37. Text-Dependent Speaker Recognition

M. Hébert

Text-dependent speaker recognition characterizes a speaker recognition task, such as verification or identification, in which the set of words (or lexicon) used during the testing phase is a subset of the ones present during the enrollment phase. The restricted lexicon enables very short enrollment (or registration) and testing sessions to deliver an accurate solution but, at the same time, represents scientific and technical challenges. Because of the short enrollment and testing sessions, text-dependent speaker recognition technology is particularly well suited for deployment in large-scale commercial applications. These are the bases for presenting an overview of the state of the art in text-dependent speaker recognition as well as emerging research avenues. In this chapter, we will demonstrate the intrinsic dependence that the lexical content of the password phrase has on the accuracy. Several research results will be presented and analyzed to show key techniques used in text-dependent speaker recognition systems from different sites. Among these, we mention multichannel speaker model synthesis and continuous adaptation of speaker models with threshold tracking. Since text-dependent speaker recognition is the most widely used voice biometric in commercial deployments, several results drawn from realistic deployment scenarios are also included.

37.1 Brief Overview

There exists significant overlap and fundamental differences between text-dependent and text-independent speaker recognition. The underlying technology and algorithms are very often similar. Advances in one field, frequently text-independent speaker recognition because of the NIST evaluations [37.1], can be applied with success in the other field with only minor modifications. The main difference, as pointed out by the nomenclature, is the lexicon allowed by each. Although not restricted to a specific lexicon for enrollment, text-dependent speaker recognition assumes that the lexicon active during the testing is a subset of the enrollment lexicon. This limitation does not exist for text-independent speaker recognition where any word can be uttered during enrollment and testing. The known overlap between the enrollment and testing phase results in very good accuracy with a limited amount of enrollment material (typically less than 8 s of speech). In the case of unknown-text speaker recognition, much more enrollment material is required (typically more than 30 s) to achieve similar accuracy. The theme of *lexical content* of the enrollment and testing sessions is central to text-dependent speaker recognition and will be recurrent during this chapter.

Traditionally, text-independent speaker recognition was associated with speaker recognition on entire conversations. Lately, work from *Sturim* et al. [37.2] and others [37.3] has helped bridge the gap between text-dependent and text-independent speaker recognition by using the most frequent words in conversational speech and applying text-dependent speaker recognition techniques to these. They have shown the benefits of using text-dependent speaker recognition techniques on a text-independent speaker recognition task.

Table 37.1 illustrates the challenges encountered in text-dependent speaker recognition (adapted from [37.4]). It can be seen that the two main sources of degradation in the accuracy are channel and lexical mismatch. Channel mismatch is present in both text-dependent and text-independent speaker recognition, but mismatch in the lexical content of the enrollment and testing sessions is central to text-dependent speaker recognition.

Throughout this chapter, we will try to quantify accuracy based on application data (from trial data collections, comparative studies or live data). We will favor live data because of its richness and relevance. Special care will be taken to reference accuracy on publicly available data sources (some may be available for a fee), but in some other cases an explicit reference is impossible to preserve contractual agreements. Note that a comparative study of off-the-shelf commercial text-dependent speaker verification systems was presented at Odyssey 2006 [37.5].

This chapter is organized as follows. The rest of this section explains at a high-level the main components of a speaker recognition system with an emphasis on particularities of text-dependent speaker recognition. The reader is strongly encouraged, for the sake of completeness, to refer to the other chapters on speaker recognition. Section 37.2 presents the main technical and commercial deployment challenges. Section 37.3 is formed by a collection of selected results to illustrate the challenges of Sect. 37.2. Concluding remarks are found in Sect. 37.4.

37.1.1 Features

The first text-dependent speaker recognition system descriptions that incorporate the main features of the current state of the art date back to the early 1990s. In [37.6] and [37.7], systems have feature extraction, speaker models and score normalization using a likelihood ratio scheme. Since then, several groups have explored different avenues. The work cited below is not restricted to the text-dependent speaker recognition field, nor is it intended as an exhaustive list. Feature sets usually come in two flavors: MEL [37.8] or LPC (linear predictive coding) [37.6, 9] cepstra. Cepstral mean subtraction and feature warping have proved effective on cellular data [37.10] and are generally accepted as an effective noise robustness technique. The positive role of dynamic features in text-dependent speaker recognition has recently been reported in [37.11]. Finally, a feature mapping approach [37.12] has been proposed as an equivalent to speaker model synthesis [37.13]; this is an effective channel robustness technique.

Table 37.1 Effect of different mismatch types on the EER for a text-dependent speaker verification task (after [37.4]). The corpus is from a pilot with 120 participants (gender balanced) using a variety of handsets. Signal-to-noise ratio (SNR) mismatch is calculated using the difference between the SNR during enrollment and testing (verification). For the purposes of this table, an absolute value of this difference of more than 10 db was considered mismatched. Channel mismatch is encountered when the enrollment and testing sessions are not on the same channel. Finally, lexical mismatch is introduced when the lexicon used during the testing session is different from the enrollment lexicon. In this case, the password phrase was always a three-digit string. LD0 stands for a lexical match such that the enrolment and testing were performed on the same digit string. In LD2, only two digits are common between the enrollment and testing; in LD4 there is only one common digit. For LD6 (complete lexical mismatch), the enrollment lexicon is disjoint from the testing lexicon. Note that, when considering a given type of mismatch, the conditions are matched for the other types. At EERs around 8%, the 90% confidence interval on the measures is 0.8%

Type of mismatch	Accuracy (EER) (%)
No mismatch	7.02
SNR mismatch	7.47
Channel mismatch	9.76
Lexical mismatch (LD2)	8.23
Lexical mismatch (LD4)	13.4
Complete lexical mismatch (LD6)	36.3

37.1.2 Acoustic Modeling

Several modeling techniques and their associated scoring schemes have been investigated over the years. By far the most common modeling scheme across

speaker recognition systems is the hidden Markov model (HMM) [37.14]. The unit modeled by the HMM depends heavily on the type of application (Fig. 37.1). In an application where the enrollment and testing lexicon are identical and in the same order (*My voice is my password* as an example), a sentence-level HMM can be used. When the order in which the lexicon appears in the testing phase is not the same as the enrollment order, a word-level unit is used [37.9, 15]. The canonical application of word-level HMMs is present in digit-based speaker recognition dialogs. In these, all digits are collected during the enrollment phase and a random digit sequence is requested during the testing phase. Finally, phone-level HMMs have been proposed to refine the representation of the acoustic space [37.16–18]. The choice of HMMs in the context of text-dependent speaker recognition is motivated by the inclusion of inherent time constraints.

The topology of the HMM also depends on the type of application. In the above, standard left-to-right N-state HMM have been used. More recently, single-state HMMs [also called Gaussian mixture models (GMMs)] have been proposed to model phoneme-level acoustics in the context of text-dependent speaker recognition [37.19] and later applied to text-independent speaker recognition [37.20]. In this case, the temporal information represented by the sequence of phonemes is dictated by an external source (a speech recognition system) and not inscribed in the model's topology. Note that GMMs have been extensively studied, and proved very effective, in the context of text-independent speaker recognition.

In addition to the mainstream HMMs and GMMs, there exists several other modeling methods. Support vector machine (SVM) classifiers have been suggested for speaker recognition by *Schmidt* and *Gish* [37.21] and have become increasingly used in the text-independent speaker recognition field [37.22, 23]. To our knowledge, apart from [37.24, 25], there has been no thorough study of an SVM-based system on a text-dependent speaker recognition task. In this context, the key question is to assess the robustness of an SVM-based system to a restricted lexicon. Dynamic time warping (DTW) algorithms have also been investigated as the basis for text-dependent speaker recognition [37.26, 27]. Finally, neural networks (NNs) modeling methods also form the basis for text-dependent speaker recognition algorithms [37.28, 29]. Since the bulk of the literature and advances on speaker recognition are based on algorithms that are built on top of HMM or GMM, we will focus for the rest of this chapter on those. We believe, however, that the main conclusions and results herein apply largely to the entire field of text-dependent speaker recognition.

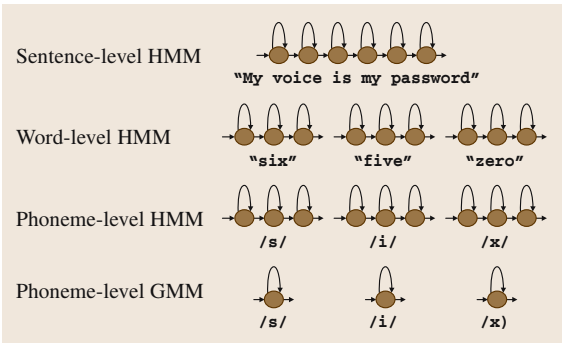

Fig. 37.1 Hidden Markov model (HMM) topologies

37.1.3 Likelihood Ratio Score

As mentioned in a previous section, speaker recognition can be split into speaker identification and verification. In the case of speaker identification, the score is simply the likelihood, template score (in the case of DTW), or posterior probability in the case of an NN. For speaker verification, the standard scoring scheme is based on the competition between two hypothesis [37.30].

- H_0: the test utterance is from the claimed speaker C, modeled by λ;
- H_1: the test utterance is from a speaker other than the claimed speaker C, modeled by $\overline{\lambda}$.

Mathematically, the likelihood ratio $[L(\boldsymbol{X}|\lambda)]$ detector score is expressed as

$$L(\boldsymbol{X}|\lambda) = \log p(\boldsymbol{X}|\lambda) - \log p(\boldsymbol{X}|\overline{\lambda}) \,, \tag{37.1}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mathrm{T}\}$ is the set of feature vectors extracted from the utterance and $p(\boldsymbol{X}|\lambda)$ is the likelihood of observing $\boldsymbol{X}$ given model λ. H_0 is represented by a model λ of the claimed speaker C. As mentioned above, λ can be an HMM or a GMM that has been trained using features extracted from the utterances from the claimed speaker C during the enrollment phase. The representation of H_1 is much more subtle because it should, according to the above, model all potential speakers other than C. This is not tractable in a real system. Two main approaches have been studied to model $\overline{\lambda}$. The first consists of selecting N background or cohort speakers, to model individually $(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ and to combine their likelihood score on the test utterance.

