

Chapter 2

Voice Analytics Process

2.1 Introduction

Analysis of a large volume of audio conversation is required to derive reliable analytics that is statistically significant. However, analyzing audio conversations manually is not only time-consuming and boring but also error prone. Automatic processing of audio conversations is the only option; this automatic mechanism of processing audio conversations to derive usable information is called Speech or Voice Analytics. So speech analytics can, at a very broad level, be defined as

The process of automatically deriving usable information by processing and analyzing a large quantum of statistically significant audio conversations between the customer and the agent to extract usable and actionable information.

Deriving usable information from speech conversation is most often a two-step process (as seen in Fig. 2.1), namely that of

- Converting the audio conversation between the customer and the agent into text transcripts, followed by
- Analysis of the text transcripts, using natural language text processing techniques, to derive usable analytics.

While several other mechanisms exist, the most common one being that of spotting *key words* and *key phrases* in the audio conversation without completely transcribing the entire audio conversation into text [1]. The common process adopted by most commercial voice analytics tools available in the market [2–5] is that of converting the audio into text and then processing text to derive useful information, this is shown in Fig. 2.1.

A comprehensive picture of call center voice analytics process is sketched in Fig. 2.2. The process flow can be summarized as

Step 1: A user initiates the call which lands on the private branch exchange (PBX) of the call center. Note that the customers can call from any device (land-line, mobile, VoIP), and from any environment (office, home, street). This

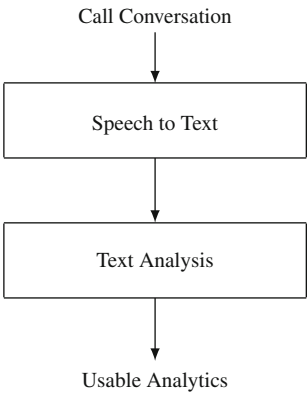


Fig. 2.1 Two-step voice analytics process

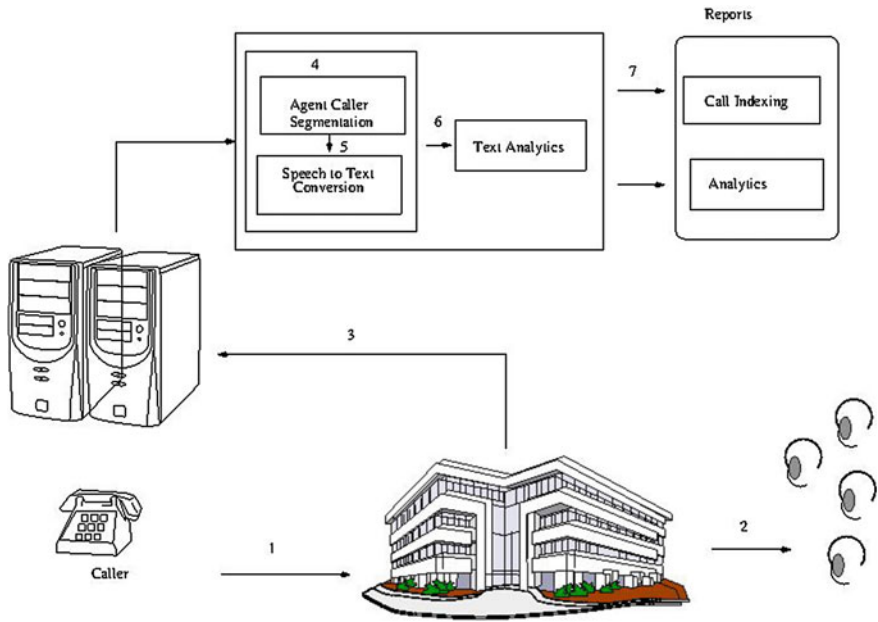


Fig. 2.2 Typical voice analytics process

is particularly important because the environment and the channel play a prominent role in analyzing audio conversation.

Step 2: After waiting in the queue (more often than not), the call lands on the agent’s desk. Very often the wait time aggravates the customer satisfaction index [6] leading to an irate customer even before the customer starts conversing with the agent.

Step 3: While the conversation is on, the call between the agent and the customer is being recorded along with some meta data (the number from where the call was made, the time of the day, etc.) on a server. These audio calls need to be analyzed for information to infer details being conversed. Typically, these calls are interlaced with speech of the customer, the agent and once in a while *hold music*.

Step 4.1: The interlaced speech is first processed to separate the hold music and voice.

Step 4.2: The voice portion is then separated into speech spoken by the agent and that spoken by the customer.

Step 5: Each of this speech segment is then passed through a speech to text converter, essentially an automatic speech recognition engine.

Step 6: The converted text is processed using text analysis tools to derive actionable and usable information.

Step 7: Capture the information embedded in the conversation which can be indexed for later search or generate reports.

So before the actual conversion of audio to text happens, there are two essential steps of preprocessing audio conversation, namely that of separating the hold music from voice and that of distinguishing the customer and the agent spoken speech. We look at this next.

2.2 Music Voice Separation

One of the important steps in analyzing call center conversation is the separation of voice and music. There are several approaches (see references in [7]) available to separate an audio which is interlaced by music and voice. One approach is to use nonlinear speech features like Teager energy [8] to obtain modulation-based features from an audio stream. These features can be used in a supervised learning scheme for segment-wise discrimination of vocal and music component in an audio stream.

