Università Politecnica delle Marche
Dottorato di Ricerca in Ingegneria dell'Informazione
Curriculum in Ingegneria Elettronica, Biomedica e delle Telecomunicazioni

---

Sistemi di interazione vocale per la domotica

Tesi di Dottorato di:
Laura Falaschetti

Tutor:
Prof. Claudio Turchetti

Coordinatore del Curriculum:
Prof. Francesco Piazza

15° ciclo - nuova serie

Università Politecnica delle Marche
Dottorato di Ricerca in Ingegneria dell'Informazione
Curriculum in Ingegneria Elettronica, Biomedica e delle Telecomunicazioni

---

Sistemi di interazione vocale per la domotica

Tesi di Dottorato di:
Laura Falaschetti

Tutor:
Prof. Claudio Turchetti

Coordinatore del Curriculum:
Prof. Francesco Piazza

15° ciclo - nuova serie

Università Politecnica delle Marche
Dottorato di Ricerca in Ingegneria dell'Informazione
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

A coloro che hanno creduto in me e che mi hanno sostenuto in questo meraviglioso cammino

Ringraziamenti

Non sono mai stata brava con le parole, più con i numeri ed i linguaggi di programmazione, ma cercherò in queste poche righe di esprimere al meglio la mia gratitudine verso le persone che hanno creduto in me e mi hanno sostenuto in questi tre anni meravigliosi.

In primis ringrazio caldamente il mio tutor accademico, il prof. Claudio Turchetti. Il professore, con la sua infinita esperienza, saggezza e conoscenza, mi ha accompagnato costantemente in questo cammino, facendomi crescere giorno dopo giorno, consigliandomi nel corso della ricerca e guidandomi con idee valide e innovative. Lo ringrazio per i preziosi insegnamenti ricevuti e per aver reso possibile la realizzazione del mio grande sogno nel cassetto, fare ricerca.

Ringrazio il prof. Giorgio Biagetti, che con la sua immensa conoscenza, ha sempre saputo aiutarmi nei momenti di difficoltà con illuminanti deduzioni; il suo aiuto è stato prezioso fin dai tempi della tesi di Laurea Specialistica, e sono felice di avere potuto contare sul suo supporto anche durante il mio percorso di dottorato.

Ringrazio il prof. Paolo Crippa per avermi guidato, nel mondo a me completamente sconosciuto, della produzione scientifica. Pubblicazioni, citazioni, impact factor, stato dell'arte, conferenze... mi sarei persa completamente senza il suo aiuto.

Ringrazio anche il prof. Simone Orcioni per i consigli on-the-fly sull'ottimizzazione degli algoritmi che man mano stavo sviluppando.

Ringrazio il mio co-tutor per il primo anno di dottorato, l' Ing. Alessandro Curzi che mi ha aperto le porte del dottorato trasferendomi parte della sua conoscenza che spazia nei campi più disparati dell'ingegneria, ma sopratutto per avermi "indottrinato" alla programmazione intensiva ed ai sistemi operativi open source.

"The last but not the least", il mio collega e credo di poter dire amico, l' Ing. Michele Alessandrini, con il quale ho condiviso le gioie della

programmazione ed i dolori dell'assistenza studenti... oltre che innumerevoli caffè!

Un sentito ringraziamento ai colleghi dottorandi e dottorati dell'open space, insieme abbiamo formato una grande famiglia ed abbiamo condiviso momenti veramente piacevoli. In particolare Nicola, Giuseppe e Cristiano, che da semplici colleghi sono diventati i miei migliori amici.

Un ringraziamento speciale alla mia famiglia. A mia madre, che mi ha insegnato fin dall'infanzia il valore della conoscenza e della dedizione allo studio; sono questi i principi solidi che mi hanno guidata in questi anni. A mia sorella, che mi ha sempre spronato ad affrontare le sfide lavorative credendo fortemente nelle mie capacità, anche quando io non ne credevo possibile la riuscita. A mio padre, che mi ha sempre lasciata libera di scegliere per il mio futuro. Spero fortemente che siano fieri del mio percorso. A Fabrizio, che mi è stato accanto in ogni mia scelta, in ogni mio passo, supportandomi e "sopportandomi" in tutti questi anni. Grazie a tutti, con immenso e sincero affetto.