The other approach uses speech from a pool of speakers to train a single model, called a general, background or universal background model (UBM). A variant of the UBM, widely used for its channel robustness, is to train a set of models by selecting utterances based on some criteria such as gender, channel type, or microphone type [37.8]. This technique is similar in spirit to the one presented in [37.12]. Note that for the case of text-dependent speaker recognition, it is beneficial to train a UBM with data that lexically match the target application [37.19].

37.1.4 Speaker Model Training

In order to present a conceptual understanding of the text-dependent speaker recognition field, unless otherwise stated, we will assume only two types of underlying modeling: a single GMM for all acoustic events, which is similar to the standard modeling found in text-independent tasks. We will call this the *single-GMM* approach. The other modeling considered is represented as *phoneme-level GMM* on Fig. 37.1. This approach will be called phonetic-class-based verification (PCBV as per [37.19]). This choice is motivated by simplicity, availability of published results, as well as current trends to merge known and text-independent speaker recognition (Sect. 37.1).

For these types of modeling, training of the speaker model is performed using a form of Bayesian adaptation [37.30, 31], which alters the parameters of $\overline{\lambda}$ using the features extracted from the speech collected during the enrollment phase. As will be shown later, this form of training for the speaker models is well suited to allow adaptation coefficients that are different for means, variances, and mixture weights. This, in turn has an impact on the accuracy in the context of text-dependent speaker recognition.

37.1.5 Score Normalization and Fusion

Although the score coming from the likelihood ratio detector (37.1) discriminates genuine speakers from imposters well, it remains fragile. Several score normalization techniques have been proposed to improve robustness. We will discuss a few of those.

The first approach is called the H-norm, which stands for handset normalization, and is aimed at normalizing handset variability [37.32], especially cross-channel variability. A similar technique called the Z-norm has also been investigated in the context of text-dependent speaker recognition with adaptation [37.33]. Using a set ($\simeq 200$) of impostor test utterances with known handset and/or gender labels, a newly trained speaker model is challenged. The scores calculated using (37.1) are fitted using a Gaussian distribution to estimate their mean $[\mu_{\mathrm{H}}(\lambda)]$ and standard deviations $[\sigma_{\mathrm{H}}(\lambda)]$ for each label H. At test time, a handset and/or gender labeler [37.8,32] is used to identify the label H of the testing utterance. The normalized score is then

$$L_{\text{H-norm}}(\boldsymbol{X}|\lambda, H) = \frac{L(\boldsymbol{X}|\lambda) - \mu_{\mathrm{H}}(\lambda)}{\sigma_{\mathrm{H}}(\lambda)} \,. \tag{37.2}$$

This technique is computationally very efficient because $\mu_{\mathrm{H}}(\lambda)$ and $\sigma_{\mathrm{H}}(\lambda)$ are calculated once after the enrollment. It can, however, become inefficient when adaptation of the speaker model (Sect. 37.1.6) occurs because $\mu_{\mathrm{H}}(\lambda)$ and $\sigma_{\mathrm{H}}(\lambda)$ need to be recalculated.

Another score normalization technique widely used is called test normalization or T-norm [37.34]. This approach is applied to text-dependent speaker recognition in [37.35]. It can be viewed as the dual to H-norm in that, instead of challenging the target speaker model with a set imposter test utterances, a set of imposter speaker models (T) are challenged with the target test utterance. Assuming a Gaussian distribution of those scores, $\mu_{\mathrm{T}}(\boldsymbol{X})$ and $\sigma_{\mathrm{T}}(\boldsymbol{X})$ are calculated and applied using

$$L_{\text{T-Norm}}(\boldsymbol{X}|\lambda, T) = \frac{L(\boldsymbol{X}|\lambda) - \mu_{\mathrm{T}}(\boldsymbol{X})}{\sigma_{\mathrm{T}}(\boldsymbol{X})} \,. \tag{37.3}$$

By construction, this technique is computationally very expensive because the target test utterance has to be applied to the entire set of imposter speaker models.

Notwithstanding the computational cost, the H-norm and T-norm developed in the context of text-independent speaker recognition need to be adapted for the text-dependent speaker recognition. These techniques have been shown to be heavily dependent on the lexicon of the set of imposter utterances (H-norm [37.36]) and on the lexicon of the utterances used to train the imposter speaker models (T-norm [37.35]). The issue of lexical dependency or mismatch is not present in a text-independent speaker recognition task, but heavily influences text-dependent speaker recognition system designs [37.4]. We will come back to this question later (Sect. 37.3.2 and Sect. 37.3.4).

Finally, as in the text-independent speaker recognition systems, score fusion is present in text-dependent systems and related literature [37.29]. The goal of fusion is to combine classifiers that are assumed to make uncorrelated errors in order to build a better performing overall system.

37.1.6 Speaker Model Adaptation

Adaptation is the process of extending the enrollment session to the testing sessions. Common wisdom tells us that *the more speech you train with, the better the accuracy will be*. This has to be balanced with requirements from commercial deployments where a very long enrollment sessions is negatively received by end customers. A way to circumvent this is to fold back into the enrollment material any testing utterance that the system has a good confidence of having been spoken by the same person as the original speaker model. Several studies on unknown [37.37, 38] and text-dependent [37.39, 40] speaker recognition tasks have demonstrated the effectiveness of this technique. Speaker model adaptation comes in two flavors. Supervised adaptation, also known as retraining or manual adaptation, implies an external verification method to assess that the current speaker is genuine. That can be achieved using a secret piece of information or another biometric method. The second method is called unsupervised adaptation. In this case, the decision taken by the speaker recognition system (a verification system in this case) is used to decide on the application of adaptation of the speaker model with the current test utterance. Supervised adaptation outperforms its unsupervised counterpart in all studies. A way to understand this fact is to consider that unsupervised adaptation requires a good match between the target speaker model and the testing utterance to adapt the speaker model; hence this new utterances does not bring new variability representing the speaker, the transmission channel, the noise environment, etc. The supervised adaptation scheme, since it is not based on the current utterance, will bring these variabilities to the speaker model in a natural way. Under typical conditions, supervised adaptation can cut, on text-dependent speaker verification tasks, the error rates by a factor of five after 10–20 adaptation iterations.

37.2 Text-Dependent Challenges

The text-dependent speaker recognition field faces several challenges as it strives to become a mainstream biometric technique. We will segregate those into two categories: technological and deployment. The technology challenges are related to the core algorithms. Deployment challenges are faced when bringing the technology into an actual application, accepting live traffic. Several of these challenges will be touched on in subsequent sections where we discuss the current research landscape and a set of selected results (Sect. 37.3). This section is a superset of challenges found in a presentation by *Heck* at the Odyssey 2004 workshop [37.41]. Note that the points raised here can all give rise to new research avenues; some will in fact be discussed in following sections.

37.2.1 Technological Challenges

Limited Data and Constrained Lexicon

As mentioned in Sect. 37.1, text-dependent speaker recognition is characterized by short enrollment and testing session. Current commercial applications use enrollment sessions that typically consist of multiple repetitions (two or three) of the enrollment lexicon. The total speech collected is usually 4–8 s (utterances are longer than that, but silence is usually not taken into account). The testing session consists of a single (or sometimes two) repetitions of a subset of the enrollment lexicon, for a total speech input of 2–3 s. These requirements are driven by usability studies which show that shorter enrollment and testing sessions are best perceived by end customers.

The restricted nature of the lexicon (hence text-dependent speaker recognition), is a byproduct of the short enrollment sessions. To achieve deployable accuracies under the short enrollment and testing constraints, the lexicon has to be restricted tremendously. Table 37.2 lists several examples of enrollment lexicon present is deployed applications. Table 37.3 describes typical testing strategies given the enrollment lexicon. In most cases, the testing lexicon is chosen to match the enrollment lexicon exactly. Note that, for random (and pseudo random) testing schemes, a *2-by-4* approach is sometimes used: in order to reduce the cognitive load, a four-digit string repeated twice is requested from

Table 37.2 Examples of enrolment lexicon

Abreviation	Description
E	Counting from 1 to 9: *one two three* ...
T	10-digit telephone number
S	9-digit account number
N	First and last names
MVIMP	*My voice is my password*

Table 37.3 Examples of testing lexicon. Note that the abbreviations refer to Table 37.2 and each line gives depicts a potential testing lexicon given the enrolment lexicon

Abreviation	Description
E	Counting from 1 to 9: *one two three ...*
R	Random digit sequence *2 6 8 5 2 6 8 5*
pR	Pseudorandom digit sequence from E *2 3 6 7 2 3 6 7*
T	Same 10-digit telephone number as enrolment
	Random digit sequence selected from enrolment lexicon
	Pseudorandom digit sequence selected from enrolment lexicon
S	Similar to T but for a nine-digit account number
N	First and last names
MVIMP	*My voice is my password*

the user. This makes for a longer verification utterance without increasing the cognitive load: a totally random eight-digit string could hardly be remembered by a user. Table 37.4 shows a summary of the accuracy in different scenarios. We reserve discussion of these results for Sect. 37.3.2.

Channel Usage

It is not rare to see end customers in live deployments using a variety of handset types: landline phones, pay phones, cordless phones, cell phones, etc. This raises the issue of their impact on accuracy of channel usage. A cross-channel attempt is defined as a testing session originating from a different channel than the one used during the enrollment session. It is not rare to see the proportion of cross-channel calls reach 25–50% of all genuine calls in certain applications. The effect on the accuracy is very important ranging from doubling the EER [37.4, 42] to quadrupling [37.42] the EER on some commercial deployments. This is a significant area where algorithms must be improved. We will come back to this later.

