the relative frequency that a given word w_i occurs given the preceding words $w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1}$, i.e.,

$$P(w_i|w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1}) = \frac{F(w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1}, w_i)}{F(w_{i-N+1}, w_{i-N+2}, \dots, w_{i-1})}$$

where F is the number of occurrences of the sequence of words in its argument given some training corpus (text available for building a model). The training corpus needs to be as representative of the task as possible.

Trigram language models are mostly used by large-vocabulary continuous speech recognition systems [16, 93].

7.2 Model Complexity

An approach to evaluate a language model is the word recognition error rate. However, this approach requires a working speech recognition system. Alternatively, we can measure the average number of possible words that follow any given word sequence in the language. This is the derivative measure of entropy known as test-set perplexity [1, 129]. Given a language model $P(W)$, where W is a word sequence with Q words, the entropy of the language model can be defined as

$$\begin{aligned} H(W) &= -\frac{1}{Q}\log_2 P(W) \\ &= -\frac{1}{Q}\sum_{i=1}^{Q}\log_2 P(\mathrm{w_i}|\mathrm{w_{i-N+1}}, \mathrm{w_{i-N+2}}, \dots, \mathrm{w_{i-1}}). \end{aligned}$$

Note that as Q approaches infinity, the entropy approaches the asymptotic entropy of the source defined by the measure $P(W)$. This means that the typical length of the sequence must approach infinity, which is of course impossible. Thus, $H(W)$ should be estimated on a sufficient large Q. The perplexity of the language is then defined as

$$PP(W) = 2^{H(W)} = \mathrm{P}\left(\mathrm{w_1}, \mathrm{w_2}, \dots, \mathrm{w_Q}\right)^{-1/\mathrm{Q}}.$$

For a digit recognition task (vocabulary has 10 words: ‘zero’ to ‘nine’ plus ‘oh’), where every digit can occur independently of every other digit, the language perplexity is 11. In the 5000-word Wall Street Journal task (read speech), the language perplexity is 128 for a bigram language model and 176 for a trigram language model [23]. Language models with low perplexity indicate a more predictable language. However, since the perplexity is not related to the complexity of recognizing some acoustic pattern, reducing the language model perplexity does not guarantee an improvement in speech recognition performance [7].

The perplexity can also be interpreted as the geometric mean of the word branching factor (an estimate of the size of the word list that the recognizer must chose when deciding which word was spoken).

7.3 Smoothing *N*-grams

Due to the sparseness of data, not all n-grams can be reliably estimated. For a training corpus of millions of words, and a word vocabulary of several thousand words, more than 50% of word trigrams are likely to occur either once or not at all in the training set [16]. This problem can be reduced by smoothing the *N*-gram frequencies [129].

One simple smoothing technique is to interpolate trigram, bigram, and unigram relative frequencies. Considering a trigram model (*N*=3), the interpolation is defined as

$$P(w_i|w_{i-2},w_{i-1}) = \lambda_3 \frac{F(w_{i-2},w_{i-1},w_i)}{F(w_{i-2},w_{i-1})} + \lambda_2 \frac{F(w_{i-1},w_i)}{F(w_{i-1})} + \lambda_1 \frac{F(w_i)}{\sum F(w_i)}$$

where the nonnegative weights satisfy $\lambda_1+\lambda_2+\lambda_3=1$. The smoothing probabilities, λ_1, λ_2, λ_3, are obtained by applying the principle of cross-validation. The problem with this approach is that it uses information from lower-order distributions even when the estimate of the probability of an *N*-gram is reliable.

Backoff smoothing methods provide a better smoothing than interpolation because lower-order counts are only used when an *N*-gram count is not reliable. One very famous method is the Katz smoothing (or Katz backoff) [130]. This method reduces (using a discounting factor) the unreliable probability estimates given by the observed frequencies and redistributes the discounted probability mass among the *N*-grams that never occurred in the training data. For a bigram model, Katz smoothing is defined as

$$P_{Katz}(w_i|w_{i-1}) = \begin{cases} \dfrac{F(w_{i-1},w_i)}{F(w_{i-1})} & \text{if } r > k \\ d_r \dfrac{F(w_{i-1},w_i)}{F(w_{i-1})} & \text{if } 0 < r \leq k \\ \alpha(w_{i-1})P(w_i) & \text{if } r = 0 \end{cases}$$

where r is the count for an *N*-gram w_{i-1},w_i, k is a count threshold (in the range of 5 to 8), d_r is a discount coefficient, and α is a normalization coefficient defined by

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w_i:r>0} P_{Katz}(w_i|w_{i-1})}{1 - \sum_{w_i:r>0} P(w_i)}.$$

The ML estimate is used when the *N*-gram count exceeds some threshold k (assuming that it is a reliable estimate). When the count is below the threshold and above zero, the same ML count is used but weighted by a discount factor. The discounted probability mass is then distributed among the zero-count bigrams according to the next lower-order distribution, e.g., unigram model. The discount factor is based on the Good-Turing estimate (an estimate that adjusts the count of an *N*-gram by the *N*-grams that have the same count)

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}$$

where r^* is the adjusted count of an *N*-gram that occurs *r* times. Another backoff method is the Kneser-Ney [130]. Unlike the described methods, Kneser-Ney smoothing uses a modified backoff distribution based on the number of contexts where each word occurs in, rather than the number of occurrences of the word. Another difference is that Kneser-Ney uses an absolute discounting (estimated on held out data). For a bigram model, Kneser-Ney smoothing is defined as

$$P_{KN}(w_i|w_{i-1}) = \begin{cases} \frac{\max\{F(w_{i-1}, w_i) - D, 0\}}{F(w_{i-1})} & \text{if } F(w_{i-1}, w_i) > 0 \\ \alpha(w_{i-1}) P_{KN}(w_i) & \text{otherwise} \end{cases}$$

where $P_{KN}(w_i) = \mathbb{C}(\bullet w_i)/\sum_{w_i} \mathbb{C}(\bullet w_i)$ is the number of unique words preceding w_i. The normalization coefficient α is defined by

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w_i : F(w_{i-1}, w_i) > 0} \frac{\max\{F(w_{i-1}, w_i) - D, 0\}}{F(w_{i-1})}}{1 - \sum_{w_i : F(w_{i-1}, w_i) > 0} P_{KN}(w_i)}$$

Chen and Goodman [130] proposed one additional modification to Kneser-Ney smoothing, the use of multiple discounts, one for one counts, another for two counts, and another for three or more counts. This formulation, Modified Kneser–Ney smoothing, typically outperforms the regular Kneser–Ney smoothing. More information on smoothing can be found on [130-132].

8 Decoding

The goal of the decoder is to search for the most likely word sequence *W* given some observed acoustic data *X*, that is,

$$\widetilde{W} = \underset{W \in \omega}{\operatorname{argmax}} P(W) P(X|W).$$

One approach is to search for all possible word sequences. However, for large vocabulary sizes, the search can become prohibitive (even with the current computing capability). Several techniques have been developed to reduce the computational load by using dynamic programming to perform the search and limiting the search to a small part of the search space. Therefore, it is not guaranteed that the decoder can find the most likely W.

Before details of decoding are described, it is important to note that the multiplication of the acoustic model probability and the language model probability is not performed in real applications. The problem is that HMM acoustic models usually underestimate the acoustic probability (due to independence assumption) giving to the language model little weight. A solution for such problem is to add a weight to raise

the language model probability (also known as language model scaling factor, LMSF) [8], as follows

$$\widetilde{W} = \underset{W \in \omega}{\operatorname{argmax}} P(X|W)P(W)^{LMSF}$$

where $LMSF > 1$ (between 5 and 15, in many systems) and is determined empirically to optimize the recognition performance. This weighting has a side effect as a penalty for inserting new words. A solution is to add a scaling factor that penalizes word insertions called word insertion penalty (*WIP*), as follows

$$\widetilde{W} = \underset{W \in \omega}{\operatorname{argmax}} P(X|W)P(W)^{LMSF} WIP^{N}$$

where N is the number of words in the sentence W. Thus, if the language model probability decreases (large penalty), the decoder will prefer fewer longer words. If the language model probability increases (small penalty), the decoder will prefer a greater number of shorter words instead. The insertion penalty is also determined empirically to optimize the recognition performance.

8.1 Search Space

The search space can be described by a finite state machine, where the states are the words and the transitions are defined by the language model. Fig. 26 shows an example of a finite state machine for a bigram language model. A start state was added to the model, where the transition between the state and the word states has a probability according to the language model. Each word transition has a probability equal to the corresponding bigram probability.