Ancona, Novembre 2016

Laura Falaschetti

Sommario

Una delle questioni aperte nell'ambito dell'home automation è la realizzazione di interfacce uomo-macchina che siano non solo efficaci per il controllo di un sistema, ma anche facilmente accessibili. La voce è il mezzo naturale per comunicare richieste e comandi, quindi l'interfaccia vocale presenta notevoli vantaggi rispetto alle soluzioni touch-screen, interruttori ecc. Il lavoro di tesi proposto è finalizzato alla realizzazione di un sistema di interazione vocale per l'home automation, in grado non solo di riconoscere singoli comandi veicolati da segnali vocali, ma anche di personalizzare i servizi richiesti tramite il riconoscimento del parlatore e di interagire mediante il parlato sintetizzato. Per ciascuna tipologia di interazione vocale, verranno proposte soluzioni volte a superare i limiti dell'approccio classico in letteratura. In prima analisi, verrà presentato un sistema di riconoscimento vocale distribuito (DSR) per il controllo delle luci, che implementa ottimizzazioni ad-hoc per operare nell'ambiente in modo non invasivo e risolvere le problematiche di uno scenario reale. Nel sistema DSR sarà integrato un algoritmo di identificazione del parlatore per ottenere un sistema in grado di personalizzare i comandi sulla base dell'utente riconosciuto. Un sistema di identificazione vocale deve essere in grado di classificare l'utente con frasi della durata inferiore a 5 s. A tal fine verrà proposto un algoritmo basato su truncated Karhunen-Loève transform con performance, su brevi sequenze di speech (< 3.5 s), migliori della convenzionale tecnica basata su Mel-Cepstral coefficients. Verrà infine proposto un framework di sintesi vocale Hidden Markov Model/unit-selection basato su Modified Discrete Cosine Transform, che garantisce la perfetta ricostruibilità del segnale e supera i limiti imposti dalla tecnica Mel-cepstral. Gli algoritmi ed il sistema proposto saranno applicati a segnali acquisiti in condizioni realistiche, al fine di verificarne l'adeguatezza.

Abstract

One of the open questions in home automation is the realization of human-machine interfaces that are not only effective for the control of the available functions, but also easily accessible. The voice is the natural way to communicate requests and commands, in this way speech interface offers considerable advantages over solutions such as touch-screen, switches etc. The proposed thesis is aimed at studying and realizing a speech interaction system for home automation to be able not only to recognize individual commands conveyed by voice signals, but also to customize the services requested through a speaker recognizer and to interact by means of synthesized speech. For each speech interaction mechanism, solutions are suggested to overcome the traditional limitations of previous work. In the first analysis, it is offered a speech distributed recognition system (DSR), for the voice control of a lighting system, that implements strategies and ad-hoc optimizations and is able to solve the typical problems of a real scenario. The DSR system can also be integrated with a speaker identification algorithm in order to obtain a system able to customize the spoken commands on the user specific settings. In the home automation, a speaker identification system must be able to classify the user with sequences of speech frames of a duration less than 5 s. To this goal, an algorithm based on truncated Karhunen-Loève transform able to produce results, with short sequences of speech frames (< 3.5 s), better than those achieved with the Mel-Cepstral coefficients, is suggested. Moreover, this work presents a novel Hidden Markov Models/unit-selection speech synthesis framework based on Modified Discrete Cosine Transform, which guarantees the perfect reconstruction of the speech signal and overcomes the main lacks of Mel-cepstral technique. The algorithms and the proposed system will be applied to signals acquired under realistic conditions, in order to verify its adequacy.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

La domotica, detta anche *home automation*, è la disciplina che si occupa di studiare le tecnologie atte a migliorare la qualità della vita nella casa. Il termine "domotica" è infatti un neologismo derivante dalla contrazione della parola latina domus (casa) unita al sostantivo "automatica", quindi significa "scienza dell'automazione delle abitazioni"; ha dunque come oggetto di studio privilegiato proprio l'automazione della casa. Quest'area fortemente interdisciplinare richiede l'apporto di tecnologie basate principalmente sull'ingegneria informatica ed elettronica, aventi per obiettivo la realizzazione di una serie di dispositivi integrati che permettano di automatizzare e facilitare l'adempimento delle varie operazioni solitamente svolte in un edificio. Tali tecnologie utilizzano informazioni ottenute da una rete informatica alla quale i dispositivi devono essere collegati. Le principali finalità della domotica sono: ottimizzare la parte impiantistica delle costruzioni in termini di funzionalità, di sicurezza e di risparmio energetico; aumentare le possibilità di intrattenimento audio-video per rendere un'abitazione più confortevole; fornire assistenza alle persone che si trovano in condizioni di isolamento o di inabilità. Quest'ultima finalità è legata al concetto di Ambient Assisted Living (AAL), termine che descrive un insieme di soluzioni tecnologiche destinate a rendere attivo, intelligente e cooperativo l'ambiente nel quale viviamo, efficace nel sostenere la vita indipendente, capace di fornire maggiore sicurezza, semplicità e benessere nello svolgimento delle attività della vita quotidiana. In questo campo, lo sviluppo di case intelligenti, dette anche *smart homes*, è visto come una via promettente per raggiungere un elevato livello di accessibilità, per anticipare e rispondere alle esigenze specifiche di persone anziane e disabili [1]. In realtà, il campo di applicazione è molto più ampio perché, essendo, come detto, l'obiettivo generale quello di migliorare la qualità della vita, le smart homes sono