Aging of Speaker Models

It has been measured in some commercial trials [37.42] and in data collections [37.15] that the accuracy of a text-dependent speaker recognition system degrades slowly over time. In the case of [37.15], the error rate increased by 50% over a period two months. There exists several sources of speaker model aging, the main ones being

Table 37.4 Speaker verification results (EERs) for different lexicon. Refer to Tables 37.2 and 37.3 for explanations of the acronyms. Empty cells represent the fact that pseudorandom strings (pR) do not apply to S since the pseudorandom string is extracted from an E utterance. Italicized results depict conditions that are not strictly text-dependent speaker verification experiments. At EERs of 5–10%, the 90% confidence interval on the measures is 0.3–0.4%

Verify	E	S	R	pR	N
Enroll					
E	6.16%	10.2%	13.2%	10.4%	*36.2%*
S	11.6%	5.05%	14.2%		*39.3%*
R	10.7%	9.43%	11.5%	10.0%	*36.4%*
pR	10.0%		11.3%	8.05%	*35.6%*
N	*38.9%*	*39.7%*	*39.3%*	*39.1%*	10.6%

natural aging, channel usage, and behavioral changes. Natural aging is related to the physiological changes that occur to the phonatory apparatus over long periods of time. Channel usage changes over time can cause the speaker model to become outdated with respect to the current channel usage. Finally, behavioral changes occur when users get more exposure to the voice interface and thus alter the way in which they interact with it. As an example, first-time users of a speech application (usually the enrollment session) tend to cooperate with the system by speaking slowly under friendly conditions. As these users get more exposure to the application, they will alter the way that they interact with it and use it in adverse conditions (different channels, for example). All of these factors affect the speaker models and scoring, and thus are reflected in the accuracy. The common way to mitigate this effect is to use speaker model adaptation (Sect. 37.1.6 and Sect. 37.3.5).

37.2.2 Commercial Deployment Challenges

Dialog Design

One of the main pitfalls in deploying a speech-based security layer using text-dependent speaker recognition is poor dialog design choices. Unfortunately, these decisions are made very early in the life cycle of an application and have a great impact on the entire life of the application. Examples [37.41] are

1. small amount of speech collected during enrollment and/or verification
2. speech recognition difficulty of the claim of identity (such as a first and last names in a long list)

3. poor prompting and error recovery
4. lexicon mismatch between enrollment and verification

One of the challenges in deploying a system is certainly protection against recordings since the lexicon is very restricted. As an example, in the case where the enrollment and verification lexicon is *My voice is my password* or a telephone number, once a fraudster has gained access to a recording from the genuine speaker, the probability that they can gain access has greatly increased. This can be addressed by several techniques (one can also think about combining them). The first technique consists of explicitly asking for a randomized subset of the lexicon. This does not lengthen the enrollment session and is best carried out if the enrollment lexicon consists of digits. The second is to perform the verification process across the entire dialog even if the lexical mismatch will be high (Sect. 37.3.2 and Sect. 37.3.6), while maintaining a short enrollment session. A third technique is to keep a database of *trusted* telephone numbers for each user (home, mobile, and work) and to use this external source of knowledge to improve security and ease of use [37.43]. Finally, a challenge by a secret *knowledge* question drawn from a set of questions can also be considered. These usually require extra steps during the enrollment session. It is illusory to think that a perfect system (no errors) can be designed, the goal is simply to raise the bar of

1. the amount of information needed and
2. the sophistication required by a fraudster to gain access.

There are two other considerations that come into play in the design of an application. The first is related to the choice of the token for the identity claim in the case of speaker verification. The identity claim can be combined with the verification processing in systems that have both speaker and speech recognition. In this case, an account number or a name can be used. As can be seen from Table 37.4, verification using text (first and last names) is challenging, mainly due to the short length of speech. For a very large-scale deployment, recognition can also be very challenging. *Heck* and *Genoud* have suggested combining verification and recognition scores to re-sort the N-best list output from the recognizer and achieve significant recognition accuracy gains [37.44]. Other means of claiming an identity over the telephone include caller identification (ID) and keypad input. In these cases, the verification utterance can be anything, including a lexicon common to all users. The second consideration is the flexibility that the enrollment lexicon provides to dynamically select a subset of the lexicon with which to challenge the user in order to protect against recordings (see above). This is the main reason why digit strings (telephone and account number, for example) are appealing for a relatively short enrollment session. A good speaker model can be built to deliver good accuracy even with a random subset of the enrollment lexicon as the testing lexicon (Sect. 37.3.2).

Cost of Deployment

The cost of deploying a speaker recognition system into production is also a challenge. Aside from dialog design and providing the system with a central processing unit (CPU), storage, and bandwidth, setting the operating point (the security level or the target false-acceptance rate) has a major impact on cost. As can be seen from the discussion above, there are a wide variety of dialogs that can be implemented, and all of these require their own set of thresholds depending on the level of security required. This is a very complex task that is usually solved by collecting a large number of utterances and hiring professional services from the vendor to recommend those thresholds. This can be very costly for the application developer. Recently, there has been an effort to build off-the-shelf security settings into products [37.36]. This technique does not require any data and is accurate enough for small- to medium-scale systems or initial security settings for a trial. Most application developers, however, want to have a more-accurate picture of the accuracy of their security layer and want a measurement on actual data of the standard false accept (FA), false reject (FR), and reprompt rates (RR, the proportion of genuine speakers that are reprompted after the first utterance). To this end a data collection is set up. The most expensive portion of data collection is to gather enough impostor attempts to set the decision threshold to achieve the desired FA rate with a high level of confidence. Collecting genuine speaker attempts is fairly inexpensive by comparison. An algorithm aimed at setting the FA rate without specifically collecting impostor attempts has been presented [37.45]. See Sect. 37.3.7 for more details.

Forward Compatibility

Another challenge from a deployment perspective, but that has ramifications into the technology side, is forward compatibility. The main point here is that the database of enrollee (those that have an existing speaker model) should be forward compatible to revision of: (a) the

application, and (b) the software and its underlying algorithms. Indeed, an application that has been released using a security layer based on a first name and last name lexicon is confined to using this lexicon. This is very restrictive. Also, in commercial systems, the enrollment utterances are not typically saved: the speaker model is the unit saved. This speaker model is a parameterized version of the enrollment utterances. The first step that goes into this parameterization is the execution of the front-end feature extractor (Sect. 37.1.1). The definition of these features is an integral part of the speaker model and any change to this will have a negative impact on accuracy. This also restricts what research can contribute to an existing application.

37.3 Selected Results

In this section, we will present several results that either support claims and assertions made earlier or illustrate current challenges in text-dependent speaker recognition. It is our belief that most if not all of these represent potential areas for future advances.

37.3.1 Feature Extraction

Some of the results presented below are extracted from studies done on text-independent speaker recognition tasks. We believe that the algorithms presented should also be beneficial to text-dependent tasks, and thus could constitute the basis for future work.

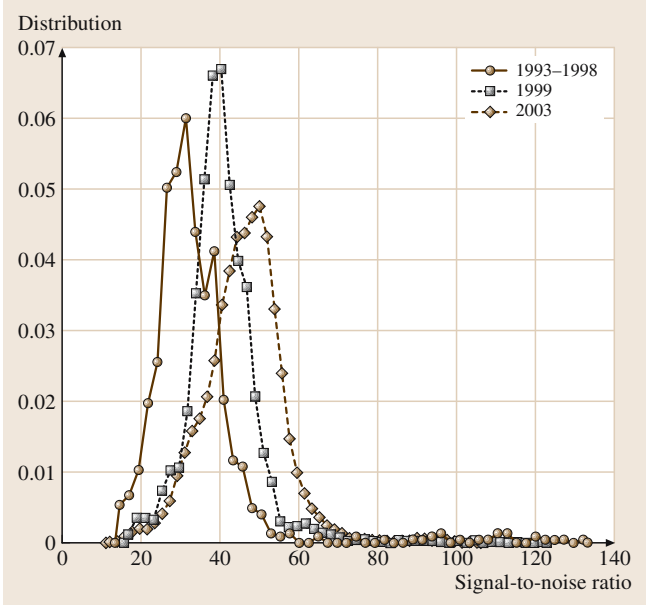

Fig. 37.2 Signal-to-noise ratio distribution from cellular waveforms for three different periods. The data are from a mix of in-service data, pilot data, and data collection

Impact of Codecs on Accuracy

The increasing penetration of cellular phones in society has motivated researchers to investigate the impact of different codecs on speaker recognition accuracy. In 1999, a study of the impact of different codecs was presented [37.46]. Speech from an established corpora was passed through different codecs (GSM, G.729 and G723.1), and resynthesized. The main conclusion of this exercise was that the accuracy drops as the bit rate is reduced. In that study, speaker recognition from the codec parameters themselves was also presented.

Figure 37.2 presents the distribution of the signal-to-noise ratio (SNR) from different internal corpora (trials and data collections) for cellular data only. We have organized them by time periods. A complete specification of those corpora is not available (codecs used, environmental noise conditions, analog versus digital usage, etc.). Nevertheless, it is obvious that speech from cellular phones is cleaner in the 2003 corpora than ever before. This is likely due to more-sophisticated codecs and better digital coverage. It would be interesting to see the effect on speaker recognition and channel identification of recent codecs like CDMA (code division multiple access) in a study similar to [37.46]. This is particularly important for commercial deployments of (text-dependent) speaker recognition, which are faced with the most up-to-date wireless technologies.