Audio signal $x(t)$ is nonlinear, time-varying, and can be looked upon as a Amplitude Modulation–Frequency Modulation (AM–FM) model [8], namely

$$x(t) = a(t) \cos(\phi(t)) \quad (2.1)$$

where, $a(t)$ is the time-varying amplitude and $\phi(t)$ is defined as

$$\phi(t) = \omega_c t + \omega_m \int_0^t q(\tau) d\tau + \theta \quad (2.2)$$

where ω_c is the center frequency and ω_m is the maximum frequency deviation from the ω_c , $|q(t)| \leq 1$ and $\theta = \phi(0)$ is some arbitrary constant phase offset. The time-varying instantaneous angular frequency ω_i is defined as

$$\begin{aligned} \omega_i(t) &\stackrel{\text{def}}{=} \frac{d}{dt} \phi(t) \\ &= \omega_c + \omega_m q(t) \end{aligned} \quad (2.3)$$

Equation (2.1) has both an AM and FM structure hence $x(t)$ can be called an AM–FM signal.

It has been shown that this nonlinear modeling of speech helps in extraction of robust speech features [9]. Two different information signals can be simultaneously transmitted in the amplitude $a(t)$ and the frequency $\omega_i(t)$. The AM–FM model can be used to represent any speech signal $s(t)$ as a sum of AM–FM signals, namely

$$s(t) = \sum_{k=1}^K a_k(t) \cos(\phi_k(t)) \quad (2.4)$$

where K is the number of speech formants. Clearly, $a(t)$ and $\omega_i(t)$ for $k = 1, 2, \dots, K$ represents the speech signal $s(t)$. However, these have to be estimated from the speech signal.

Given a speech signal over some time interval, the problem is to estimate the amplitude envelope $|a(t)|$ and the instantaneous frequency $\omega_i(t)$ for each k and at each time t . One of the ways to estimate $a(t)$ and $\omega_i(t)$, is to first isolate individual resonance by bandpass filtering the speech signal around its formants and then estimating amplitude and frequency modulating signals of each resonance based on an *energy-tracking operator* as described in [10]. The Teager energy operator ψ (TEO) is defined as

$$\psi_c[x(t)] \stackrel{\text{def}}{=} \left[\frac{d}{dt} x(t) \right]^2 - x(t) \left[\frac{d^2}{dt^2} x(t) \right] \quad (2.5)$$

When ψ given by (2.5) is applied to the bandpass filtered speech signal (2.1), we get the instantaneous source energy, namely

$$\psi[x(t)] \approx a^2(t) \omega_i^2(t) \quad (2.6)$$

In the discrete form as is applicable to most speech processing systems [11], Eq. (2.5) can be written as

$$\psi[x[n]] = x^2[n] - x[n+1]x[n-1] \quad (2.7)$$

where $x[n]$ is the sampled speech signal.

The TEO is typically applied to a bandpass filtered speech signal, since its intent is to reflect the energy of the nonlinear flow within the vocal tract for a single resonant frequency. Although the output of the bandpass filter still contains more than one frequency component, it can be considered as an AM–FM signal, $r(t) = a(t) \cos(2\pi f(t)t)$. The TEO output of $r(t)$ can be approximated as

$$\psi[r(t)] \approx [a(t) 2\pi f(t)]^2 \quad (2.8)$$

The AM–FM demodulation can be achieved by separating the instantaneous energy given in (2.6) into its amplitude and frequency components. ψ is the main ingredient of the first Energy Separation Algorithm (ESA) developed in [8] and used for signal and speech AM–FM demodulation, namely

$$f[n] \approx \cos^{-1} \left(1 - \frac{\psi[y[n]] + \psi[y[n+1]]}{4\psi[x[n]]} \right) \quad (2.9)$$

and

$$|a[n]| \approx \sqrt{\frac{\psi[x[n]]}{\left[1 - \left(1 - \frac{\psi[y[n]] + \psi[y[n+1]]}{4\psi[x[n]]} \right)^2 \right]}} \quad (2.10)$$

where $y[n] = x[n] - x[n-1]$ and $f[n]$ is the FM component at sample n and $a[n]$ is the AM component at sample n . In practice, the speech signal is bandpass filtered using Gabor filters because of their optimal time-frequency discriminability [8], namely

$$s(t) = x(t) * g(t) \quad (2.11)$$

where $g(t)$ is given by

$$g(t) = \frac{1}{\sqrt{2\pi}\sigma} \left(e^{-\frac{t^2}{2\sigma^2}} \right) \left(e^{j(2\pi\omega_0 t)} \right) \quad (2.12)$$

where ω_0 is the center frequency and σ is the bandwidth of the Gabor filter.

In the case of audio signals, a Gabor filter-bank (placed at various critical band frequencies such as formant frequencies or at frequencies determined by Mel-scale) with a narrow bandwidth are used. The extraction of AM–FM components (2.9) and (2.10) from the bandpass filtered signal may be carried out using the Teager energy of the filtered signal. The efficiency of nonlinear speech features, namely instantaneous modulation features such as instantaneous amplitude and instantaneous frequencies have been studied for various applications like phoneme classification, speech recognition [9, 12], assessment of vocal fold pathology [13], stress detection [14], and music voice separation [7].