The combination of the language model with the acoustic models produces an HMM that models all acceptable sequence of words. The states of the HMM search space are replaced by the word HMMs. Given that the search space is now modeled by an HMM, the most likely word sequence can be found by using the Viterbi algorithm.

The complexity of the decoder is highly dependent on the complexity of the search space. In large-vocabulary continuous speech recognition, the number of words in the vocabulary produces a large state search HMM. The problem is increased by the sub-word (e.g., phonemes, syllables) modeling, commonly used in continuous speech recognition systems. In this case, each word is obtained by concatenating sub-word models. So the search space can go from thousands of states to millions of states. Thus, the decoder has to efficiently search throughout the search space.

8.2 Viterbi Search

The Viterbi search is a breadth-first search algorithm with dynamic programming. That is, all paths through the search space are pursued in parallel and gradually are pruned away as the best path (minimum cost) emerges.

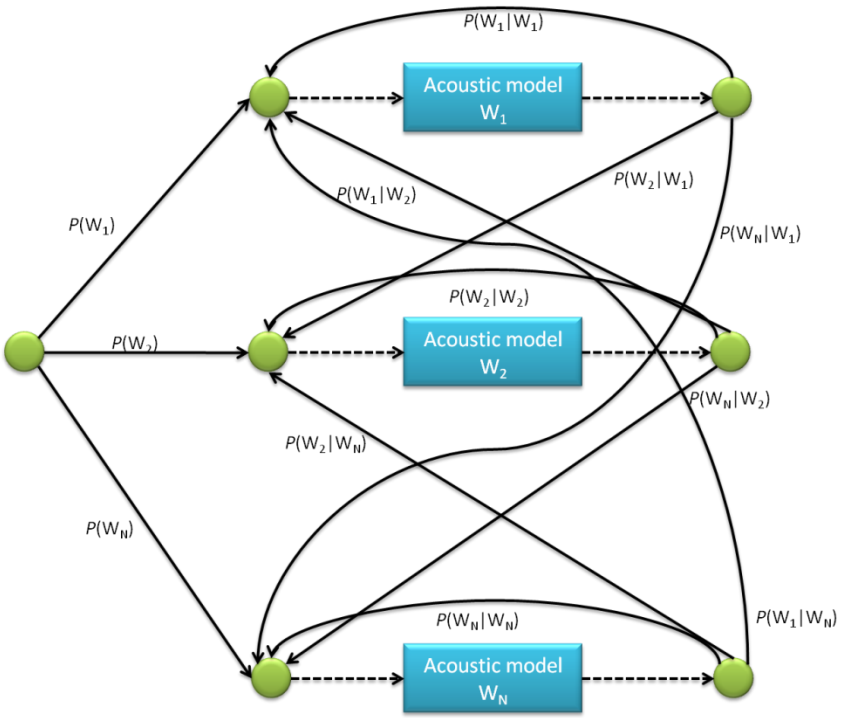

Fig. 26. Search space for a bigram language model.

The Viterbi search is time-synchronous because it completely processes time t before going on time t+1. Therefore the search can be performed in real time. For every time t, all the states are updated by the best score from all states in time t-1. Thus, each state at time t has a single best predecessor. This information allows the algorithm to determine the best state sequence for the entire search by tracking back the best predecessors at the end of the search.

The problem with Viterbi search is that it becomes computationally infeasible when the search space contains a huge number of states (for example, large vocabulary speech recognition systems). The complexity of the Viterbi search is $O(N^2T)$ (assuming that every state can transition to every state at each time step). One way to reduce such problem is limiting the search space. A widely used search technique that explores portions of the search space is the beam-search [129].

The beam-search only keeps the best partial paths. At the end of time t, the state with the highest probability P_{max} is found. Then, every other state at time t with probability less than $B \times P_{max}$, where B is a threshold (or beam width) less than 1, is excluded from consideration at time t + 1. This method significantly reduces the computational cost of the search, with little or no loss of accuracy [23]. The beam search combined with the Viterbi algorithm produces one of the most powerful search strategies for large vocabulary speech recognition.

8.3 Stack Decoding

The stack decoding algorithm is a depth-first technique based on the forward algorithm in which the most promising path is pursued until the end of the acoustic data [129]. The algorithm defines the search space as a tree where the branches correspond to words, not-terminal nodes correspond to incomplete sentences, and terminal nodes correspond to complete sentences. Thus, the stack decoding algorithm uses an objective function to search for the optimal word path in the tree search space. The algorithm can be summarized as

1. Initialize the stack with a null hypothesis (scores from all possible one-word hypotheses). Arrange the entries in descending order.
2. Pop the hypothesis with the highest score off the stack, name it as current-hypothesis.
3. If current-hypothesis is a complete sentence, output it and terminate.
4. Extend current-hypothesis by appending a word in the lexicon to its end. Compute the score of the new hypothesis and insert it into the stack. Do this for all the words in the lexicon.
5. Go to 2.

Unlike the Viterbi, the stack decoder compares the goodness of partial paths of different lengths to direct the search. Since stack decoding is asynchronous, it becomes necessary to detect when a phone/word ends, so the search can extend to the next phone/word. These two decisions are basically comparing partial theories with different lengths, so one function can be used for both decisions (for example, normalized forward probability)[23].

The problem with stack decoding is that the extension of a path implies the calculation of the probability that the immediate segment corresponds to every word/phone in the vocabulary. One way to reduce such computational burden is to extend a path by only those words that have some acoustic similarity to the observed acoustic sequence. This pruning process is referred to as fast match [129, 133]. The fast match is a computationally cheap method that selects a limited list of such words. Then, the expensive calculation can be performed on such list. On a 20,000-word dictation task, the fast match scheme was about 100 times faster than the simple stack decoding method with only 0.34% increase in word error rate [23].

8.4 *N*-best and Multipass Search

The complexity of the knowledge sources (e.g., acoustic and language models) together with the increasing size of vocabulary has been affecting the efficiency/feasibility of search algorithms by increasing the complexity of the search space. An alternative is to divide the decoding process into stages, where more refined knowledge sources are applied as the processing progresses through the stages. This processing strategy is referred to as multipass search [23]. As the decoding progresses through the stages, the set of hypotheses is reduced so that more refined and computationally demanding knowledge sources can be used to produce the most likely sequence. Therefore, the multipass strategy using gradually more refined knowledge sources could generate better results than a search algorithm with limited

models due to computation and memory constraints. For example, the first stage could use word-internal context-dependent phones with a bigram language model to generate a set of hypotheses. Then, in the second stage, cross-word context dependent phones with trigram language model could be used. This two-stage multipass decoding would produce performance comparable to a single-pass Viterbi search, but with less computational resources [134].

N-best paradigm is the most known multipass search strategy. The basic idea is to use computationally inexpensive knowledge sources to find *N* alternative sentence hypotheses. Then, each of these hypotheses is rescored with more expensive and more accurate knowledge sources in order to determine the most likely utterance. Table 6 shows an example of a 10-best list generated for a North American Business sentence.

Table 6. An example 10 –best list for a North American Business sentence (adapted from [23]).

1.	I will tell you would I think in my office
2.	I will tell you what I think in my office
3.	I will tell you when I think in my office
4.	I would sell you would I think in my office
5.	I would sell you what I think in my office
6.	I would sell you when I think in my office
7.	I will tell you that I think in my office
8.	I will tell you why I think in my office
9.	I will tell you would I think on my office
10.	I Wilson you think on my office

The *N* hypotheses can be represented by a more compact hypotheses representation: word lattice or word graphs. Fig. 27 shows the respective word lattice and word graph for the 10-best list in Table 6. Word lattices are composed by word hypothesis associated with time interval. Word graphs are directed acyclic graphs, in which arcs are labeled by words. Arcs can also carry score information such as the acoustic and language model scores. In general, word graphs are used to represent *N*-best lists, because it provides an explicit specification of word connections. Besides, word lattices and word graphs are so similar that often these terms are used interchangeably.