Feature Mapping

Feature mapping was introduced by *Reynolds* [37.12] to improve channel robustness on a text-independent speaker recognition task. Figure 37.3a describes the offline training procedure for the background models. The root GMM is usually trained on a collection of utterances from several speakers and channels using k-means and EM (expectation maximization) algorithms. MAP (maximum a posteriori) adaptation [37.31] is used to adapt the root GMM with utterances coming from single channels to produce GMMs for each channel. Because of

the MAP adaptation structure, there exists a one-to-one correspondence between the Gaussians from the root and channel GMMs, and transforms between Gaussians of these GMMs can be calculated [37.12]. The transforms from the channel GMMs Gaussians to the root GMM Gaussians can be used to map features from the those channels onto the root GMM. The online procedure is represented on Fig. 37.3b. For an incoming utterance, the channel is first selected by picking the most likely over the entire utterance based on $\log p(\boldsymbol{X}|\overline{\lambda})$ from (37.1). The features are then mapped from the identified channel onto the root GMM. At this point, during training of the speaker model, mapped features are used to adapt the root GMM. Conversely, during testing, the mapped features are used to score the root and speaker model GMMs to perform likelihood-ratio scoring (Sect. 37.1.3).

Feature mapping has proved its effectiveness for channel robustness (see [37.12] for more details). It is of interest for text-dependent speaker recognition because it is intimately related to speaker model synthesis (SMS) [37.13], which has demonstrated its effectiveness for such tasks [37.40]. To our knowledge, feature mapping has never been implemented and tested on a text-dependent speaker recognition task.

Speaker and Speech Recognition Front Ends

The most common feature extraction algorithms for speech recognition are mel-filter cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCCs). These algorithms have been developed with the objective of classifying phonemes or words (lexicon) in a speaker-independent fashion. The most common feature extraction algorithms for speaker recognition are, surprisingly, MFCC or LPCC. This is surprising because of the fact that speaker recognition objective is the classification of speakers, with no particular emphasis on lexical content. A likely, but still to be proven, explanation for this apparent dichotomy is that MFCC and LPCC are very effective at representing a speech signal in general. We believe that other approaches are worth investigating.

Several studies have tried to change the speaker recognition paradigm for feature extraction (see [37.47, 48], to name a few). In [37.47], a neural net with five layers is discriminatively trained to maximize speaker discrimination. Then the last two layers are discarded and the resulting final layer constitutes the feature extractor. Authors report a 28% relative improvement over MFCCs in a text-independent speaker recognition task.

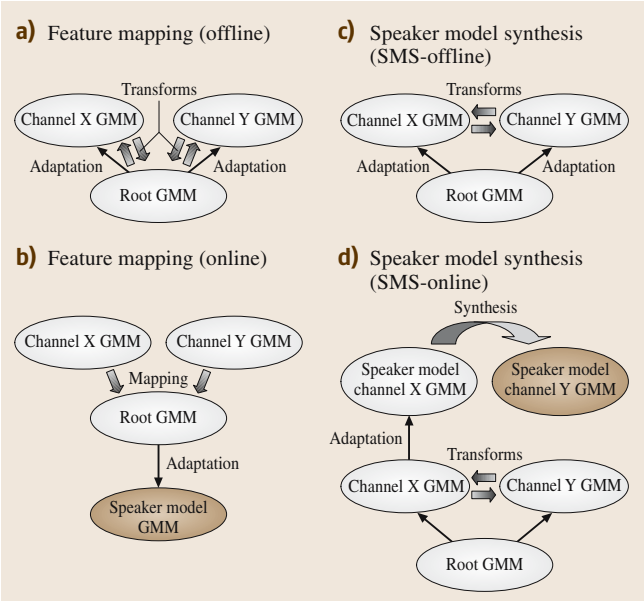

Fig. 37.3a–d Feature mapping and speaker model synthesis (SMS). GMMs with oblique lines were constructed using synthesized data

Although developed with channel robustness in mind, we believe that this technique holds a lot of potential. In [37.48], wavelet packet transforms are used to analyze the speech time series instead of the standard Fourier analysis. Authors report a 15–27% error rate reduction on an text-independent speaker recognition task. Despite the improvements reported, these algorithms have not reached mainstream adoption to replace MFCCs or LPCCs.

37.3.2 Accuracy Dependence on Lexicon

As mentioned in the Chap. 36, the theme of the lexical content of the password phrase is central in text-dependent speaker recognition. A study by *Kato* and *Shimizu* [37.15] has demonstrated the importance of preserving the sequence of digits to improve accuracy. The authors report a relative improvement of more than 50% when the digit sequence in the testing phase preserves the order found during enrollment.

Table 37.4 presents a similar trend as well as additional conditions. The data for these experiments was collected in September of 2003 from 142 unique speakers (70 males and 72 females). Each caller was requested to complete at least four calls from a variety of handsets (landline, mobile, etc.) in realistic noise conditions. In each call, participants were requested to read a sheet

with three repetitions of the phrases E, S, R, pR, and N (refer to Tables 37.2 and 37.3 for explanations of the acronyms). There were only eight unique S digit strings in the database in order to use round-robin imposter attempts, and a given speaker was assigned only one S string. The interesting fact about this data set is that we can perform controlled experiments: for every call, we can substitute E for S and vice versa, or any other types of utterances. This allows the experimental conditions to preserve:

1. callers
2. calls (and thus noise and channel conditions) and vary lexical content only

The experiments in Table 37.4 are for speaker verification and the results presented are the equal error rates (EERs). All results are on 20 k genuine speakers attempts and 20 k imposter attempts. The enrollment session consists of three repetitions of the enrollment token while the testing sessions has two repetitions of the testing token. Let us define and use the following notation to describe an experiment's lexical content for the enrollment and verification: eXXX_vYY, which defines the enrollment as three repetitions of X, and the testing attempts as two repetitions of Y. For example, the EER for eEEE_vRR is 13.2%.

The main conclusion of *Kato* and *Shimizu* [37.15] are echoed in Table 37.4: sequence-preserving digit strings improves accuracy. Compare the EERs for eEEE_vRR with eEEE_vpRpR. Also, eEEE_vEE, eSSS_vSS, epRpRpR_vpRpR, and eNNN_vNN all perform better than eRRR_vRR. This illustrates the capture by the speaker model of coarticulation: E and R utterances have exactly the lexicon (1 to 9) but in a different order. Note that the accuracy of the first and last names is significantly worse than E or S on the diagonal of Table 37.4. This is due to the average length of the password phrase: an E utterance has on average 3.97 s of speech while an N utterance has only 0.98. Finally, we have included cross-lexicon results, which are more relevant to text-independent speaker recognition (for example eEEE_vNN). This illustrates the fact that, with very short enrollment and verification sessions, lexically mismatched attempts impact accuracy significantly. In [37.4], the effect of lexical mismatch is compared with the effect of SNR mismatch and channel mismatch. It is reported that a moderate lexical mismatch can degrade the accuracy more than SNR and is comparable to channel mismatch (Table 37.1). Finally, *Heck* [37.41] noted that 'advances in robustness to linguistic mismatches will form a very fruitful bridge between text-independent and dependent tasks.' We share this view and add that solving this problem would open avenues to perform accurate and non-intrusive protection against recordings by verifying the identity of a caller across an entire call even with a very short enrollment session. We will explore this more in Sect. 37.3.6.

37.3.3 Background Model Design

The design of background models is crucial to the resulting accuracy of a speaker recognition system. The effect of the lexicon can also be seen in this context. As an example, in a text-dependent speaker recognition task based on *My voice is my password* (MVIMP) as the password phrase, adapting a standard background model with utterances of the exact target lexicon can have a significant positive impact. Without going into the details of the data set, the EER drops from 16.3% to 11.8% when 5 k utterance of MVIMP were used to adapted the background model. This is consistent with one of the results from [37.19]. In [37.49], an algorithm for the selection of background speakers for a target user is presented as well as results on a text-dependent task. The algorithm is based on similarity between two users' enrollment sessions. Lexical content was not the focus of that study, but it would be interesting to see if the lexical content of each enrollment sessions had an influence on the selection of competitive background speakers, i. e., whether similar speakers have significant lexical overlap.

From the point of view of commercial deployments, the use of specialized background models for each password phrase, or on a per-target user basis, is unrealistic. New languages also require investments to develop language-specific background models. The technique in [37.50] does not require offline training of the background model. The enrollment utterances are used to train the 25-state HMM speaker model and a lower-complexity background model. The reasoning behind this is that the reduced complexity model will smear the speaker characteristics that are captured by the higher-complexity model (speaker model). Unfortunately, this technique has never been compared to a state-of-the-art speaker recognition system.

37.3.4 T-Norm in the Context of Text-Dependent Speaker Recognition

As mentioned in Sect. 37.1.5, the T-norm is sensitive to the lexicon of the utterances used to train the imposter speaker models composing the cohort [37.45]. In that study, the data used is a different organization of the data set described in Sect. 37.3.2 that allows a separate set of speakers to form the cohort needed by the T-norm. The notation introduced in Sect. 37.3.2 can also be adapted to describe the lexicon used for the cohort: eXXX_vYY_cZZZ describes an experiment for which the speaker models in the cohort are enrolled with three repetitions of Z. The baseline system used for the experiments in that study is described in *Teunen* et al. [37.13]. It uses gender- and handset-dependent background models with speaker model synthesis (SMS). The cohort speaker models are also tagged with gender and handset; the cohorts are constructed on a per-gender and per-handset basis. During an experiment, the selection of the cohort can be made after the enrollment session based on the detected handset and gender from the enrollment session. It can also be made at test time using the handset and gender detected from the testing utterance. We denote the set of cohorts selected at testing by C_t. In the results below, we consider only the experiments eEEE_vEE or eSSS_vSS with lexically rich (cSSS) or lexically poor cohorts (cEEE). A note on *lexically rich and poor* is in order: the richness comes from the variety of contexts in which each digit is found. This lexical richness in the cohort builds robustness with respect to the variety of digits strings that can be encountered in testing.