Figure 2.3 shows the typical distributions for vocal and music segments [7]. The instantaneous frequencies using three different Gabor filters ($\omega_0 = 240, 738, 1361$)

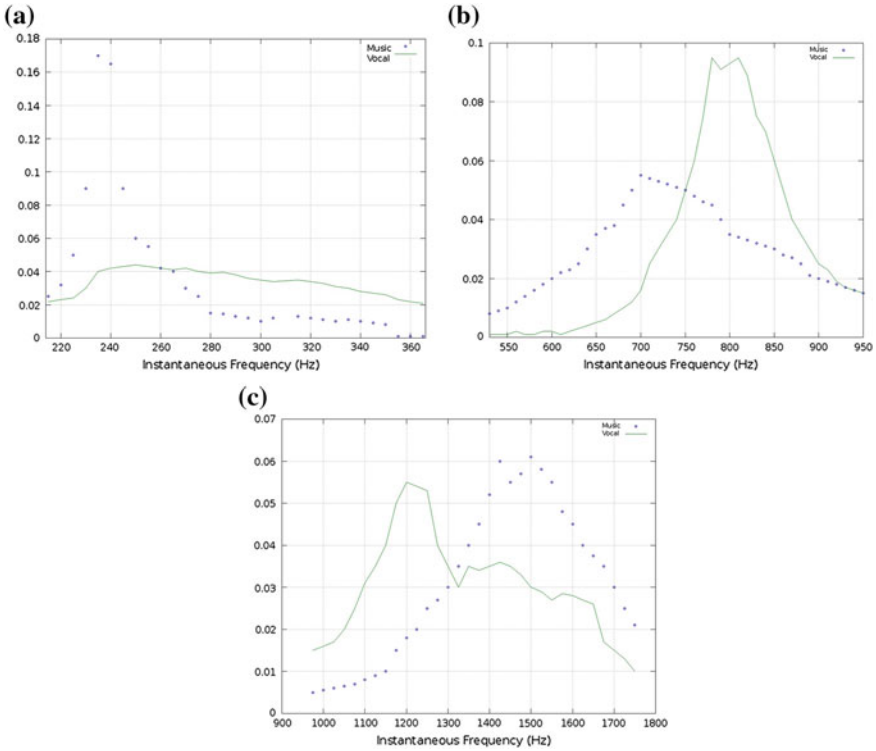


Fig. 2.3 Comparison of distribution of instantaneous frequencies in three bands for voice and music. **a** Band 1 (center frequency $\omega_0 = 240$). **b** Band 2 (center frequency $\omega_0 = 738$). **c** Band 3 (center frequency $\omega_0 = 1361$)

are extracted from the audio signal. The instantaneous feature distribution for voice and music segments of the audio, for three different bands, namely band 1 (center frequency $\omega_0 = 240$), band 2 (center frequency $\omega_0 = 738$) and band 3 (center frequency $\omega_0 = 1361$) are shown in Fig. 2.3a, b and c respectively.

It can also be seen from the distribution plots (Fig. 2.3) that the instantaneous frequency has very distinct distributions for voice and music segments in all the three frequency bands. This observation is exploited to distinguish voice and music components very reliably.

Once the music and voice in the audio conversation have been separated, the next task is that of distinguishing the voice spoken by the agent and that spoken by the customer. This is generally called speaker segmentation.

2.3 Agent Customer Speech Segmentation

A typical audio conversation available for processing and hence analysis is a conversation in *natural language* involving both the voice agent and the customer. For the purpose of meaningful analytics, it is mandatory that we know *who spoke what*, meaning we be able to distinguish the part of the audio conversation where the agent spoke and that part of the conversation where the customer spoke.

There are several audio segmentation techniques proposed in the literature for audio broadcast and group meetings [15–17], however, there is not much work done for telephone conversations [18] because telephone conversational speech has a much faster *speaker change rate*, large variation in speaking style of the speaker, presence of nonspeech sounds, crosstalk, and babble. Speaker segmentation is generally a multistep process consisting of

1. identifying the speaker change points in the audio conversation, followed by
2. identifying the number of speakers and then
3. associating each speech segment to a particular user.

Change point detection is, in general, the process of identifying change in some characteristics of the speech [19]. In a call center telephone conversation, the change in characteristics can be assumed to be that corresponding to the change from one person speaking to another. Typically, in telephone conversations, in addition to the noise and cross talk which is all too common, there are several instances when more than one person is speaking at the same time (talk over). Additionally, the conversation is skewed in the sense that the length or duration of one person speaking could be very small while that of the other person could be very large (a complaining customer). These are some of the challenges that one faces in telephone conversation compared to news audio broadcast where it is more controlled in terms of duration between changes in speakers.

A common process adopted for automatic change point detection in a telephone conversation is the use of a baseline change point detection technique [20]. The idea is to divide the entire speech into small overlapping windows of size 2–5 s; preprocess each window to extract speech features like MFCC (Mel Frequency Cepstral Coefficient) and LSF (Line Spectral Frequency) [21] which are commonly used speech features for speaker segmentation [20] and then compare adjacent windows to identify change points in audio.

Though MFCC is extensively used in speech recognition; LSF speech parameters perform much better for speaker segmentation [22–24].

The comparison helps decide if the two adjacent windows of speech have similar characteristics or different, meaning if they originate from the same source or different sources. As seen in Fig. 2.4 the adjacent overlapping windows are represented by a red box and a blue box and are compared to determine if these adjacent windows of speech have similar characteristics or different. This process is applied on the entire speech signal by sliding along the time axis.