chiamate a rispondere alle esigenze specifiche delle persone, sia come tecnologia assistiva che come tecnologia di consumer.

Per questo i sistemi domotici devono essere dotati di interfacce intuitive che ne rendano immediato l'utilizzo. La voce rappresenta il mezzo naturale per comunicare richieste e comandi [2, 3], e quindi l'interfaccia vocale presenta notevoli vantaggi rispetto ad altre soluzioni tattili come touch-screen, interruttori ecc.. L'interfaccia vocale rende possibile l'interazione con il linguaggio naturale in modo che l'utente non debba imparare procedure complesse [4, 5]. Un aspetto particolarmente significativo è che la voce è in grado di veicolare diversi tipi di informazioni: caratteristiche del parlato e caratteristiche del parlatore. Questo significa che un sistema controllato da segnali vocali può essere in grado di riconoscere i comandi impartiti dall'utente per il controllo del sistema stesso ma anche di personalizzarli sulla base alle specifiche esigenze dell'utente riconosciuto. Inoltre, la tendenza è quella di realizzare sistemi che siano il meno invasivi possibile, che permettano un rispetto della privacy con un costo ridotto e facilità di installazione. Un sistema di interazione vocale è in grado di rispondere anche a tutte queste richieste. Infatti un sistema di questo tipo può essere realizzato semplicemente utilizzando uno o più microfoni ed una Single-Board Computer che funge da front-end e sarà quindi installata dall'utente, mentre tutta l'elaborazione è demandata ad un calcolatore esterno. Questo è esattamente il concetto che è implementato nei sistemi di riconoscimento vocale distribuito (*Distributed Speech Recognition*).

Un sistema domotico ad interazione vocale deve comunque essere in grado di far fronte a diverse problematiche che si possono presentare in uno scenario reale di utilizzo:

- attivazione del sistema ed interazione real-time;
- speech capture: il sistema deve essere in grado di distinguere la voce in condizioni di:
  - distant speech [6]: condizione che si verifica quando la distanza tra microfono e parlatore è elevata;
  - rumore dell'ambiente: in un ambiente domestico sono presenti sorgenti di rumore interne ed esterne che degradano il rapporto segnale/rumore del segnale vocale, con conseguenze negative sul rate di riconoscimento;

  - riverbero dell'ambiente in cui si trovano la sorgente (parlatore) ed il sensore (microfono) [7]: le ripetute riflessioni che avvengono in un ambiente chiuso creano una serie di copie del segnale emesso dalla sorgente determinando una distorsione del segnale ricevuto dal microfono.

- eliminazione del testo che non corrisponde alla grammatica dei comandi: problemi legati all'interpretazione dei comandi nascono poiché l'utente potrebbe non aderire perfettamente alla grammatica e la conversazione potrebbe essere confusa con comandi di controllo;

- interpretazione ed attuazione dei comandi in tempo reale;

- riconoscimento del parlatore per garantire privacy e personalizzazione dei comandi;

- garantire una risposta intelligibile e naturale nel caso in cui il sistema preveda un sintetizzatore vocale.

Obiettivi In questo lavoro di tesi verranno prese in considerazione tutte le possibili interazioni vocali, riconoscimento, identificazione e sintesi vocale, e per ciascuna di esse verrà fornita una trattazione dettagliata fino alla descrizione dei metodi implementati da poter applicare nell'ambito domotico.

In particolare, l'obiettivo sarà quello di portare un'innovazione in ciascuno dei tre ambiti, in modo tale che questi metodi possano essere utilizzati separatamente o integrati in un unico sistema di interazione, in questo caso finalizzato alla domotica (nello specifico il controllo delle luci) ma applicabile anche ad altri campi (sistemi di controllo, sicurezza, comunicazione e informazione..). Questo sistema sarà quindi in grado non solo di riconoscere singoli comandi veicolati da segnali vocali, ma anche di riconoscere il parlatore e di interagire mediante il parlato sintetizzato [8, 4, 9].