Table 37.5 shows the accuracy using test-time cohort selection C_t in a speaker verification experiment. It is interesting to note that the use of a lexically poor cohort (cEEE) in the context of an eSSS_vSS experiment significantly degrades accuracy. In all other cases in Table 37.5, the T-norm improves the accuracy. A smoothing scheme was introduced to increase robustness to the lexical poorness of the cEEE cohort. It is suggested that this smoothing scheme increases the robustness to lexical mismatch for the T-norm. The smoothing scheme is based on the structure of (37.1), which can be rewritten in a form similar to (37.3) using $\mu(\boldsymbol{X}) = \log p(\boldsymbol{X}|\overline{\lambda})$ and $\sigma(\boldsymbol{X}) = 1$. The smoothing is then an interpolation of the normalizing statistics between standard T-norm [$\mu_{\mathrm{T}}(\boldsymbol{X})$ and $\sigma_{\mathrm{T}}(\boldsymbol{X})$] and background model normalization [$\log p(\boldsymbol{X}|\overline{\lambda})$ and 1]. Figure 37.4 shows DET (detecttion error trade-off) curves for the eSSS_vSS experiment with different cohorts. It is shown

Table 37.5 The FR rates at FA = 1% for various configurations [37.35]. Based on the lower number of trials (the impostor in our case), the 90% confidence interval on the measures is 0.6%. (© 2005 IEEE)

Experimental set-up	Baseline (no T-norm)	T-norm C^t cEEE	T-norm C^t cSSS
eEEE_vEE	17.10%	14.96%	14.74%
eSSS_vSS	14.44%	16.39%	10.42%

that the T-norm with a cEEE cohort degrades the accuracy compared to the baseline (no T-norm) as mentioned above. Smoothed T-norm achieves the best accuracy irrespective of the cohort's lexical richness (a 28% relative improvement of FR at fixed FA).

37.3.5 Adaptation of Speaker Models

Online adaptation of speaker models [37.39,40] is a central component of any successful speaker recognition application, especially text-dependent tasks because of the short enrollment sessions. The results presented in this section all follow the same protocol [37.40]. Unless otherwise stated, the data comes from a Japanese digit data collection. There were 40 speakers (gender balanced) making at least six calls: half from landlines

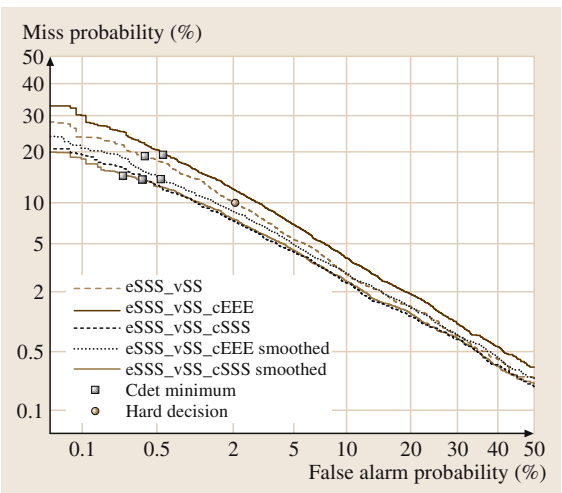

Fig. 37.4 DET curve showing the T-norm and its smoothed variant in the case of eSSS_vSS with the cohort selected at testing time (C^t). The interested reader should refer to the source paper for additional details. For clarity, the order in the legend is the same as the order of the curves at false alarm probability equal to 0.1% (after [37.35], ©2005 IEEE)

and have from cellular phones. The data was heavily recycled to increase the number of attempts by enrolling several speaker models for a given speaker and varying the enrollment lexicon (130–150 on average). For any given speaker model, the data was divided into three disjoint sets: an enrollment set to build the speaker model, an adaptation set, and a test set. The adaptation set was composed of one imposter attempt for every eight genuine attempts (randomly distributed). The experiments were designed as follows. First all of the speaker models were trained and the accuracy was measured right after the enrollment using the test set. Then, one adaptation utterance was presented to each of the speaker models. At this point a decision to adapt or not was made (see below). After this first iteration of adaptation, the accuracy was measured using the test set (without the possibility of adaptation on the testing data). The adaptation and testing steps were repeated for each adaptation iterations in the adaptation set. This protocol was designed to control with great precision all the factors related to the adaptation process: the accuracy was measured after each adaptation iteration using the same test set and they are therefore directly comparable.

Two different types of adaptation experiments can be designed based on how the decision to update the speaker models is made: supervised and unsupervised [37.39]. Both types give insight into the adaptation process and its effectiveness, and both have potential applicability in commercial deployments. Supervised adaptation experiments use the truth about the source of the adaptation utterances: an utterance is used for updating a speaker model only when it is from the target speaker. This allows the update process of the speaker models to be optimal for two reasons. The first is that there is no possibility of corruption of a speaker model by using utterances from an imposter. The second comes from the fact that all adaptation utterances from the target speaker are used to update speaker model. This allows more data to update the speaker model, but more importantly it allows poorly scoring utterances to update the speaker model. Because these utterances score poorly, they have the most impact on accuracy because they bring new and unseen information (noise conditions, channel types, etc.) into the speaker model. This has a significant impact on the cross-channel accuracy, as we will show below. Supervised adaptation can find its applicability in commercial deployments in a scenario where two-factor authentication is used, where one of the factors is acoustic speaker recognition. As an example, in a dialog where acoustic speaker recognition and authentication using a secret challenge question are used, supervised adaptation can be used if the answer to the secret question is correct.

In unsupervised adaptation, there is no certainty about the source of the adaptation utterance and usually the score on the adaptation utterance using the non-updated speaker model is used to make the decision to adapt or not [37.33, 36, 39, 40]. A disadvantage of this approach is the possibility that the speaker models may become adapted on imposter utterances that score high. This approach also reduces the number of utterances that are used to update the speaker model. More importantly, it reduces the amount of new and unseen information that is used for adaptation because this new and unseen information will likely score low on the existing speaker model and thus not be selected for adaptation.

Variable Rate Smoothing

Variable rate smoothing (VRS) was introduced in [37.30] for text-independent speaker recognition. The main idea is to allow means, variances, and mixture weights to be adapted at different rates. It is well known that the first moment of a distribution takes fewer samples to estimate than the second moment. This should be reflected in the update equations for speaker model adaptation by allowing the smoothing coefficient to be different for means, variances, and mixture weights. The authors reported little or no gains on their task. However, VRS should be useful for text-dependent speaker recognition tasks due to the short enrollment sessions. Please refer to [37.30] for the details. In [37.51], VRS was

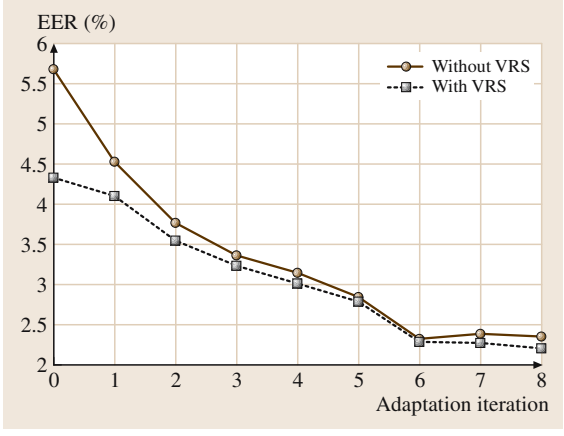

Fig. 37.5 The effect of unsupervised adaptation on the EER (percentage) with and without variable rate smoothing. Adaptation iteration 0 is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%. (After [37.51])

applied to text-dependent speaker recognition; Fig. 37.5 was adapted from that publication. It can be seen that, after the enrollment (iteration 0 on the graph), VRS is most effective because so little data has been used to train the speaker model: smoothing of the variances and mixture weights is not as aggressive as for means because the system does not have adequate estimates. As adaptation occurs, the two curves (with and without VRS) converge: at this point the estimates for the first and second moment in the distributions are accurate, the number of samples is high, and the presence of different smoothing coefficients becomes irrelevant.

Random Digit Strings

We now illustrate the effect of speaker model adaptation on contextual lexical mismatch for a digit-based speaker verification task. The experimental set-up is from a different organization of the data from Sect. 37.3.2 to follow the aforementioned adaptation protocol. Figure 37.6 illustrates the results. The testing is performed on a pseudorandom digit string (see Table 37.3 for details). Enrollment is either performed on a fixed digit string (eEEE) or on a series on pseudorandom digit strings (epRpRpR). Before adaptation occurs, the accuracy of epRpRpR is better than eEEE because the enrollment lexical conditions are matched to testing. However, as adaptation occurs and more pseudorandom utterances are added to the eEEE speaker model, the two curves converge. This shows the power of adaptation to reduce lexical mismatch and to alter the *enrollment* lexicon: in this context, the concept of *enrollment* lexicon becomes fuzzy as adaptation broadens the lexicon that was used to train the speaker model.

Speaker Model Synthesis and Cross-Channel Attempts

Speaker model synthesis (SMS) [37.13] is an extension of handset-dependent background modeling [37.8]. As mentioned before, SMS and feature mapping are dual to each another. Figure 37.3c presents the offline component of SMS. It is very similar to the offline component of feature mapping except that the transforms for means, variances, and mixture weights are derived to transform sufficient statistics from one channel GMM to another rather than from a channel GMM to the root GMM. During online operation, in enrollment, a set of utterances are tagged as a whole to a specific channel (the likeliest channel GMM – the enrollment channel). Then speaker model training (Sect. 37.1.4) uses adaptation with variable rate smoothing [37.30,51] of the enrollment channel GMM. The transforms that have been derived offline are then used at test time to synthesize the enrolled channel GMM across *all* supported channels (Fig. 37.3d). The test utterance is tagged using the same process as enrollment by picking the likeliest channel GMM (the testing channel). The speaker model GMM for the testing channel and the testing channel GMM are then used in the likelihood ratio scoring scheme described in Sect. 37.1.3 and (37.1).