Several algorithms can be modified to provide an *N*-best list of hypotheses [23, 134, 135]. The stack decoding algorithm produces a complete sentence by choosing the best partial hypothesis (hoping that it will lead to the best path). Instead of selecting only the best partial hypothesis, the algorithm could select, according to the same objective function, the *N*-best hypotheses. Another algorithm that can be extended is the forward-backward. Forward-backward search algorithms use an approximate time-synchronous search in the forward direction to facilitate a more complex and expensive search in the backward direction. A simplified acoustic or language model is used to perform a fast and efficient forward-pass search in which the scores of all partial hypotheses that fall above a pruning beam width are stored. Then a normal within-word beam search is performed in the backward direction to generate a list of the *N*-best hypotheses. The backward search yields a high score on a hypothesis only if there also exists a good forward path leading to a word-ending at that instant of time.

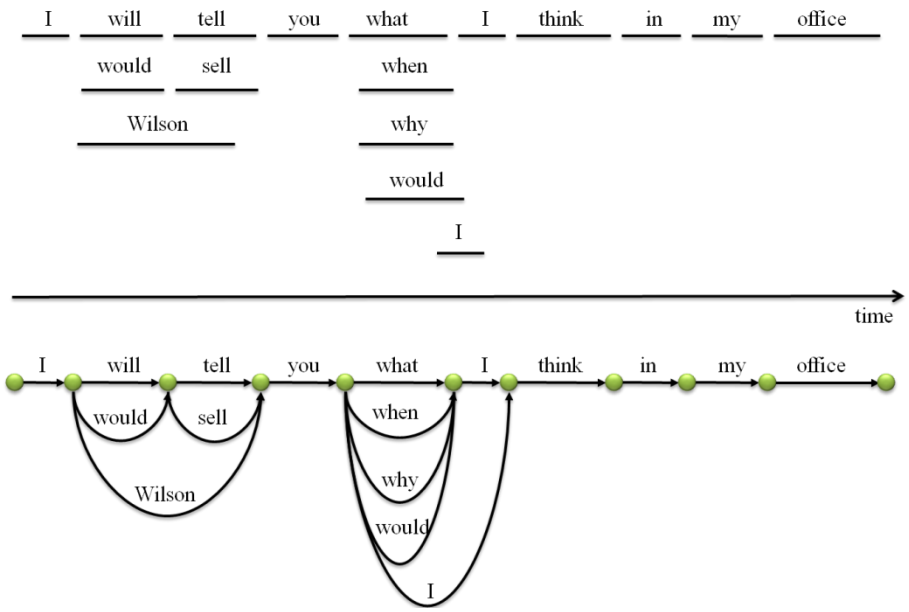

Fig. 27. Examples of word lattice (top) and word graph (bottom) for the *N*-best list in Table 6 (adapted from [23]).

9 Speech Recognition Evaluations

Since the beginning of the speech research, several speech recognition systems have been developed for all kinds of purpose. Most of the work was on tasks and speech data elaborated by the developers themselves. The problem is that it is almost impossible to replicate results to perform any type of comparison. Differences in the measurement methodology, task conditions, or testing data can lead to an erroneous comparison between systems.

The efforts from several agencies (NIST, Evaluations and Language resources Distribution Agency - ELDA[7], DARPA) and the availability of speech resources (LDC, ELRA) and has been facilitating the evaluation of speech recognition systems. The development of standard frameworks for evaluation has provided the means to researchers and developers to assess the performance of systems on standard corpora under well-defined task conditions. Such evaluations allow speech researchers to evaluate the system performance with respect to amount of training data, speaker/channel/environment variability, memory and computational requirement.

Several campaigns have been organized to perform such assessment. Besides assessing systems performance, these campaigns has been helping the research

[7] http://www.elda.org/

community to share information about the area and to show the most promising technologies. This section looks at some evaluation campaigns.

9.1 Technology and Corpora for Speech to Speech Translation - TC-STAR

TC-STAR[8] is a European integrated project focusing on Speech-to-Speech Translation (SST). To encourage significant breakthrough in all SST technologies, annual open competitive evaluations are organized. Automatic Speech Recognition (ASR), Spoken Language Translation (SLT) and Text-To-Speech (TTS) are evaluated independently and within an end-to-end system. The project targets a selection of unconstrained conversational speech domains (speeches and broadcast news) and three languages: European English, European Spanish, and Mandarin Chinese. The evaluation data comprises of recordings of the European Parliament Plenary Sessions (EPPS) (3 hours for English, 3 hour for Spanish), National Parliament Sessions in Spanish (3 hours) and Broadcast News in Mandarin (Voice of America).

Three TC-STAR evaluation campaigns took place from 2005 to 2007. In the first campaign only two core technologies were evaluated: ASR and SLT. In the remainder evaluations, all three core technologies were evaluated.

9.2 DARPA Programs

Since the 1970s, DARPA devised several speech recognition tasks with increasing complexity. The tasks challenged the speech research community resulting in several speech resources and systems throughout the years. Most of the evaluations were designed by NIST.

One of the first projects was the ARPA SUR (Speech Understanding Research) in 1971. A five-year contract project, the ARPA SUR had the goal of developing a recognition system with 90 percent sentence accuracy for continuous-speech sentences, using thousand-word vocabularies, not in real time. Of four principal ARPA SUR projects, the only one to meet the stated goal was Carnegie Mellon University's Harpy system, which achieved a 5 percent error rate on a 1,011-word vocabulary on continuous speech using a type of language model.

Toward the end of the 1980s, the collection of a new corpus for military application started a new speech recognition task. The Resource Management (RM)[74] task was to perform speech recognition to be used in a military environment in order to query a ships database about the locations and properties of naval ships throughout the world. The vocabulary was about 1000 words, and the spoken queries were read, in a sound booth, from a computer generated list of possible commands to the system. The WER at the end of the trials for this task was on the order of 2% (the only curve before 1991 in Fig. 28)[136].

After the RM task, the DARPA program moved to another read speech task: the Wall Street Journal [85]. Also knows as the North American Business (NAB) task, the goal was to recognize read speech (speaker-independent mode) from the Wall Street Journal, with a vocabulary size as large as 64,000 words (participants could submit to a 5,000-word condition). Since new words appear on a day-to-day basis in the newspaper, systems had to deal with out-of-vocabulary (OOV). The participants had to use predefined language

[8] http://www.tc-star.org

models in their systems to facilitate the comparison across sites. The systems were also evaluated on different conditions (e.g., speaker/language model adaptation and noise/channel compensation). The WER for the 5,000The WER at the end of the trials for this task was on the order of 6.6% [93].

In parallel to the WSJ task, a task for enabling users to make travel plans using spontaneous speech was developed: the Airline Travel Information System (ATIS) [12]. The goal was not only to transcribe the speech, but also to understand it so that the query could be successfully performed. Thus, systems had to deal OOV words without affecting the meaning of the already recognized speech. The task vocabulary was about 2500 words. The WER ranged was reduced from 15.7% (in 1991) to 2.5% (in 1994) [93, 136].

9.3 National Institute of Standards and Technology

Since the mid 1980s, the National Institute of Standards and Technology (NIST)[9] have been helping to advance the state-of-the-art by designing evaluation tools, coordinating periodic evaluation tests, and making data available for several speech domains. Fig. 28 shows the performance of some speech recognition systems evaluated on NIST campaigns in the past twenty years.

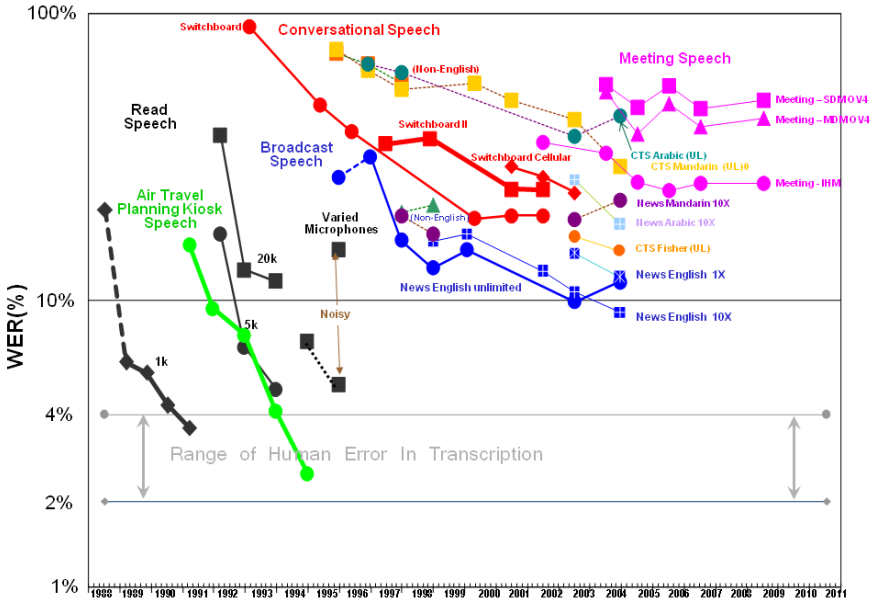

Fig. 28. NIST Speech-to-Text Benchmark Test History[10].