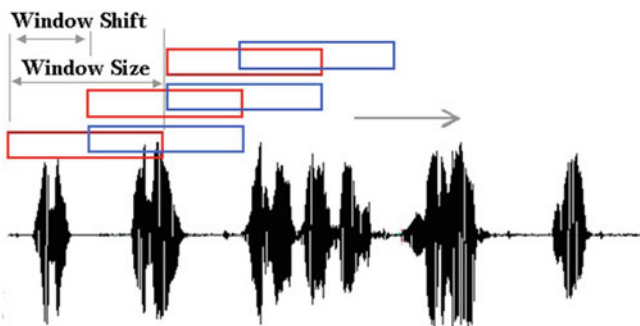


Fig. 2.4 Adjacent window comparison using overlapping windows

A speaker change point detection approach is described in detail for telephone conversation in [19]. It consists of a preprocessing step followed by identification of potential speaker change points followed by the selection of actual change points.

2.3.1 Two Pass Speaker Change Detection

The baseline change point detection technique used in speaker segmentation systems [25] is to divide the entire speech into small overlapping windows of size 25 s, then compare adjacent windows and decide whether the two adjacent windows of speech have similar characteristics or different, meaning if they originate from the same source or different sources. As shown using Fig. 2.4, the adjacent overlapping windows (represented by a red box and a blue box) are compared to determine if these windows of speech have similar characteristics or different. This decision is done by sliding along the time axis.

For a pair of adjacent windows being compared and found to be originating from different sources, the endpoint of the first window would be considered the change point. Clearly, the window size constrains the detection of short duration changes. Also, the localization (closeness of a detected change point to the actual change point) of the detected changes directly depends on the size of window overlap and the window shift [26].

2.3.1.1 First Pass: Coarse Detection

One can safely assume that the audio conversation, especially the call center conversation, starts with one speaker speaking for at least a few seconds (say *Speaker 1*, welcome message by the agent). These data are used to build a reference model λ_1 for *Speaker 1*. The rest of the audio conversation is analyzed using a sliding window (W) of length 2 s.

A hypothetical segment boundary is assumed at the center of the window with the first part (say W_1) of the window being assumed as a continuation of a speaker and the second part (say W_2) of the window being assumed as generated from the other speaker. A speech feature is extracted for the two sub-windows W_1 and W_2 under consideration using speech frames of length 20 ms with a frame overlap of 10 ms. Let there be N frames in each of the two sub-windows W_1 and W_2 . We have,

$$W_1 = \{w_{11}, w_{12}, \dots, w_{1N}\}$$

and

$$W_2 = \{w_{21}, w_{22}, \dots, w_{2N}\}$$

The task is to decide if these two sub-windows belong to the same or different acoustic conditions and hence speakers. Assume that the hypothesis $\mathcal{H}_0$ indicates that the two sub-windows belong to one single multivariate Gaussian process or a single speaker, namely

$$\mathcal{H}_0 : W_1, W_2 \sim N(\mu_W, \Sigma_W) = N_W$$

The hypothesis $\mathcal{H}_1$ indicates that the two segments (W_1, W_2) are generated by two different multivariate Gaussian processes or two speakers, namely,

$$\mathcal{H}_1 : W_1 \sim N(\mu_{W_1}, \Sigma_{W_1}) = N_{W_1} \quad \text{and}$$

$$W_2 \sim N(\mu_{W_2}, \Sigma_{W_2}) = N_{W_2}$$

where μ_W, μ_{W_1} and μ_{W_2} are the mean vectors and Σ_W, Σ_{W_1} , and Σ_{W_2} are the covariance matrices of the entire window W and the two sub-windows W_1, W_2 respectively. The generalized log likelihood ratio (GLR, $\mathcal{R}_W$) between the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ for the window W is defined as

$$\mathcal{R}_W = \log L(W, N_W) - (\log L(W_1, N_{W_1}) + \log L(W_2, N_{W_2})) \quad (2.13)$$

The GLR ($\mathcal{R}_W$) is computed [27] for a pair of adjacent sub-windows of same size and the analysis window is then shifted by a step length of 0.5 s along the speech signal and the likelihood ratio for the new window is computed. Negative $\mathcal{R}_W$ indicates that the sub-windows are better represented by N_{W_1} and N_{W_2} rather than the whole window W being represented by N_W meaning W had a speaker change point. The GLR distances thus computed for all windows for the audio track are computed and a threshold T (chosen empirically through experiments) is used to detect all possible candidate speaker change points. The threshold is so chosen, so that one does not miss out on any actual change points.

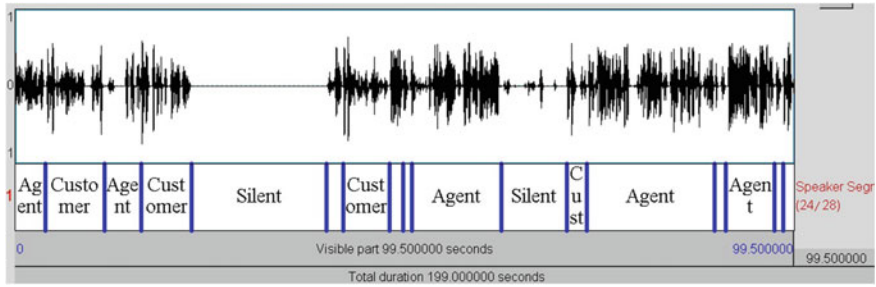


Fig. 2.5 Audio conversation with speaker segmentation

2.3.1.2 Second Pass: Speaker Change Detection

A second-level analysis is carried out to narrow down on the most likely speaker change point identified in the first pass. The candidate speaker change points detected in the above step are considered sequentially and we try to find the pair of segments that have a high likelihood of being from different speakers. For this, we use the initial part of the data which is assumed to be from *Speaker 1* having a model parameter set represented as, λ_1 . The similarity of the sub-window $W_1 = [w_1, w_2, \dots, w_N]$ in the neighborhood of a candidate speaker change point detected in the first pass is computed as the log-likelihood