The power of speaker model adaptation (Sect. 37.1.6) when combined with SMS is its ability to synthesize

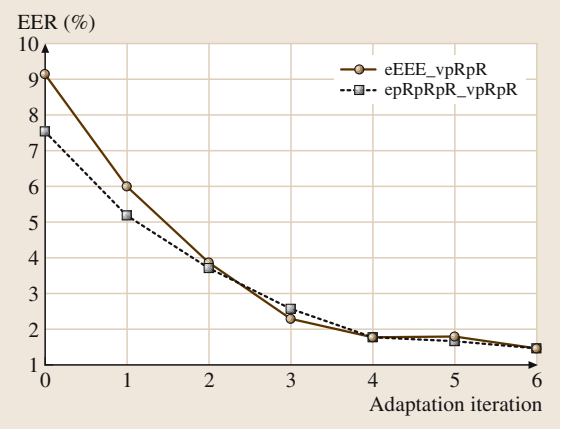

Fig. 37.6 The effect of unsupervised adaptation on reducing the contextual lexical mismatch as depicted by a reduction of the EER. Adaptation iteration 0 is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%

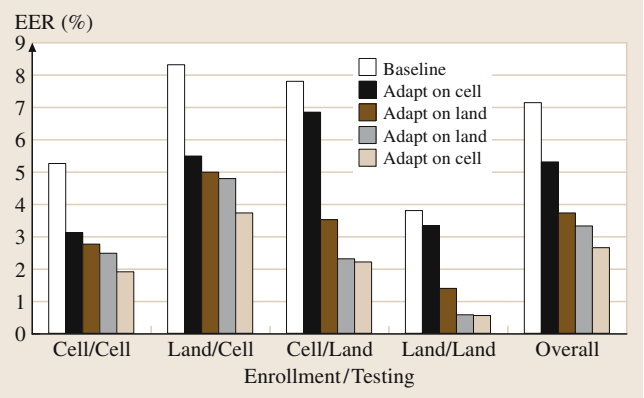

Fig. 37.7 The effect of speaker model adaptation and SMS on the cross-channel accuracy (EER). The interested reader should refer to this paper for additional details. The baseline is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.6%. (After [37.40])

sufficient statistics across all supported channels. For example, assume that a speaker is enrolled on channel X and a test utterance is tagged as belonging to channel Y. Then, if the test utterance is to be used for adaptation of the speaker model, the sufficient statistics from that utterance is gathered. The transform from Y → X is used to synthesize sufficient statistics from channel Y to channel X before adaptation of the speaker model (on channel X) occurs. Concretely, this would allow adaptation utterances from channel Y to improve the accuracy on all other channels.

Figure 37.7 illustrates the effect of speaker model adaptation with SMS. Results are grouped in *enrollment/testing* conditions: within a group, the enrollment and testing channels are fixed, the only variable is the adaptation material. For each group, the first bar is the accuracy after enrollment. The second bar is the accuracy after one iteration of adaptation on cellular data. The third bar shows the accuracy after the iteration of adaptation on cellular data followed by an iteration on a landline data, and so on. Note that these results are for supervised adaptation and thus an iteration of adaptation on a given speaker model necessarily means an actual adaptation of the speaker model. There are two interesting facts about this figure. The first important feature is that the biggest relative gain in accuracy is when the channel for the adaptation data is matched with the previously unseen testing utterance channel (see the relative improvements between the first and second bars in the *cell/cell* and *land/cell* or between the second and third bars in the *cell/land* and *land/land* results). This is expected since the new data is matched to the (previously unseen) channel of the testing utterance. The other important feature illustrates that the SMS (resynthesis of sufficient statistics) has the ability to improve accuracy even when adaptation has been performed on a different channel than the testing utterance. As an example, in the first block of Fig. 37.7, there is an improvement in accuracy between the second and third bars. The difference between the second and third bars is an extra adaptation iteration on *land* (landline data), but note that the testing is performed on *cell*. This proves that the sufficient statistics accumulated on the *land* channel have been properly resynthesized into the *cell* channel.

Setting and Tracking the Operating Point

Commercial deployments are very much concerned with the overall accuracy of a system but also the operating point, which is usually a specific false-acceptance rate. As mentioned earlier, setting the operating point for a very secure large-scale deployed system is a costly exercise, but for internal trials and low-security solutions, an approximate of the ideal operating point is acceptable. In [37.36] and later in [37.52], a simple algorithm to achieve this has been presented: frame-count-dependent thresholding (FCDT). The idea is simple: parameterize the threshold to achieve a target FA rate as a function of

1. the length of the password phrase
2. the maturity of the speaker model (how well it is trained)

At test time, depending on the desired FA rate, an offset is applied to the score (37.1). Note that the applied offset is speaker dependent because it depends on the length of the password phrase and the maturity of the speaker model.

This parameterization has been done on a large Japanese corpora. The evaluation was conducted on 12 test sets from different languages composed of data collection, trial data and in-service data [37.36]. The operating point for the system was set up at a target FA rate of 0.525% using the above algorithm. The average of the actual FA rates measured was 0.855% with a variance of 0.671%; this new algorithm outperformed previous algorithms [37.33].

In the context of adaptation of the speaker model, the problem of setting an operating point is transformed into a problem of maintaining a constant operating point for all speakers at all times [37.37]. Note that a similar problem arises in the estimation of confidence in speech recognition when adaptation of the acoustic

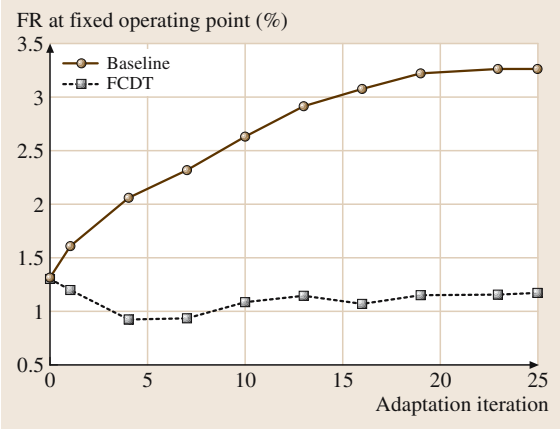

Fig. 37.8 The effect of speaker model adaptation on the FA rate with and without frame-count-dependent thresholding (FCDT). Adaptation iteration 0 is the enrollment session. The 90% confidence interval on the measures is 0.3%. (After [37.36])

models is performed [37.53]. FCDT, as well as other algorithms [37.33], can perform this task. Figure 37.8 presents the false-acceptance rate at a fixed operating point as a function of unsupervised adaptation iterations for an English digits task. After enrollment, both systems are calibrated to operate at FA = 1.3%. Then adaptation is performed. We can very easily see that the scores of the imposter attempts drift towards higher values, and hence the FA rate does not stay constant: the FA rate has doubled after 10 iterations. For commercial deployments, this problem is a crucial one: adaptation of the speaker models is very effective to increase the overall accuracy, but it must not be at the expense of the stability of the operating point. FCDT accomplishes this task: the FA rate stays roughly constant across the adaptation cycles. This leads us to think that FCDT is an effective algorithm to normalize scores against the levels of maturity of the speaker models.

Note that the imposter score drift towards higher values during speaker model adaptation in text-dependent tasks is the opposite behavior from the case of text-independent tasks [37.38, Fig. 3]. This supports the assertion that the existence of a restricted lexicon for text-dependent models has a significant impact on the behavior of speaker recognition systems: both text-dependent [37.36] and text-independent [37.38] systems being GMM-based. During the enrollment and adaptation sessions, several characteristics of the speech signal are captured in the speaker model: the speaker's intrinsic voice characteristics, the acoustic conditions (channels and noise), and the lexicon. In text-dependent speaker recognition, because of the restricted lexicon, the speaker model becomes a lexicon recognizer (the *mini-recognizer* effect). This effect increases the imposter scores because they use the target lexicon.

The FCDT algorithm can be implemented at the phone level in order to account for cases where the enrollment session (and/or speaker model adaptation) does not have a consistent lexicon. In [37.36], all experiments were carried out with enrollment and testing sessions that used exactly the same lexicon for a given user; this might seem restrictive. In the case of phone-level FCDT, the FCDT algorithm would be normalizing maturities of phone-level speaker models.

In the literature on T-norm (for text-dependent or text-independent systems; see Sect. 37.3.4), the speaker models composing the cohorts were all trained with roughly the same amount of speech. In light of the aforementioned results, this choice has the virtue of normalizing against different maturities of speaker models. We believe that the FCDT algorithm can also be used in the context of the T-norm to achieve this normalization.

37.3.6 Protection Against Recordings

As mentioned, protection against recordings is important for text-dependent speaker recognition systems. If the system is *purely* text dependent (that is the enrollment and testing utterances have the same lexicon sequence), once a fraudster has gained access to a recording, it can become relatively easy to break into an account [37.42]. This, however, must be put in perspective. A high-quality recording of the target speaker's voice is required as well as digital equipment to perform the playback. Furthermore, for any type of biometric, once a *recording* and playback mechanism are available the system becomes vulnerable. The advantage that voice authentication has over any other biometrics is that it is natural to prompt for a different sequence of the enrollment sequence: this is impossible for iris scans, fingerprints, etc. Finally, any nonbiometric security layer can be broken into almost 100% of the time once a *recording* of the secure token is available (for example, somebody who steals a badge can easily access restricted areas).

Several studies that assess the vulnerability of speaker recognition systems to altered imposter voices have been published. The general paradigm is that a fraudster gains access to recordings of a target user. Then using different technique the imposter's voice is altered to sound like the target speaker for any password phrase. An extreme case is a well-trained text-to-speech (TTS) system. This scenario is unrealistic because the amount of training material required for a good-quality TTS voice is on the order of hours of high-quality, phonetically balanced recorded speech. Studies along these lines, but using a smaller amount of data, can be found in [37.54, 55]. Even if these studies report the relative weakness of GMM-based speaker recognition systems, these techniques require sophisticated signal processing software and expertise to perform experimentation, along with high-quality recordings. A more-recent study [37.56] has also demonstrated the effect of speech transformation on imposter acceptance. This technique, again, requires technical expertise and complete knowledge of the speaker recognition system (feature extraction, modeling method, UBM, target speaker model, and algorithms). This is clearly beyond the grasp of fraudsters because implementations of security systems are usually kept secret, as are the internals algorithms of commercial speaker recognition systems.