[9] http://www.itl.nist.gov/iad/mig/

[10] http://www.itl.nist.gov/iad/mig/publications/ASRhistory/index.html

9.3.1 NIST Rich Transcription Evaluation Project

The goal of the Rich Transcription evaluation series[11] is to create recognition technologies that will produce transcriptions which are more readable by humans and more useful for machines. This evaluation series is organized by NIST and happens since 2002.

The set of research tasks can be categorized into two main tasks:

1. Speech-to-Text Transcription (STT): consists of generating a sequence of words from audio data. Systems were evaluated using the Word Error Rate (WER) metric.
2. Metadata Extraction (MDE): consists of extracting metadata information from the audio. Some of the sub-tasks were:

   - Speaker diarization: find the segments of time in which each speaker is talking. No information about the number of speakers or training data for each speaker was available.
   - Events detection: detect events like filler word (e.g., 'hummmm', 'huh-huh'), word correction, sentence boundaries, and interruption point in broadcast news speech and conversational telephone speech in English.
   - Speaker Attributed Speech-To-Text: convert spoken words into streams of text with the speaker indicated for each word.
   - Speech Activity Detection: detect when someone in a meeting space is talking,
   - Source Localization: determine the three dimensional position of a person who is talking in a meeting space.

The conditions of the tasks included:

- Processing time: categories (based on Real Time) are used to classify the systems.
- Domain: broadcast news speech, conversational telephone speech, and meeting room speech (sub-domains include small room and a lecture room) in English. In 2004, the speech –to-text task used audio data from Chinese (Mandarin) and Arabic broadcast news and conversational telephone speech.
- Microphone (only for the meeting room speech): multiple distant microphones (Meeting-MDM curve in Fig. 28), multiple microphone arrays, single distant microphone (Meeting-SDM curve in Fig. 28) and individual head microphone (Meeting-IHM curve in Fig. 28).

The WER for the meeting room (conference room) using individual head microphone went from 32.7% in 2004 to 25.5% in 2009. As evident by the result that the meeting domain continues to be the most difficult actively researched domain for speech recognition.

9.3.2 Conversational Telephone Recognition Evaluation

The goal of the Conversation Telephone Recognition evaluation series was to evaluate the state-of-the-art in conversational speech recognition over the telephone. Four main evaluations were conducted from 1997 to 2001. Besides English,

[11] http://www.itl.nist.gov/iad/mig/tests/rt

participants could submit systems for other languages, such as, Arabic, German, Mandarin, and Spanish (not all languages were available at every evaluation).

The data for testing and training came from the Switchboard [13] and Callhome [14] corpora. The data for the non-English language came from the set of Callhome corpora. The evaluation data for each non-English language was about 20 conversations (each conversation is on average 5 minutes long) and about 20 conversations for the English task. Since the data are recorded conversations, each conversation is represented as a sequence of "turns", where each turn is the period of time when one speaker is speaking. The beginning and ending times of each of these turns were supplied as side information to the recognition system.

All the evaluations for the English language were mainly ran on the Switchboard corpus. This corpus consists or recordings in a telephone-based discussion over topics that were selected by an automated system. Most of the data comes from college students around United States. The first use of Switchboard data for speech recognition was in 1993, with a reported error of 90%. In 1995, the performance improved to 48% and by 2001 the performance reached the 19% WER. These results can be compared in the curve labeled as Switchboard in Fig. 28. The 2001 evaluation evaluate systems on excerpts from cellular phones resulting in a 29.2% WER (curve labeled as Switchboard Cellular in Fig. 28).

9.3.3 Broadcast News Recognition Evaluation

The goal of the Broadcast News Recognition evaluation[12] is to measure objectively the state of the art and help to motivate the research on the problem of accurately transcribing broadcast news speech. The evaluation campaign occurred between 1996 and 1999. The performance measure was WER (and character error rate for Mandarin). The campaign provided annotated acoustic training/development data and language model data.

The evaluation data was excerpts from radio and television programs in English, Spanish and Mandarin. The English evaluation data was approximately three hours of television news programs from CNN, ABC, and C-SPAN, as well as news radio broadcasts from NPR and PRI. In 1997, it was included evaluation data from two languages (one hour per language): Spanish (1 radio and 2 television stations) and Mandarin (2 radio and 1 television stations). The evaluation data included a combination of read speech and spontaneous speech, as well as a combination of recording environments in broadcast studios and in the field. Given the diversity in the data in 1996, the sites had to report results on six focus conditions. The focus ranged from the fluent, apparently read, speech of news anchors, to spontaneous, disfluent, speech collected in various potentially noisy environments. In the remaining evaluations, sites had to report results only on a single evaluation set. In addition to the regular submission, sites could submit results for systems with processing time less than 10xRT in the 1998 and 1999 campaigns.

The Broadcast News evaluations started with WERs of 31% in 1996 experienced a low WER of 13% in 1998, and ending the era, in 1999 with a WER of 15% [136], as shown in Fig. 28 (Broadcast Speech curve).

[12] http://www.itl.nist.gov/iad/mig/tests/bnr/

9.4 Evaluation de Systemes de Transcription Enrichie d'Emissions Radiophoniques - ESTER

The goal of ESTER evaluation campaign[13] is to evaluate automatic broadcast news transcription systems for the French language. The ESTER campaign implemented several tasks divided into three main categories: orthographic transcription, event detection and tracking (e.g. speech vs. music, speaker tracking) and information extraction (e.g. named entity detection). ESTER is organized jointly by the Francophone Speech Communication Association (AFCP), the French Defense expertise and test center for speech and language processing (DGA/CEP), and the Evaluation and Language resources Distribution Agency (ELDA), is part of the EVALDA project dedicated to the evaluation of language technologies for the French language. The ESTER campaign [137] was held in two phases from 2003 to 2005. The second campaign, ESTER 2 [138], also was held in two phases from 2007 to 2009.

The orthographic task was further divided into two sub-tasks, according to the processing time: systems operating in real-time or less (named TTR task) and otherwise (named TRS task). In the TTR task, participants were asked to process 8 hours of data.

The training data was composed by audio from several radio stations. Some transcriptions were provided for a small part of the data. Text resources from a French newspaper (around 450 million words) were provided for the development of language models. In the second campaign, more training data was available to the participants and two African radio stations were included in the task.

In the first campaign, the test set consisted of 10 hours of radio broadcast news shows taken from the same radio stations in the training set, plus an extra radio. In the second campaign, the test set consisted of 7 hours from radio stations in the training set.

The WER in the first campaign was 16.8% for the TTR task and 11.9% for the TRS task. In the second campaign, TRS was the only task and yielded a WER of 10.8%. A comparison has showed that most systems have improved; despite the test data was more difficult in the second campaign (more spontaneous speech, a larger proportion of telephone speech, the presence of strong accent and of background noises).

10 Developments in Speech Recognition

Automatic speech recognition still remains far from being a solved problem. The characteristics of the different applications (environment, noise, number of speakers, channel, vocabulary, speaker, language, and so on) impose different requirements that are not achieved by any system nowadays. However, major developments have been accomplished so that such problems can be one day solved.

The increase of computer processing has accelerated the development of the speech recognition systems. Decades ago, changes in the design of recognition systems were due to the capability of computer processing. The increase of the computer power has enabled us to run more complex algorithms and more meaningful experiments in less time.

[13] http://www.afcp-parole.org/ester/

The availability of corpora (speech and text) has enabled us to study speech and language so that better models can be created. In fact, such availability is crucial for system development because there are more ASR systems developed for English than any other language. In some languages, we are still far of having a fully functional speech recognition system because of lack of such resources. The internet has been helpful in providing text resources (electronic newspapers, magazines) for developing speech recognition systems for several languages [139-142]. Organizations such as Linguistic Data Consortium (LDC)[14], European Language Resources Association (ELRA)[15], National Institute of Standards and Technology (NIST)[16] have been collecting and distributing speech resources to research and development.

Several speech research tools are available for any researcher who wants to develop a speech recognition system. There are recognition engines (Sphinx [90], HTK [77], Julius [143], CSLU Toolkit [144], ISIP Foundation Classes [145]) that allow the development of large-vocabulary speech recognition systems or tools to build language models (CMU Statistical Language Modeling [146] and SRI Language Modeling [147]).