$$S(W_1|\lambda_1) = \sum_{i=1}^N \log(p(w_i|\lambda_1)) \quad (2.14)$$

The acoustic probability that an observed feature vector w_i was generated by the model, λ_1 is given by,

$$p(w_i|\lambda_1) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{-\frac{1}{2}((w_i - \mu)^T \Sigma^{-1} (w_i - \mu))\right\} \quad (2.15)$$

where d is the dimension of the feature vector w_i . Since true densities of the *Speaker 1* is unknown, they can be approximated by sample mean and variances for computing the speaker model λ_1 . The change points for which the adjacent sub-window segments have a large difference in similarity computed from (2.14) are identified as valid speaker change points. The first valid speaker change point signals the beginning of a second speaker segment. The first speaker model parameters are modified using all the frames up to the detected change point.

Figure 2.5 shows a sample speech conversation after speaker segmentation. Once this is done, a speech to text conversion (Fig. 2.6a) would result in a corresponding text transcription as shown in Fig. 2.6b. However, there are challenges in the process of converting speech to text.

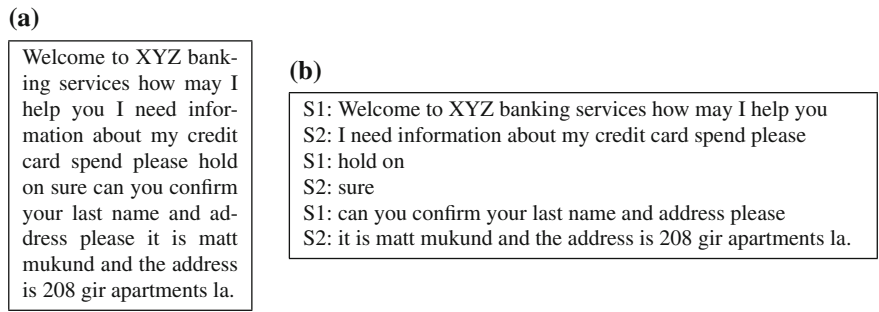


Fig. 2.6 Sample transcription. a Without and b with speaker segmentation

2.4 Speech to Text Conversion

The process of speech to text conversion is commonly know as automatic speech recognition (ASR). It should noted that like most AI systems, the process of building a speech recognition system is based on learning, namely the recognition system needs to be taught what it is required to perform. There are essentially (see Fig. 2.7) three main blocks involved in this recognition process. The first one is the acoustic models (AM), the second is the lexicon and the third is the statistical language model (SLM).

The acoustic model (AM) is an important component of speech to text conversion process, it models various sounds (phonemes) that make up a spoken word. In general, AMs are statistical models and is trained using large amount of speech corpus.

Speech corpus is a carefully constructed data set which consist of the spoken acoustic data plus a manually transcribed text corresponding to the audio data.

The lexicon (or the pronunciation dictionary, see Fig. 2.8) contains a list of words that should be recognized by the ASR engine. There are different approaches to construct lexicons using grapheme to phoneme mapping techniques [28, 29], however,

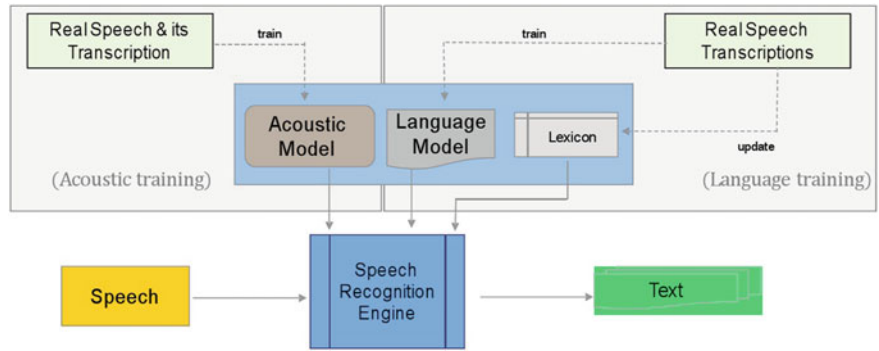


Fig. 2.7 Automatic speech recognition process

Fig. 2.8 A sample Lexicon that assists ASR

Word	Phonemic Sequence
ALLIANZ	AE L IY AH N Z
INSURANCE	IH N SH UH R AH N S
POLICY	P AA L AH S IY
SURRENDER	S ER EH N D ER
SECURITY	S IH K Y UH R AH T IY
SUPERVISOR	S UW P ER V AY Z ER
TAX	T AE K S
TRANSACTION	T R AE N Z AE K SH AH N