Speaker Recognition Across Entire Calls

Protection against recordings can be improved by performing speaker recognition (in this case verification) across entire calls. The results presented here illustrate a technique to implement accurate speaker recognition across entire calls with a short enrollment session (joint unpublished work with Nikki Mirghafori). It relies heavily on speaker model adaptation (Sect. 37.1.6) and PCBV (Sect. 37.1.4). The verification layer is designed around a password phrase such as an account number. The enrollment session is made up of three repetitions of the password phrase only, while the testing sessions are composed of one repetition of the password phrase followed by non-password phrases. This is to simulate a dialog with a speech application after initial authentication has been performed. Adaptation is used to learn new lexical items that were not seen during enrollment and thus improve the accuracy when non-password phrases are used. The choice for this set-up is motivated by several factors. This represents a possible *upgrade* for currently deployed password-based verification application. It is also seamless to the end user and does not require re-enrollment: the non-password phrases are learnt using speaker model adaptation during the verification calls. Finally it is believed that this technique represents a very compelling solution for protection against recordings.

Note that this type of experiment is at the boundary between text-dependent and text-independent speaker recognition because the testing session is cross-lexicon for certain components. It is hard to categorize this type of experimental set-up because the enrollment session is very short and lexically constrained compared to its text-independent counterpart. Also, the fact that some testing is made cross-lexicon means that it does not clearly belong to the text-dependent speaker recognition field.

In order to benchmark this scenario, Japanese and Canadian French test sets were set up with eight-digit strings (account number) as the *password phrase*. The initial enrollment used three repetitions of the password phrase. We benchmark accuracy on the password and on non-password phrases. In these experiments, the non-password phrases were composed of general text such as first/last names, dates, and addresses. For adaptation, we used the same protocol as in Sect. 37.3.5 with a held-out set composed of non-password phrases (supervised adaptation). Section 37.3.5 has already demonstrated the effectiveness of adaptation on password phrases; these results show the impact, on both password and non-password phrases, of adapting on non-password phrases. Figure 37.9 presents the EER as a function of adaptation iteration, when adapting on non-password phrases for a single GMM or PCBV solution and testing on password phrases. It can be seen that the GMM solution is very sensitive to adaptation on non-password phrases,

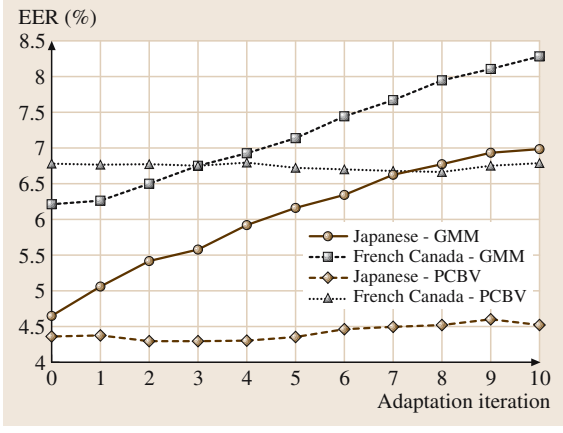

Fig. 37.9 The effect of speaker model adaptation with non-password phrases on the accuracy of password phrases (EER). Adaptation iteration 0 is the enrollment session. The experiments were carried out on over 24 k attempts from genuine speaker and imposters. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%

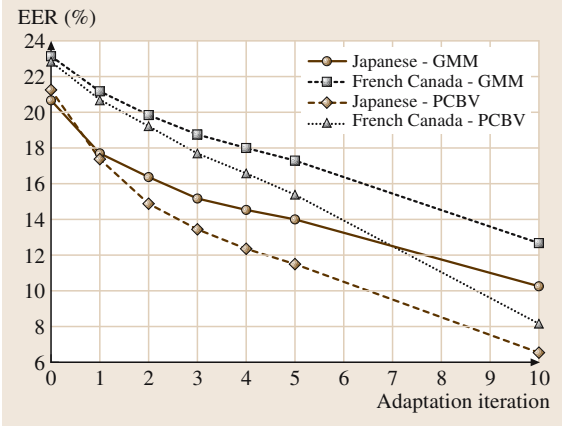

Fig. 37.10 The effect of speaker model adaptation with non-password phrases on the accuracy of non-password phrases (EER). Adaptation iteration 0 is the enrollment session. The experiments were carried out on over 14 k attempts from genuine speaker and imposters. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.5%

Table 37.6 The measured FA rate using an automatic impostor trial generation algorithm for different conditions and data sets. Note that the target FA rate was 1.0%

Experimental setup	English_US_1	English_US_2	English_UK_1	Average
[37.45] algorithm without offset	2.10%	1.15%	2.39%	1.88%
[37.45] algorithm with 0.15 offset	1.13%	0.81%	1.35%	1.10%
New binning without offset	0.86%	0.75%	1.25%	0.95%

whereas the PCBV is not. This is due to the fact that PCBV uses alignments from a speech recognition engine to segregate frames into different modeling units while the GMM does not: this leads to smearing of the speaker model in the case of the GMM solution. Figure 37.10 shows the improvements in the accuracy on non-password phrases in the same context. Note that iterations 1–5 do not have overlapping phrases with the testing lexicon: iteration 10 has some overlap, which is not unrealistic from a speech application point of view. As expected, the accuracy of the non-password phrase is improved by the adaptation process for both GMM and PCBV, with a much greater improvement for PCBV. After 10 adaptation iterations, the accuracy is 6–8% EER (and has not yet reached a plateau), which makes this solution a viable solution. It can also be noted that PCBV with adaptation on non-password phrases improves the accuracy faster than its single-GMM counterpart, taking half the adaptation iterations to achieve a similar EER (Fig. 37.10). In summary, speaker model adaptation and PCBV form the basis for delivering stable accuracy on password phrases while dramatically improving the accuracy for non-password phrases. This set of results is another illustration of the power of speaker model adaptation and represents one possible implementation for protection against recordings. Any improvement in this area is important for the text-dependent speaker recognition field as well as commercial applications.

37.3.7 Automatic Impostor Trials Generation

As mentioned above, application developers usually want to know how secure their speech application is. Usually, the design of the security layer is based on the choice of the password phrase, the choice of the enrollment and verification dialogs, and the security level (essentially the FA rate). From these decisions follow the FR and RR rates. Using off-the-shelf threshold settings will usually only give a range of target FA rates, but will rarely give any hint on the FR and RR for the current designed dialog [37.36]. Often application developers want a realistic picture of the accuracy of their system (FA, FR, and RR) based on their data. Since the FA rate is important, this has to be measured with a high degree of confidence. To do this, one requires a tremendous amount of data. As an example, to measure an FA of $1\% \pm 0.3\%$ nine times out of ten, 3000 imposter trials are required [37.1]. For a higher degree of precision such as $\pm 0.1\%$, more than 30 000 imposter trials are needed. Collecting data for imposter trials results in a lot of issues; it is costly, requires data management and tagging, cannot really be done on production systems if adaptation of speaker models is enabled, etc. However, collecting genuine speaker attempts can be done simply by archiving utterances and the associated claimed identity; depending on the traffic, a lot of data can be gathered quickly. Note that some manual tagging may be required to flag true imposter attempts – usually low-scoring genuine speaker attempts. The data gathered is also valuable because it can come from the production system.

For password phrases that are common to all users of a system, generating imposter attempts is easy once the data has been collected and tagged: it can be done using a round-robin. However, if the password phrase is unique for each genuine speaker, a round-robin cannot be used. In this case, the lexical content of the imposter attempts will be mismatched to the target speaker models, the resulting attempt will be grossly unchallenging and will lead to underestimation of the actual FA rate. In [37.45], an algorithm to estimate the FA rate accurately using only genuine speaker attempts was presented. The essence of the idea is to use a round-robin for imposter trial generation, but to quantify the amount of lexical mismatch between the attempt and target speaker model. Each imposter attempt will have a lexical mismatch value associated with it. This can be thought of as a lexical distance (mismatch) between two strings. Intuitively, we want the following order for the lexical mismatch value with respect to the target string $1234 : 1234 < 1256 < 1526 < 5678$. Note that [37.45] and the following are based on digit strings, but can easily be applied to general text by using phonemes as the atom instead of digits. A variant of the Levenstein distance was used to bin imposter attempts. For each bin,

the threshold to achieve the target FA rate was calculated. A regression between the Levenstein distance and threshold for the target FA is used to extrapolate the operational threshold for the target FA rate. For the development of this algorithm, three test sets from data collections and trials were used. These had a set of *real* impostor attempts that we used to assess the accuracy of the algorithm. The first line of Table 37.6 shows the *real* FA rate measured at the operational threshold as calculated by the algorithm above. In [37.45], to achieve good accuracy, an offset of 0.15 needed to be introduced (the second line in the table). The algorithm had one free parameter. It was later noticed that, within a bin with a given Levenstein distance, some attempts were more competitive than others. For example, the target/attempt pairs 97 526/97 156 and 97 526/97 756 had the same Levenstein distance. However, the second pair is more competitive because all of the digits in the attempt are present in the target and hence have been seen during the enrollment. A revised binning was performed and is presented as the last line in Table 37.6. The average measured FA rate is much closer to the target FA rate and this revised algorithm does not require any free parameters.

Once the threshold for the desired FA rate has been calculated, it is simple to extract the FR and RR rates from the same data. Reducing the cost of deployment is critical for making speaker recognition a mainstream biometric technique. Any advances in this direction is thus important.

37.4 Concluding Remarks

This chapter on text-dependent speaker recognition has been designed to illustrate the current technical challenges of the field. The main challenges are robustness to channel and lexical mismatches. Several results were presented to illustrate these two key challenges under a number of conditions. Adaptation of the speaker models yields advantages to address these challenges but this needs to be properly engineered to be deployable on a large scale while maintaining a stable operating point. Several new research avenues were reviewed.