The development of methods and algorithms to extract information from the speech and to model speech information has been fundamental to advance the speech recognition state-of-the-art. The development of perceptually motivated speech representations MFCC and PLP provided the standard speech features for successful speech recognition systems. The feature normalizations, like CMS, RASTA and VTLN improved the robustness of such features to channel, noisy, and speaker variability. The shift to the statistical framework was the landmark to the introduction of HMMs, which has been the cornerstone of speech recognition systems for decades. The importance of HMMs increased with the development of several algorithms for training (e.g., MMI, MPE) and adaptation (e.g., MAP, MLLR). These methods produced better acoustic models allowing porting speech recognition systems to new domains or tasks. The algorithms (e.g., stack decoding) and strategies (e.g., *N*-best) of search have enabled the expansion of the vocabulary and the use of several knowledge sources to provide the best possible transcription.

The development of new methods, the increasing computing power, and the availability of speech resources has expanded the range of speech-enabled applications. The first applications were designed for isolated word recognition with a small vocabulary. Today, the effort on speech recognition is on conversational speech over telephone, broadcast news, and meeting speech that is characterized by continuous speech, different channels, multiple speakers, different languages, and large vocabulary.

Substantial effort has been employed to port speech technology to new tasks and languages. The adaptation capability of several speech recognition systems together with the increasing availability of speech and text data for a large number of languages have enable the development of speech technologies for languages other than English.

---

[14] http://www.ldc.upenn.edu

[15] http://www.elra.info

[16] http://www.itl.nist.gov/iad/mig/

10.1 Speech Recognition in Portuguese

One of the basic requirements for developing speech recognition systems for a given task or language is data (speech or text). The number of publications on speech recognition can demonstrate such issue. Researchers have complained about few data samples or no samples at all. The result is that most of the research was performed on data readily available, that is, English data. Even in countries where English is not the official language (or the second language), Systems were tested on English corpora. However, this scenario has been changing in the past decade through the cooperation among researchers around the world.

Several research centers, universities, government and private companies have been collaborating so speech can be collected and systems can be developed. Such collaborations result in more speech resources in less time and possibly at a lower cost. Consequently, more robust and reliable speech-enable applications can be developed.

In the last decade, several collaborative efforts have been started on producing speech corpora and advancing the state-of-the-art speech recognition for Portuguese languages The FalaBrasil research group has been developing and making available several resources for the Portuguese language, such as pronunciation dictionary, language and acoustic models, text and speech corpora. The Núcleo Interinstitucional de Lingüística Computacional (NILC)[17] is focused on research and development in computational linguistics and natural language processing. Some of the resources produced by NILC are text corpora and lexicons. Linguateca[18] is a distributed language resource center for Portuguese, providing resources like corpora and lexica for the Portuguese language. Recently, in a collaborative research at the Speech Digital Processing Laboratory, a BP data was collected [148]. Several projects have also been proposed so speech technology can be shared among researchers:

- POrting Speech Technologies to other varieties of Portuguese[19] (PoSTPort) (2008-2010): the goal is porting spoken language technologies originally developed for European Portuguese to other varieties of Portuguese, namely those spoken in South-American and African countries.
- Spoltech (1999-2001): the goal was to extend a speech recognition and synthesis toolkit to BP in a collaborative research from universities in Brazil and in the United States [149]. Some of the results of this collaboration are the Spoltech corpus (microphone speech from a variety of regions in Brazil with phonetic and orthographic transcriptions) and the BP CSLU toolkit[20].

One example of research group that has been performing a collaborative research to advance the speech technology for Portuguese is the internally recognized Spoken Language Systems Laboratory (L²F) at INESC-ID. Among all the contributions, the laboratory has developed a speech recognition system for European Portuguese, called Audimus. This recognizer has been the platform for several advances in speech recognition for Portuguese. An overview is given in the next section.

---

[17] http://nilc.icmc.sc.usp.br/nilc/tools/corpora.htm
[18] http://www.linguateca.pt/
[19] https://www.l2f.inesc-id.pt/wiki/index.php/PoSTPort
[20] http://www.cslu.ogi.edu/toolkit/

10.1.1 AUDIMUS Speech Recognition System

AUDIMUS is a speaker independent, large-vocabulary continuous speech recognizer [150, 151]. The system is based on a HMM model with output probabilities estimated by MLP. The structure of the system is shown in Fig. 29.

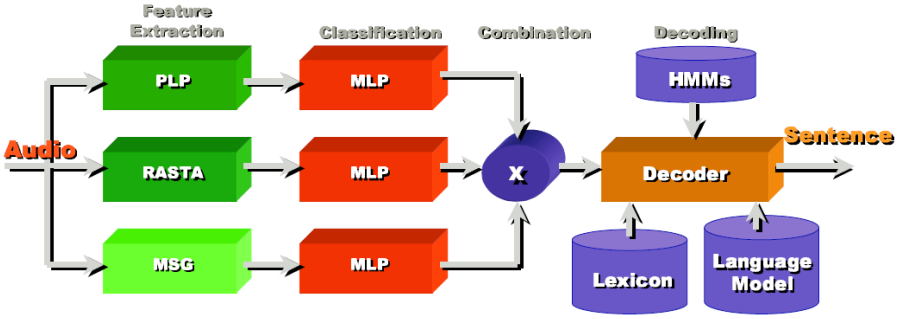

Fig. 29. AUDIMUS Speech Recognition System (adapted from [151]).

The acoustic modeling combines monophone posterior probabilities generated by three MLPs trained on three distinct feature sets. The first set consists of log-energy and 12th order PLP cepstral coefficients (plus their first time derivative) resulting in a 26-dimensional feature vector. The second set consists of log-energy and 12th order log-RASTA cepstral coefficients (plus their first time derivative) resulting in a 26-dimensional feature vector. The third set consists of 28 Modulation SpectroGram (MSG) coefficients. The MSG speech representation captures the slow modulations that encode phonetic information, critical-band frequency analysis, automatic gain control, and sensitivity to spectro-temporal peaks in the signal. The contextual information is captured by appending the adjacent feature vectors. The PLP and log-RASTA features are appended with 6 feature vectors before and after the current feature vector. The MSG feature is appended with 7 feature vectors from each side.

Each feature set is the input data for its respective MLP. Each MLP has two fully connected non-linear hidden layers with 2,000 units each and 39 softmax output units (corresponding to 38 Portuguese phones plus silence). The probabilities associated with the same phone are merged by multiplying the probability values, which internally to the decoder corresponds to perform an average in the log-probability domain.

The lexicon includes multiple pronunciations, resulting in more than 100,000 entries. The corresponding out-of-vocabulary (OOV) rate is 0.71%. A 4-gram back-off language model was created by interpolating 4-gram newspaper text language model built from over 604 million words with a 3-gram model based on the transcriptions news (approximately 51 hours from a Portuguese corpus) with 532,000 words. The language models were smoothed using Knesser-Ney discounting and entropy pruning. The perplexity obtained in a development set (approximately 6 hours of data) was 112.9.

AUDIMUS uses a dynamic decoder that builds the search space as the composition of three Weighted Finite-State Transducers (WFSTs): the HMM/MLP transducer (one single state HMM per monophone with a fixed minimum duration), the lexicon

transducer, and the language model transducer. The WER of such system in a broadcast news task (13 hours of test data) in European Portuguese was 21.5%.

In recent work [152], the system was adapted for BP (BP) broadcast news. All manually transcribed, the training data was about 851 minutes (131,000 words), development data was about 102 minutes (15,000 words) and test data was about 106 minutes (18,000 words). The corresponding out-of-vocabulary rate was 3.3% using the European Portuguese lexicon. The language model was reduced to a 3-gram back-off model (the perplexity was 197). This system achieved a WER of 26.9%.

11 Final Considerations

In the past decades, several advances in automatic speech recognition were accomplished. The technology progressed from systems that could recognize digits or a few words from only one speaker to speaker-independent, large vocabulary, continuous speech recognition. The number of deployed speech-based applications reflects the advances over the years.

The availability of speech resources and the increasing computer power has been also facilitating the development of new technologies. Today, researchers who could not develop a complete speech recognition system can focus on a specific problem by using speech recognition toolkits. Text and corpora has been distributed all over the world so better speech recognition systems can be developed. New methods and systems can be developed faster with the possibility of running several experiments.