Fig. 2.9 A sample text corpus used to build SLM

THANKS FOR CALLING THIS IS MIKE CAN
WE START WITH A POLICY NUMBER POLICY
NUMBER OOH LET ME FIND THAT REAL QUICK
OK POLICY NUMBER EIGHT FIVE ZERO FIVE
TWO ONE FOUR AND THE CUSTOMER IT'S
BEVERLY B ROCKWOOD CAN YOU TRANSFER
TO THE CUSTOMER PLEASE YA GRANNY YOU
GET ON THE PHONE HELLO HEY HOW YOU
DOING MAAM YOUR NAME PLEASE MY NAME
IS BEVERLY B ROCKWOOD THANK YOU BEVERLY
CAN YOU VERIFY THE ADDRESS WE HAVE N
FILE FOR YOU PLEASE FOURTEEN TWENTY
FOUR EAST VIRGINIA AVENUE AND THE LAST
IS THE POST OFFICE BOX NUMBER IT'S THE
POST OFFICE BOX NUMBER OH OK MY HUSBAND
DID THAT WAY HAT'S THE REASON WHY I AM
KIND OF LANKY AN YOUR ADDRESS WITH THE
POST OFFICE BOX AGAIN

A lexicon, in general is handcrafted by a Linguistic and takes into account the actual manner in which a particular word is Pronounced. It should be noted that the same word can have more than one pronunciation.

Statistical Language Model or just the language model (LM) models the syntax and grammar of the sentences that are expected to be spoken and hence to be recognized. Generally, this is constructed through a process of training which requires significant amount of text corpus (see Fig. 2.9). Both SLM and the lexicon need to be fine-tuned specifically to suit the domain for which speech to text recognition is to be applied. For example, specific product names need to be appended to the standard lexicon of the ASR engine, so that the ASR engine can recognize the occurrence of the product name in the audio call. In a way the ASR engine performance largely depends on the language model and lexicon (see Fig. 2.8).

Speech to text conversion task is highly data driven, meaning the larger the volume of data used for training, the better is the performance. in terms of lower word error rate (WER). For example, a study [30] indicates that between 6,00,000 and 8,00,000 h of data acoustic training data would be required for WER → 0. More realistically, as discussed in [31] WER as low as 17 % could be achieved on voice search queries

Table 2.1 Types of error in ASR

Actual	A	B	C	D	*
Hypothesis	*	B	S	D	E

in English by using SLM trained on 230 billion words. However, it should be noted that apart from the amount of training data, the type of training data matters a lot in terms of the training data resembling the actual test scenario. Resemblance in terms of environmental conditions.

The performance of an Speech to text conversion process is generally measured as Word Error Rate (WER),

$$WER = \frac{\text{\# of word errors}}{\text{\# of reference words}}$$

where \# of word errors is the sum of number of word deletions, word insertions and word substitutions required to match the output of a Speech to text hypothesis to the actual speech utterance.

We see above (in Table 2.1) that there are three types of word errors. The word A is not recognized at all (Deletion error), C is recognized as S (Substitution error), E is falsely recognized when it is not actually present in the speech (Insertion error). For the above example, $WER = 3/4$. $WER \rightarrow 0$ indicates better accuracy WER.

The current adopted method in most commercial voice analytics tools (for example, [2–5]) for analyzing call center conversations, as mentioned earlier, is one of converting speech to text followed by natural language text processing. However, there are several challenges in using this two-step approach to analyze call center conversations.

2.4.1 Speech-to-Text Conversion Challenges

There are a significant number of challenges that crop up during the process of converting speech to text, especially for natural language spoken conversations on the telephone channel.

As observed earlier, building a speech recognition system to automatically transcribe naturally spoken call center conversation requires extensive training, not only in terms of construction of a SLM and a lexicon but also in terms of training acoustic models.

A language is said to be resource rich if it has a reasonable quality and quantity of speech corpus. Else the language is categorized as being Deficient.

Unfortunately, the state of the art, even for a resource rich language like English is poor because of use of slangs, different accents, different environmental conditions to name a few. Post extensive training, a well-trained speech recognition engine even for a resource rich language like English, the speech recognition performance on a natural language spoken conversation has a WER of about 40 – 50 % (see Appendix F).

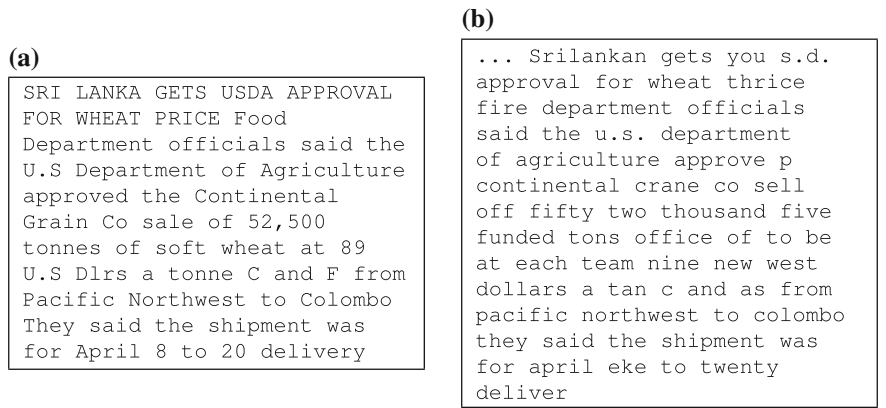


Fig. 2.10 Example of noise due to speech to text process. **a** Actual text and **b** ASR transcribed text

Table 2.2 Factors impeding
Speech to text

Category	Parameters
Source/Channel	Background noise, speech codec, microphone, telephone
Linguistic	language, dialect, mixed language, accent, pronunciation, domain, vocabulary size, proper-names, OOV
Non-linguistic	Spoken speech (anger, hesitation, fast)
Operational	real time, offline, multi-pass ASR

Subsequently, irrespective of the type of text analysis approach adopted, the analysis of the text is affected by the noise in the transcription. This affects the usable information derived from these noisy transcriptions. There is a relatively new area of work concentrating on noisy text analysis [33–35].