When relevant, parallels between the text-dependent and text-independent speaker recognition fields were drawn. The distinctions between the two fields becomes thin when considering the work by *Sturim* et al. [37.2] and text-dependent speaker recognition with heavy lexical mismatch, as described in Sect. 37.3.6. This research area should provide a very fertile ground for future advances in the speaker recognition field.

Finally, special care was taken to illustrate, using relevant (live or trial) data, the specific challenges facing text-dependent speaker recognition in actual deployment situations.

References

37.1 A. Martin, M. Przybocki, G. Doddington, D.A. Reynolds: The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspectives, Speech Commun. **31**, 225–254 (2000)

37.2 D.E. Sturim, D.A. Reynolds, R.B. Dunnk, T.F. Quatieri: Speaker verification using text-constrained gaussian mixture models, Proc. IEEE ICASSP **2002**(1), 677–680 (2002)

37.3 K. Boakye, B. Peskin: Text-constrained speaker recognition on a text-independent task, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)

37.4 D. Boies, M. Hébert, L.P. Heck: Study of the effect of lexical mismatch in text-dependent speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)

37.5 M. Wagner, C. Summerfield, T. Dunstone, R. Summerfield, J. Moss: An evaluation of commercial off-the-shelf speaker verification systems, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)

37.6 A. Higgins, L. Bahler, J. Porter: Speaker verification using randomized phrase prompting, Digit. Signal Process. **1**, 89–106 (1991)

37.7 M.J. Carey, E.S. Parris, J.S. Briddle: A speaker verification system using alpha-nets, Proc. IEEE ICASSP, Vol. 1981 (1981) pp. 397–400

37.8 L.P. Heck, M. Weintraub: Handset dependent background models for robust text-independent

speaker recognition, Proc. IEEE ICASSP **1997**(2), 1037–1040 (1997)
37.9 A.E. Rosenberg, S. Parthasarathy: The use of cohort normalized scores for speaker recognition, Proc. IEEE ICASSP **1996**(1), 81–84 (1996)
37.10 C. Barras, J.-L. Gauvain: Feature and score normalization for speaker verification of cellular data, Proc. IEEE ICASSP **2003**(2), 49–52 (2003)
37.11 Y. Liu, M. Russell, M. Carey: The role of dynamic features in text-dependent and -independent speaker verification, Proc. IEEE ICASSP **2006**(1), 669–672 (2006)
37.12 D. Reynolds: Channel robust speaker verification via feature mapping, Proc. IEEE ICASSP **2003**(2), 53–56 (2003)
37.13 R. Teunen, B. Shahshahani, L.P. Heck: A model-based transformational approach to robust speaker recognition, Proc. ICSLP **2000**(2), 495–498 (2000)
37.14 R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*, 2nd edn. (Wiley, New York 2001)
37.15 T. Kato, T. Shimizu: Improved speaker verification over the cellular phone network using phoneme-balanced and digit-sequence-preserving connected digit patterns, Proc. IEEE ICASSP **2003**(2), 57–60 (2003)
37.16 T. Matsui, S. Furui: Concatenated phoneme models for text-variable speaker recognition, Proc. IEEE ICASSP **1993**(2), 391–394 (1993)
37.17 S. Parthasarathy, A.E. Rosenberg: General phrase speaker verification using sub-word background models and likelihood ratio scoring, Proc. ICSLP **1996**(4), 2403–2406 (1996)
37.18 C.W. Che, Q. Lin, D.S. Yuk: An HMM approach to text-prompted speaker verification, Proc. IEEE ICASSP **1996**(2), 673–676 (1996)
37.19 M. Hébert, L.P. Heck: Phonetic class-based speaker verification, Proc. Eurospeech, Vol. 2003 (2003) pp. 1665–1668
37.20 E.G. Hansen, R.E. Slygh, T.R. Anderson: Speaker recognition using phoneme-specific GMMs, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)
37.21 M. Schmidt, H. Gish: Speaker identification via support vector classifiers, Proc. IEEE ICASSP **1996**(1), 105–108 (1996)
37.22 W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, Proc. IEEE ICASSP **2006**(1), 97–100 (2006)
37.23 N. Krause, R. Gazit: SVM-based speaker classification in the GMM model space, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)
37.24 S. Fine, J. Navratil, R.A. Gopinath: A hybrid GMM/SVM approach to speaker identification, Proc. IEEE ICASSP **2001**(1), 417–420 (2001)
37.25 W.M. Campbell: A SVM/HMM system for speaker recognition, Proc. IEEE ICASSP **2003**(2), 209–212 (2003)
37.26 S. Furui: Cepstral analysis techniques for automatic speaker verification, IEEE Trans. Acoust. Speech **29**, 254–272 (1981)
37.27 V. Ramasubramanian, A. Das, V.P. Kumar: Text-dependent speaker recognition using one-pass dynamic programming algorithm, Proc. IEEE ICASSP **2006**(2), 901–904 (2006)
37.28 A. Sankar, R.J. Mammone: Growing and pruning neural tree networks, IEEE Trans. Comput. **42**, 272–299 (1993)
37.29 K.R. Farrell: Speaker verification with data fusion and model adaptation, Proc. ICSLP **2002**(2), 585–588 (2002)
37.30 D.A. Reynolds, T.F. Quatieri, R. B.Dunn: Speaker verification using adapted gaussian mixture models, Digit. Signal Process. **10**, 19–41 (2000)
37.31 J.-L. Gauvain, C.-H. Lee: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE T. Speech Audi. Process. **2**, 291–298 (1994)
37.32 D.A. Reynolds: Comparison of background normalization methods for text-independent speaker verification, Proc. EuroSpeech **1997**(2), 963–966 (1997)
37.33 N. Mirghafori, L.P. Heck: An adaptive speaker verification system with speaker dependent a priori decision thresholds, Proc. ICSLP **2002**(2), 589–592 (2002)
37.34 R. Auckenthaler, M.J. Carey, H. Lloyd-Thomas: Score normalization for text-independent speaker verification systems, Digit. Signal Process. **10**, 42–54 (2000)
37.35 M. Hébert, D. Boies: T-Norm for text-dependent commercial speaker verification applications: effect of lexical mismatch, Proc. IEEE ICASSP **2005**(1), 729–732 (2005)
37.36 N. Mirghafori, M. Hébert: Parametrization of the score threshold for a text-dependent adaptive speaker verification system, Proc. IEEE ICASSP **2004**(1), 361–364 (2004)
37.37 T. Matsui, T. Nishitani, S. Furui: Robust methods for updating model and a priori threshold in speaker verificaiton, Proc. IEEE ICASSP, Vol. 1996 (1996) pp. 97–100
37.38 C. Barras, S. Meignier, J.-L. Gauvain: Unsupervised online adaptation for speaker verification over the telephone, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)
37.39 C. Fredouille, J. Mariéthoz, C. Jaboulet, J. Hennebert, J.-F. Bonastre, C. Mokbel, F. Bimbot:

Behavior of a bayesian adaptation method for incremental enrollment in speaker verification, Proc. IEEE ICASSP, Vol. 2000 (2000)
37.40 L.P. Heck, N. Mirghafori: Online unsupervised adaptation in speaker verification, Proc. ICSLP, Vol. 2000 (2000)
37.41 L.P. Heck: On the deployment of speaker recognition for commercial applications, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004), keynote speech
37.42 K. Wadhwa: Voice verification: technology overview anf accuracy testing results, Proc. Biometrics Conference, Vol. 2004 (2004)
37.43 M.J. Carey, R. Auckenthaler: User validation for mobile telephones, Proc. IEEE ICASSP, Vol. 2000 (2000)
37.44 L.P. Heck, D. Genoud: Integrating speaker and speech recognizers: automatic identity claim capture for speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2001 (2001)
37.45 M. Hébert, N. Mirghafori: Desperately seeking impostors: data-mining for competitive impostor testing in a text-dependent speaker verification system, Proc. IEEE ICASSP **2004**(2), 365–368 (2004)
37.46 T.F. Quatieri, E. Singer, R.B. Dunn, D.A. Reynolds, J.P. Campbell: Speaker and language recognition using speech codec partameters, Proc. EuroSpeech, Vol. 1999 (1999) pp. 787–790
37.47 L.P. Heck, Y. Konig, M.K. Sönmez, M. Weintraub: Robustness to telephone handset distortion in speaker recognition by discriminative feature design, Speech Commun. **31**, 181–192 (2000)
37.48 M. Siafarikas, T. Ganchev, N. Fakotakis, G. Kokkinakis: Overlapping wavelet packet features for speaker verification, Proc. EuroSpeech, Vol. 2005 (2005)
37.49 D. Reynolds: Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. **17**, 91–108 (1995)
37.50 O. Siohan, C.-H. Lee, A.C. Surendran, Q. Li: Background model design for flexible and portable speaker verification systems, Proc. IEEE ICASSP **1999**(2), 825–829 (1999)
37.51 L.P. Heck, N. Mirghafori: Unsupervised on-line adaptation in speaker verification: confidence-based updates and improved parameter estimation, Proc. Adaptation in Speech Recognition, Vol. 2001 (2001)
37.52 D. Hernando, J.R. Saeta, J. Hernando: Threshold estimation with continuously trained models in speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)
37.53 A. Sankar, A. Kannan: Automatic confidence score mapping for adapted speech recognition systems, Proc. IEEE ICASSP **2002**(1), 213–216 (2002)
37.54 D. Genoud, G. Chollet: Deliberate imposture: a challenge for automatic speaker verification systems, Proc. EuroSpeech, Vol. 1999 (1999) pp. 1971–1974
37.55 B.L. Pellom, J.H.L. Hansen: An experimental study of speaker verification sensitivity to computer voice-altered imposters, Proc. IEEE ICASSP **1999**(2), 837–840 (1999)
37.56 D. Matrouf, J.-F. Bonastre, C. Fredouille: Effect of speech transformation on impostor acceptance, Proc. IEEE ICASSP **2006**(2), 933–936 (2006)