Despite the advances, the problem is still not solved. Speech recognition systems still suffer from degraded/noisy speech. Several basic technologies (e.g., speech activity detection) need to be solved. There are not enough speech resources for every language. There are not as many computers as speech researchers want to use to perform recognition.

In summary, there is a lot of work to do before speech recognition systems can decode the linguistic message as human do under several conditions. Better understanding of the problem with the collaboration of specialists from all involved areas (engineering, computer science, phonetics, linguistics and so on) can provide a better prospect solving such problem.

References

1. Rabiner, L.R., Juang, B.H.: Fundamentals of Speech Recognition, Prentice-Hall, NJ (1993)
2. Sakoe, H.: Two-level DP-matching - a dynamic programming-based pattern matching algorithm for connected word recognition. Readings in speech recognition. Morgan Kaufmann Publishers Inc. (1990) 180-187
3. Myers, C.S., Rabiner, L.R.: A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal **60** (1981) 1389-1409

4. Bourlard, H., Wellekens, C., Ney, H.: Connected digit recognition using vector quantization. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84., Vol. 9 (1984) 413-416
5. Burton, D., Buck, J., Shore, J.: Parameter selection for isolated word recognition using vector quantization. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84., Vol. 9 (1984) 344-347
6. Young, S.: HMMs and Related Speech Recognition Technologies. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.): Springer Handbook of Speech Processing. Springer-Verlag, Heidelberg, Berlin (2008) 539-583
7. Gold, B., Morgan, N.: Speech and Audio Signal Processing: Processing and Perception of Speech and Music. John Wiley, New York (2000)
8. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Pearson Prentice Hall, Upper Saddle River, N.J. (2009)
9. Furui, S.: Toward Spontaneous Speech Recognition and Understanding. In: Wu, C., Bing Huang, J. (eds.): Pattern Recognition in Speech and Language Processing. CRC Press, Inc., Boca Raton, FL (2002) 149-190
10. Leonard, R.: A database for speaker-independent digit recognition. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84., Vol. 9 (1984) 328-331
11. Gorin, A.L., Parker, B.A., Sachs, R.M., Wilpon, J.G.: How may I help you? : Interactive Voice Technology for Telecommunications Applications, 1996. Proceedings., Third IEEE Workshop on (1996) 57-60
12. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Hidden Valley, Pennsylvania (1990)
13. Godfrey, J.J., Holliman, E.C., McDaniel, J.: SWITCHBOARD: Telephone Speech Corpus for Research and Development. ICASSP, Vol. 1, San Francisco, CA (1992) 517-520
14. CALLHOME American English Speech,http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97S42
15. Bahl, L.R., Balakrishnan-Aiyer, S., Bellegarda, J., Franz, M., Gopalakrishnan, P., Nahamoo, D., Novak, M., Padmanabhan, M., Picheny, M., Roukos, S.: Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task. ICASSP, Detroit, MI (1995) 41-44
16. Rabiner, L., Juang, B.H.: Speech Recognition: Statistical Methods. In: Brown, K. (ed.): Encyclopedia of Language & Linguistics. Elsevier (2006) 1-18
17. Flanagan, J.L.: Speech analysis, synthesis, and perception. Springer, Berlin (1965)
18. Greenberg, S.: Speech processing in the auditory system. Springer, New York (2004)
19. Gauvain, J.-L., Lamel, L.: Large Vocabulary Speech Recognition Based on Statistical Methods. In: Wu, C., Bing Huang, J. (eds.): Pattern Recognition in Speech and Language Processing. CRC Press, Inc., Boca Raton, FL (2002) 149-190
20. Wells, J.C.: SAMPA Computer Readable Phonetic Alphabet. In: Gibbon, D., Moore, R., Winski, R. (eds.): Handbook of Standards and Resources for Spoken Language Systems. Mouton de Gruyter, New York (1997) Part IV Section B
21. Barbosa, P.A., Albano, E.C.: Brazilian Portuguese. Journal of the International Phonetic Association **34** (2004) 227-232
22. Russo, I., Behlau, M.: Percepção da fala: análise acústica do português brasileiro. Lovise, São Paulo (1993)

23. Huang, X., Acero, A., Hon, H.-W.: Spoken language processing : a guide to theory, algorithm, and system development. Prentice Hall PTR, Upper Saddle River, N.J. (2001)
24. Ladefoged, P.: A Course in Phonetics. Harcourt Brace Jovanovich College Publishers, Fort Worth (1993)
25. Stüker, S., Metze, F., Schultz, T., Waibel, A.: Integrating Multilingual Articulatory Features into Speech Recognition. Eurospeech, Geneva, Switzerland (2003) 1033-1036
26. Livescu, K., Cetin, O., Hasegawa-Johnson, M., King, S., Bartels, C., Borges, N., Kantor, A., Lal, P., Yung, L., Bezman, A., Dawson-Haggerty, S., Woods, B., Frankel, J., Magami-Doss, M., Saenko, K.: Articulatory Feature-Based Methods for Acoustic and Audio-Visual Speech Recognition: Summary from the 2006 JHU Summer workshop. Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, Vol. 4 (2007) IV-621-IV-624
27. Moore, B.C.J.: An introduction to the psychology of hearing. Academic Press, Amsterdam ; Boston (2003)
28. Fletcher, H.: Auditory Patterns. Reviews of Modern Physics **12** (1940) 47
29. Davis, K.H., Biddulph, R., Balashek, S.: Automatic Recognition of Spoken Digits. The Journal of The Acoustical Society of America **24** (1952) 637-642
30. Fry, D.B., Denes, P.: The Solution of Some Fundamental Problems in Mechanical Speech Recognition. Language and Speech **1** (1958) 35-58
31. Suzuki, J., Nakata, K.: Recognition of Japanese Vowels - Preliminary to the Recognition of Speech. Journal of the Radio Research Laboratories **37** (1961) 193-212
32. Nagata, K., Kato, Y., Chiba, S.: Spoken Digit Recognizer for Japanese Language. NEC Research and Development **6** (1963)
33. Sakai, T., Doshita, S.: Phonetic Typewriter. The Journal of The Acoustical Society of America **33** (1961) 1664
34. Martin, T.B., Nelson, A.L., Zadell, H.J., Speech Recognition by Feature Abstraction Techniques, Tech Report AL-TDR-64-176, Air Force Avionics Laboratory, (1964).
35. Vintsyuk, T.K.: Speech Discrimination by Dynamic Programming. Kibernetika **4** (1968) 81-88
36. Vintsyuk, T.K.: Element-wise Recognition of Continuous Speech Composed of Words from a Specified Dictionary. Kibernetika **2** (1971) 133-143
37. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on **26** (1978) 43-49
38. Bridle, J., Brown, M., Chamberlain, R.: An algorithm for connected word recognition. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '82., Vol. 7 (1982) 899-902
39. Ney, H.: The use of a one-stage dynamic programming algorithm for connected word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on **32** (1984) 263-271
40. Myers, C., Rabiner, L., Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on **28** (1980) 623-635
41. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. The Annals of Mathematical Statistics **37** (1966) 1554-1563
42. Cooley, J.W., Tukey, J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation **19** (1965) 297-301
43. Oppenheim, A.V., Schafer, R.W., Stockham, T.G., Jr.: Nonlinear filtering of multiplied and convolved signals. Proceedings of the IEEE **56** (1968) 1264-1291