ASR performance even under controlled conditions, like in [32], where an original article was read by a single user, the speech-to-text conversion accuracy is poor. Figure 2.10a shows the original text, while 2.10b shows the output of an ASR. Clearly, the ability to automatically transcribe a natural language speech is very poor. As noted earlier, the speech-to-text conversion mechanism is based on the process of learning. If the training data, in this case speech, is for a particular channel or a particular environment, then the speech to text processing fails if there is a variation in these environmental conditions. Hence, the presence of environmental noise in the conversational speech is a further impediment in the speech to text conversion process [36] and so is the variation in the channel characteristics. A more complete list is captured in Table 2.2.

The speech-to-text conversion process for resource deficient languages is further compounded because of lack of availability of speech corpus for training (acoustic models, statistical language models, lexicon, see Fig. 2.7) and hence the recognition accuracies are very poor. However, speech corpus is a central element for training

the acoustic models used in a speech recognition engine. Additionally, constructing a speech corpus for a language is an expensive, time-consuming, and laborious process [37]. Appendix E details on how to construct a speech corpus for a resource deficit language.

Building Speech recognition application for resource deficient languages is a challenge because of the unavailability of a speech corpus.

A mechanism to build an inexpensive speech corpus, for resource- deficient languages Indian English and Hindi, by exploiting existing collections of online speech data to build a Frugal speech corpus is proposed in [37].

For resource-deficient languages the two-step process (steps of speech to text followed text analysis) of analyzing call center conversation to derive analytics does not exist. However, there has been a segment of work that makes use of existing language resources to build speech recognition engine for a resource deficient language (see for example [37–41]). Additionally, the speech to text conversion process is further complicated in a multilingual country, like India, where people tend to use more than one language in the same sentence. This is called mixed language use [42] and is also known as code switching in literature. Speech-to-text conversion of such speech is highly poor because most often the speech recognition engines are designed with the assumption that users stick to one language during their conversation. This poses problems in automatic recognition of linguistic content of the conversation.

These challenges in linguistic processing of spoken speech make it difficult to use the two-step process to analyze call conversations, thus opening up options to use non-linguistic processing of speech to derive usable information. For resource deficient languages one needs to necessarily rely on techniques that do not require the linguistic speech to text conversion process for such languages.

Additionally, reasons to adopt non-linguistic processing could stem from the fact that sometimes it is just sufficient to know in a call center setting which call was abnormal; without the need to know what was the explicit (linguistic) reason for abnormality.

References

1. M. Gavalda, J. Schlueter, The truth is out there: Using advanced speech analytics to learn why customers call help-line desks and how effectively they are being served by the call center agent, in *Advances in Speech Recognition*, ed. by A. Neustein (Springer, USA, 2010), pp. 221–243
2. Nexidia. http://www.nexidia.com/solutions/contact_center/product_suite
3. Utopy. <http://www.utopy.com/index.php?page=intelligent-script-adherence>
4. Verint. http://verint.com/contact_center/section2a.cfm?article_level2_category_id=21&article_level2a_id=343
5. Nice. <http://www.nice.com/smartcenter-suite/interaction-analytics>
6. I. Ahmed, S.K. Kopparapu, Interactive voice response mashup system for service enhancement. *Recent Patents on Telecommun.* **1**, 100–108 (2012)