44. Atal, B.S.: Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. The Journal of The Acoustical Society of America **47** (1970) 65
45. Atal, B.S., Hanauer, S.L.: Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. The Journal of The Acoustical Society of America **50** (1971) 637-655
46. Itakura, F., Saito, S.: A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies. Electronics and Communications in Japan **53(A)** (1970) 36-43
47. Pierce, J.R.: Whither speech recognition? Journal of the Acoustical Society of America **46** (1969) 1049-1051
48. Klatt, D.H.: Review of the ARPA Speech Understanding Project. The Journal of The Acoustical Society of America **62** (1977) 1345-1366
49. Lowerre, B.: The Harpy speech understanding system. Readings in speech recognition. Morgan Kaufmann Publishers Inc. (1990) 576-586
50. Furui, S.: 50 years of progress in speech and speaker recognition. 10th International Conference on Speech and Computer - SPECOM, Patras, Greece (2005) 1-9
51. Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.: The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. ACM Comput. Surv. **12** (1980) 213-253
52. Wolf, J., Woods, W.: The HWIM speech understanding system. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '77., Vol. 2 (1977) 784-787
53. Baum, L., Petrie, T., Soules, G., Weiss, N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. The Annals of Mathematical Statistics **41** (1970) 164-171
54. Baker, J.K.: Stochastic Modeling for Automatic Speech Understanding. In: Reddy, R. (ed.): Speech Recognition. Academic Press, New York (1975) 521-542
55. Jelinek, F.: Continuous speech recognition by statistical methods. Proceedings of the IEEE **64** (1976) 532-556
56. Baker, J.K.: The Dragon system - an overview. IEEE Transactions Acoustic, Speech, and Signal Processing **23** (1975) 24-29
57. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Information Theory, IEEE Transactions on **13** (1967) 260-269
58. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society **39** (1977) 1-38
59. Rabiner, L., Levinson, S., Rosenberg, A., Wilpon, J.: Speaker-independent recognition of isolated words using clustering techniques. Acoustics, Speech and Signal Processing, IEEE Transactions on **27** (1979) 336-349
60. Bahl, L.R., Jelinek, F., Mercer, R.L.: A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-5** (1983) 179-190
61. Ferguson, J.D.: Hidden Markov Analysis: An Introduction. Hidden Markov Models for Speech. Institute for Defense Analyses, Princeton (1980)
62. Levinson, S.E., Rabiner, L.R., Sondhi, M.M.: An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. Bell Systems Technical Journal **62** (1983) 1035-1074
63. Murveit, H., Cohen, M., Price, P., Baldwin, G., Weintraub, M., Bernstein, J.: SRI's DECIPHER system. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Philadelphia, Pennsylvania (1989)

64. Paul, D.B.: The Lincoln Continuous Speech Recognition system: recent developments and results. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Philadelphia, Pennsylvania (1989)
65. Schwartz, R., Barry, C., Chow, Y.-L., Derr, A., Feng, M.-W., Kimball, O., Kubala, F., Makhoul, J., Vandegrift, J.: The BBN BYBLOS Continuous Speech Recognition system. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Philadelphia, Pennsylvania (1989)
66. Lee, K.-F., Reddy, R.: Automatic Speech Recognition: The Development of the Sphinx Recognition System. Kluwer Academic Publishers (1988)
67. Davis, S.B., Mermelstein, P.: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. IEEE Transactions Acoustic, Speech, and Signal Processing **28** (1980) 357-366
68. Furui, S.: Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics. ICASSP. IEEE, Tokyo, Japan (1986) 1991-1994
69. Furui, S.: Cepstral Analysis Technique for Automatic Speaker Verification. IEEE Transactions on Acoustics, Speech and Signal Processing **29** (1981) 254-272
70. Minsky, M.L., Papert, S.: Perceptrons: an introduction to computational geometry. MIT Press, Cambridge, Mass., (1969)
71. Makino, S., Kawabata, T., Kido, K.: Recognition of consonant based on the Perceptron model. ICASSP. IEEE, Boston, MA (1983) 738-741
72. Waibel, A., Hanazawa, T., Lang, K.J.: Phoneme Recognition Using Time-Delay Neural Networks. ICASSP. IEEE, New York, N.Y., USA (1988) 107-110
73. Lippmann, R.P.: Review of neural networks for speech recognition. Neural Comput. **1** (1989) 1-38
74. Price, P., Fisher, W.M., Bernstein, J., Pallett, D.S.: The DARPA 1000-word resource management database for continuous speech recognition. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on (1988) 651-654 vol.651
75. Zue, V., Glass, J., Phillips, M., Seneff, S.: The MIT SUMMIT Speech Recognition system: a progress report. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Philadelphia, Pennsylvania (1989)
76. Lee, C.-H., Rabiner, L.R., Pieraccini, R., Wilpon, J.G.: Acoustic modeling of subword units for large vocabulary speaker independent speech recognition. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Cape Cod, Massachusetts (1989)
77. Woodland, P., Young, S.: The HTK Tied-State Continuous Speech Recognizer. EUROSPEECH. ESCA, Berlin, Germany (1993) 2207-2210
78. Morgan, N., Bourlard, H.: Continuous speech recognition using multilayer perceptrons with hidden Markov models. ICASSP. IEEE, Albuquerque, NM (1990) 413-416
79. Hermansky, H.: Perceptual linear predictive (PLP) analysis for speech. The Journal of The Acoustical Society of America **87** (1990) 1738-1752
80. Hermansky, H., Bayya, A., Morgan, N., Kohn, P.: Compensation for the effect of the communication channel in auditory-like analysis of speech RASTA-PLP. EUROSPEECH. ESCA, Geneva, Switzerland (1991) 1367-1370
81. Morgan, N., Hermansky, H.: RASTA extensions: Robustness to additive and convolutional noise. Workshop on Speech Processing in Adverse Environments, Cannes, France (1992)
82. Eide, E., Gish, H.: A parametric approach to vocal tract length normalization. Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference

Proceedings., 1996 IEEE International Conference - Volume 01. IEEE Computer Society (1996)
83. Wegmann, S., McAllaster, D., Orloff, J., Peskin, B.: Speaker normalization on conversational telephone speech. Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on, Vol. 1 (1996) 339-341 vol. 331
84. Kumar, N.: Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition. PhD thesis, Johns Hopkins University (1997)
85. Pallett, D.S., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Lund, B.A., Przybocki, M.A.: 1993 benchmark tests for the ARPA spoken language program. Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, Plainsboro, NJ (1994)
86. Lee, C.H., Gauvain, J.L.: Bayesian Adaptive Learning and MAP estimation of HMM. Kluwer Academic Publishers, Boston (1993)
87. Lee, C.H., Lin, C.H., Juang, B.H.: A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. IEEE Transactions on Signal Processing **39** (1991) 806-814
88. Leggetter, C.J., Woodland, P.C.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. Computer Speech & Language **9** (1995) 171-185
89. Juang, B.H., Katagiri, S.: Discriminative learning for minimum error classification [pattern recognition]. Signal Processing, IEEE Transactions on **40** (1992) 3043-3054
90. Sphinx-4 A speech recognizer written entirely in the Java™ programming language,http://cmusphinx.sourceforge.net/sphinx4/.
91. Hermansky, H., Ellis, D.P.W., Sharma, S.: Tandem connectionist feature extraction for conventional HMM systems. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00.), Vol. 3 (2000) 1635-1638
92. Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., Zweig, G.: fMPE: Discriminatively Trained Features for Speech Recognition. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, Vol. 1 (2005) 961-964
93. Rabiner, L., Juang, B.-H.: Historical Perspective of the Field of ASR/NLU. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.): Springer Handbook of Speech Processing. Springer-Verlag, Heidelberg, Berlin (2008) 521-537
94. Avendaño, C., Deng, L., Hermansky, H., Gold, B.: The Analysis and Representation of Speech. Speech Processing in the Auditory System (2004) 63-100
95. Rabiner, L.R., Schafer, R.W.: Digital Processing of Speech Signals. Prentice-Hall, Englewood Cliffs, N.J. (1978)
96. Fant, G.: Acoustic theory of speech production. Mouton, s'Gravenhage, (1960)
97. Peterson, G.E., Barney, H.L.: Control Methods Used in a Study of the Vowels. The Journal of The Acoustical Society of America **24** (1952) 175-184
98. Fant, G.: Speech sounds and features. MIT Press, Cambridge (1973)
99. O'Shaughnessy, D.: Speech communication : human and machine. Addison-Wesley Pub. Co., Reading, Mass. (1987)
100. Oppenheim, A.V.: Generalized Linear Filtering. In: Gold, B., Rader, C.M. (eds.): Digital processing of signals. McGraw-Hill, New York, (1969) 233-264
101. Noll, A.M.: Cepstrum Pitch Determination. The Journal of The Acoustical Society of America **41** (1967) 293-309