7. S.K. Kopparapu, M.A. Pandharipande, and G. Sita. Music and vocal separation using multi-band modulation based features. In: IEEE Symposium on Industrial Electronics Applications (ISIEA), pp. 733–737 (2010)
8. P. Maraso, J.F. Kaiser, T.F. Quatieri, Energy separation in signal modulations with applications to speech analysis. IEEE Trans. Signal Proc. **41**, 3024–3051 (1993)
9. D. Dimitriadis, P. Maragos, A. Potamianos, Robust am-fm features for speech recognition. IEEE Signal Process. Lett. **12**, 621–624 (2005)
10. T.F. Quatieri, T.E. Hanna, G.C. O-Leary, Am-fm separation using auditory-motivated filters. IEEE Trans. Speech and Audio Proc. **5**, 465–480 (1997)
11. H.L. John, H.G. Zhou, J.F. Kaiser, Nonlinear feature based classification of speech under stress. IEEE Trans. Signal Proc. **9**, 203 (2001)
12. D. Dimitriadis, P. Maragos, Continuous energy demodulation methods and application to speech analysis. Speech Communication **48**, 819–837 (2006)
13. L.G.C. John, J.H.L. Hansen, J.F. Kaiser, Vocal fold pathology assessment using am auto-correlation analysis of the teager energy operator. In in Fourth Int. Conf. Spoken. Language **2**, 757–760 (1996)
14. R. Mandar and H. John. Frequency distribution based weighted sub-band approach for classification of emotional/stressful content in speech. In:EUROSPEECH-2003, pp. 721–724 (2003)
15. G. Rashmi, N. Balakrishnan, A novel method for two-speaker segmentation. In: INTERSPEECH-2004, pp. 2337–2340 (2004)
16. A.N. Iyer, U.O. Ofoegbu, R.E. Yantorno, S. Wenndt. Speaker modeling in conversational speech with application to speaker count. In Proceedings of ICSLP (2006)
17. S. Soni, I. Ahmed, S.K. Kopparapu. Automatic segmentation of broadcast news audio using self similarity matrix. CoRR, abs/1403.6901 (2014)
18. S.K. Kopparapu, A. Imran, G. Sita. A two pass algorithm for speaker change detection. In TENCON 2010–2010 IEEE Region 10 Conference, pp. 755–758 (2010)
19. I. Ahmed ,S. Kopparapu. Speaker change detection in telephone speech. In International Conference on Signals, Systems and Automation, Vallabh Vidyanagar, India (2009)
20. S.E. Tranter, D.A. Reynolds, An overview of automatic speaker diarization systems. IEEE Trans. Audio, Speech Lang. Process. **14**, 1557–1565 (2006)
21. Y. Tianren, X. Juanjuan, Lu Wei. The computation of line spectral frequency using the second chebyshev polynomials. In 6th International Conference on Signal Processing, pp. 190–192, vol 1 (2002)
22. P. Delacourt, C.J. Wellekens, Distbic: A speaker-based segmentation for audio data indexing. Speech Communication **32**, 111–126 (2000)
23. Lie Lu ,H.-J. Zhang, Real-time unsupervised speaker change detection. In Proceedings to 16th International Conference on Pattern Recognition, pp. 358–361 (2002)
24. A.G. Adami, S.S. Kajarekar, H. Hermansky. A new speaker change detection method for two-speaker segmentation. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing -ICASSP 02, pp. IV-3908 - IV-3911 (2002)
25. S.E. Tranter, D.A. Reynolds, An overview of automatic speaker diarization systems. IEEE Trans. Audio Speech Lang. Process. **14**, 1557–1565 (2006)
26. I. Ahmed, S.K. Kopparapu, Speaker change detection in telephone speech. In International Conference on Signals, Systems and Automation, Vallabh Vidyanagar, India (2009)
27. B. Narayanaswamy, G. Rashmi, R. Stern, *Voting for two speaker segmentation* (In Proc, ICSLP, 2006)
28. Maximilian Bisani, Hermann Ney, Joint-sequence models for grapheme-to-phoneme conversion. Speech Commun. **50**(5), 434–451 (2008)
29. M. Laxminarayana, S. Kopparapu, Semi-automatic generation of pronunciation dictionary for proper names: an optimization approach. Proceedings of the 6th International Conference on Natural Language Processing (ICON08), pp. 118–126 (2008)
30. K. M. Roger, A comparison of the data requirements of automatic speech recognition systems and human listeners. EUROSPEECH'03, pp. 2582–2584 (2003)

31. C. Ciprian, B. Dan, S. Maria, N. Patrick, K. Shankar, Large scale language modeling in automatic speech recognition. Technical report, Google Techreport (2012)
32. Reuters Transcribed Subset. http://kdd.ics.uci.edu/databases/reuters_transcribed/reuters_transcribed.html. Viewed July 08, 2013
33. J. Mamou, D. Carmel, R. Hoory, Spoken document retrieval from call-center conversations. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06, pp. 51–58, New York (2006)
34. D. Lopresti, S. Roy, K. Schulz, L.V. Subramaniam, Special issue on noisy text analytics. International J. Document Anal. Recognit. (IJ DAR) **14**(2), 111–112 (2011)
35. S. Agarwal, S. Godbole, D. Punjani, S. Roy, How much noise is too much: A study in automatic text classification. In Seventh IEEE International Conference on Data Mining, ICDM 2007. pp. 3–12 (2007)
36. M. L. Narayana, S.K. Kopparapu, Effect of noise-in-speech on mfcc parameters. In Proceedings of the 9th WSEAS international conference on signal, speech and image processing, and 9th WSEAS international conference on Multimedia, internet and video technologies, SSIP '09/MIV'09, pp. 39–43, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS)
37. I. Ahmed, S.K. Kopparapu, Speech recognition for resource deficient languages using frugal speech corpus. In IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC), pp. 750–755 (2012)
38. O. Cetin, M. Plauche, U. Nallasamy, Unsupervised adaptive speech technology for limited resource languages: A case study for tamil. In: Proceedings of International Workshop on Spoken Languages Technologies for Under-resourced languages (SLTU), Hanoi, Vietnam (2008)
39. M. Plauche, O. Cetin, U. Nallasamy, How to build a spoken dialog system with limited or no language resources. In AI in ICT4D. ICFAI University Press, India (2008)
40. T. J. Arnar, W. D. Edward, I. Koji, S. Furui. Language model adaptation for resource deficient languages using translated data. In INTERSPEECH'05, pp. 1329–1332 (2005)
41. I. Dawa, Y. Sagisaka, S. Nakamura, Investigation of asr systems for resource-deficient languages. ACTA AUTOMATICA SINICA **1**, 1–8 (2008)
42. K. Bhuvanagiri, S.K. Kopparapu, Mixed Language Speech Recognition without Explicit Identification of Language. American J. Signal Process. **2**, 92–97 (2012)

Non-Linguistic Analysis of Call Center Conversations

Kopparapu, S.K.

2015, XII, 83 p. 47 illus., 26 illus. in color., Softcover

ISBN: 978-3-319-00896-7