102. Makhoul, J.: Linear prediction: A tutorial review. Proceedings of the IEEE **63** (1975) 561-580
103. Markel, J.D., Gray, A.H.: Linear prediction of speech. Springer-Verlag, Germany (1976)
104. Quatieri, T.F.: Discrete-time speech signal processing : principles and practice. Prentice Hall PTR, Upper Saddle River, NJ (2002)
105. Fant, G.: On the predictability of formant levels and spectrum envelopes from formant frequencies. In: Halle, M., Lunt, H.G., McLean, H., van Schooneveld, C.H. (eds.): For Roman Jakobson: Essays on the Occasion of His Sixtieth Birthday. Mouton & Co, The Hague (1956) 109–120
106. Viswanathan, R., Makhoul, J.: Quantization properties of transmission parameters in linear predictive systems. IEEE Transactions on Acoustics, Speech and Signal Processing **23** (1975) 309-321
107. Schroeder, M.R.: Recognition of complex acoustic signals. In: Bullock, T.H. (ed.): Life Sciences Research Report, Vol. 55. Abakon Verlag, Berlin (1977) 323-328
108. Woodland, P.C., Gales, M.J., Pye, D., Young, S.J.: Broadcast News Transcription Using HTK ICASSP, Munich, Germany (1997) 719-722
109. Furui, S.: Comparison of Speaker Recognition Methods using Statistical Features and Dynamic Features. IEEE Transactions on Acoustics, Speech and Signal Processing **29** (1981) 342-350
110. Furui, S.: Speaker-independent isolated word recognition using dynamic features of speech spectrum. IEEE Transactions on Acoustics, Speech and Signal Processing **34** (1986) 52-59
111. Hermansky, H.: Mel cepstrum, deltas,double-deltas. - What else is new? : Robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finland (1999)
112. Hermansky, H., Morgan, N.: RASTA Processing of Speech. IEEE Transactions on Speech and Audio Processing **2** (1994) 578-589
113. Hermansky, H., Bayya, A., Morgan, N., Kohn, P.: RASTA-PLP Speech Analysis Technique. ICASSP, Vol. 1. IEEE, San Francisco (1992) 121-124
114. Koehler, J., Morgan, N., Hermansky, H., Hirsch, H.G., Tong, G.: Integrating RASTA-PLP into speech recognition. International Conference on Acoustics, Speech and Signal Processing, Vol. 1, Adelaide, Australia (1994) 421-424
115. Droppo, J., Acero, A.: Environmental Robustness. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.): Springer Handbook of Speech Processing. Springer-Verlag, Heidelberg, Berlin (2008) 653-679
116. Woodland, P.C., Povey, D.: Large scale discriminative training of hidden Markov models for speech recognition. Computer Speech & Language **16** (2002) 25-47
117. Biing-Hwang, J., Wu, H., Chin-Hui, L.: Minimum classification error rate methods for speech recognition. Speech and Audio Processing, IEEE Transactions on **5** (1997) 257-265
118. Povey, D., Woodland, P.C.: Minimum phone error and I-smoothing for improved discriminative training. Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on, Vol. 1 (2002) I-105-I-108 vol.101
119. Ganapathiraju, A., Hamaker, J., Picone, J., Ordowski, M., Doddington, G.R.: Syllable-based large vocabulary continuous speech recognition. Speech and Audio Processing, IEEE Transactions on **9** (2001) 358-366
120. Gauvain, J.L., Lamel, L.: Large-vocabulary continuous speech recognition: advances and applications. Proceedings of the IEEE **88** (2000) 1181-1200
121. Young, S.: The HTK Book. Cambridge University (1997)

122. Young, S.J., Odell, J.J., Woodland, P.C.: Tree-based state tying for high accuracy acoustic modelling. Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, Plainsboro, NJ (1994)
123. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1997)
124. Bourlard, H., Morgan, N.: Connectionist Speech Recognition - A Hybrid Approach. Kluwer Academic Publishers, Boston (1994)
125. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, 2nd ed. (1990)
126. Huang, X., Lee, K.F.: On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. Speech and Audio Processing, IEEE Transactions on **1** (1993) 150-157
127. Gauvain, J.L., Chin-Hui, L.: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. Speech and Audio Processing, IEEE Transactions on **2** (1994) 291-298
128. Kamm, T., Andreou, A.G., Cohen, J.: Vocal tract normalization in speech recognition: Compensation for systematic speaker variability. The 15th Annual Speech Research Symposium. Johns Hopkins University, Baltimore, MI (1995) 175-179
129. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge (1997)
130. Chen, S.F., Goodman, J.T.: An empirical study of smoothing techniques for language modeling. Computer Speech and Language **13** (1999) 359-393
131. Chen, S.F., Rosenfeld, R.: A survey of smoothing techniques for ME models. IEEE Transactions on Speech and Audio Processing **8** (2000) 37-50
132. Goodman, J.T.: A Bit of Progress in Language Modeling. Computer Speech and Language **14** (2001) 403-434
133. Paul, D.B.: An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Harriman, New York (1992)
134. Deshmukh, N., Ganapathiraju, A., Picone, J.: Hierarchical search for large-vocabulary conversational speech recognition: working toward a solution to the decoding problem. Signal Processing Magazine, IEEE **16** (1999) 84-107
135. Schwartz, R., Austin, S.: Efficient, high-performance algorithms for N-Best search. Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics, Hidden Valley, Pennsylvania (1990)
136. Pallett, D.S.: A look at NIST'S benchmark ASR tests: past, present, and future. Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on (2003) 483-488
137. Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.-F., Gravier, G.: The ESTER Phase II Evaluation Campaign for the Rich Transcription of French Broadcast News. Interspeech, Lisbon, Portugal (2005) 1149-1152
138. Galliano, S., Gravier, G., Chaubard, L.: The ESTER 2 Evaluation Campaign for the Rich Transcription of French Radio Broadcasts. Interspeech, Brighton, United Kingdom (2009) 283-2586
139. Matsuoka, T.: Language model acquisition from a text corpus for speech understanding. In: Hasson, R., Barlow, M., Furui, S. (eds.), Vol. 1 (1996) 413-415A
140. CETENFolha: Corpus de Extractos de Textos Electrónicos NILC/Folha de S. Paulo,http://www.linguateca.pt/CETENFolha/.
141. Hu, X., Yamamoto, H., Zhang, J., Yasuda, K., Wu, Y., Kashioka, H.: Utilization of Huge Written Text Corpora for Conversational Speech Recognition. 6th International

Symposium on Chinese Spoken Language Processing, 2008. ISCSLP '08, Kunming, China (2008) 97-100
142. AddaDecker, M., Adda, G., Gauvain, J., Lamel, L.: On the use of speech and text corpora for speech recognition in French First International Conference on Language Resources & Evaluation, Vol. II, Granada, Spain (1998)
143. Lee, A., Kawahara, T., Shikano, K.: Julius — an Open Source Real-Time Large Vocabulary Recognition Engine. EUROSPEECH, 2001, Scandinavia (2001) 1691-1694
144. Cosi, P., Hosom, J.P., Shalkwyk, J., Sutton, S., Cole, R.A.: Connected digit recognition experiments with the OGI Toolkit's neural network and HMM-based recognizers. IEEE 4th Workshop on Interactive Voice Technology for Telecommunications Applications, (IVTTA '98) (1998) 135-140
145. Ordowski, M., Deshmukh, N., Ganapathiraju, A., Hamaker, J., Picone, J.: A Public Domain Speech-to-Text System. Eurospeech, Vol. 5, Budapest, Hungary (1999) 2127-2130
146. Clarkson, P., Rosenfeld, R.: Statistical Language Modeling Using The CMU-Cambridge Toolkit. EUROSPEECH, Rhodes, Greece (1997) 2707-2710
147. Stolcke, A.: Srilm - An Extensible Language Modeling Toolkit ICSLP Vol. 2, Denver, CO, USA (2002) 901-904
148. Ynoguti, C.A., Violaro, F.: A Brazilian Portuguese Speech Database. XXVI Simpósio Brasileiro de Telecomunicações - SBrT, Rio de janeiro, Brazil (2008) 1-6
149. Adami, A.G., Barone., D.A.C.: SPOLTECH - Avanço da Tecnologia da Linguagem Humana no Brasil e Estados Unidos através da Pesquisa Colaborativa sobre Sistemas de Linguagem Falada em Português. In: Rodrigues, I., Quaresma, P. (eds.): IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR'99), Évora, Portugal (1999) 1-9
150. Meinedo, H., Caseiro, D., Neto, J., Trancoso, I.: AUDIMUS.MEDIA : A Broadcast News Speech Recognition System for the European Portuguese Language. Computational Processing of the Portuguese Language (2003) 196-196
151. Meinedo, H., Neto, J.P.: Automatic Speech Annotation and Transcription in a Broadcast News Task. Workshop on Multilingual Spoken Document Retrieval (MSDR'2003), Hong Kong (2003) 95-100
152. Abad, A., Trancoso, I., Neto, N., Ribeiro, M.d.C.G.V.: Porting an European Portuguese Broadcast News Recognition System to Brazilian Portuguese. Interspeech, Brighton, United Kingdom (2009)