Struttura La struttura con cui si procederà alla presentazione della tesi è illustrata di seguito.

Nel Capitolo 2 verrà fatto un breve cenno sulle caratteristiche e sull'attuale stato dell'arte relativo ai sistemi domotici ad interazione vocale.

Nel Capitolo 3 verrà descritto il primo metodo di interazione vocale: il *riconoscimento vocale* (*speech recognition*); dalle basi teoriche del processo di riconoscimento, alla descrizione delle tecniche attuali di Automatic Speech Recognition (ASR) e di Distributed Speech Recognition (DSR), fino alla descrizione di un proprio sistema DSR finalizzato al controllo delle luci. Il DSR fornisce ad un sistema vocale per il controllo domotico notevoli vantaggi in termini di robustezza del riconoscimento rispetto all'impiego di un canale vocale convenzionale, dove la compressione dovuta al codec audio e gli errori sul canale di trasmissione influenzano negativamente l'accuratezza del riconoscimento. Il DSR, inoltre, garantisce intrinsecamente l'impiego di nuove interfacce multimodali permettendo l'invio di informazioni collaterali simultaneamente alle feature del segnale vocale utilizzando lo stesso canale di comunicazione. Il sistema DSR proposto, si pone l'obiettivo di realizzare uno strumento di controllo che sia meno invasivo possibile e di mantenere una buona precisione nel riconoscimento anche in condizioni di distant speech e background noise, come può accadere in una situazione reale di utilizzo. Un requisito essenziale di una interfaccia vocale per l'home automation è rappresentata dalla capacità di riconoscere più parlatori. A tal fine verranno indagate tecniche di speaker adaptation, per adattare il modello acustico ad un qualsiasi parlatore [10, 11]. Un altro problema affrontato sarà quello legato all'utilizzo del modello linguistico per l'estrazione delle parole dal flusso di fonemi. I convenzionali sistemi ASR permettono di raggiungere elevate performance grazie all'utilizzo di modelli linguistici dettagliati. Questi modelli richiedono una larga quantità di materiale di addestramento del modello; il modello linguistico rimane il "collo di bottiglia" dato che richiede un'enorme quantità di materiale di training [12]. Numerose tecniche sono state proposte in letteratura per risolvere questo problema noto come "the curse of dimensionality", tuttavia nessuna di queste tecniche è stata inserita negli attuali sistemi Large Vocabulary Continuous Speech Recognition (LVCSR), in quanto devono essere integrate direttamente in fase di decodifica, al fine di sfruttare completamente le loro potenzialità. In questo ambito è stata implementata una tecnica "dictionary-based large vocabulary speech recognition" (DB-LVCSR), nella quale il task del decoder è quello di determinare la sequenza più probabile di fonemi; verrà proposto quindi un algoritmo che non si basa sulla stima del modello linguistico ma permette di realizzare l'estrazione automatica di parole da un flusso di fonemi in un sistema DB-LVCSR.

Il Capitolo 4 si concentrerà sul secondo tipo di interazione vocale: l'*identificazione del parlatore* (*speaker identification*). Verranno descritte le tecniche attuali con il loro vantaggi ed i loro limiti; limiti che si cercherà di superare proponendo un approccio innovativo del quale verrà fornita sia la descrizione matematica che i risultati sperimentali e che verrà infine integrato nel sistema DSR proposto nel Capitolo 3. L'approccio sperimentato si basa su rappresentazione Karhunen-Loève transform (KLT) discreta (DKLT) applicata al log-spectrum del segnale vocale; questa rappresentazione presenta proprietà di convergenza che garantiscono buone prestazioni in termini di precisione di classificazione, senza influenzare la speaker variability, come invece accade nell'approccio classico basato su Mel-Cepstral coefficients (MFCCs). Numerosi test sperimentali dimostreranno la validità dell'approccio proposto che verrà quindi integrato nel sistema DSR realizzato al fine di realizzare un sistema combinato di speech recognition/speaker identification in grado di capire "cosa è stato detto" e "da chi è stato detto", permettendo quindi, non solo il riconoscimento del comando ma anche la personalizzazione di questo sulla base delle richieste specifiche dell'utente identificato.

Il Capitolo 5 verterà sullo studio ed analisi delle tecniche e sistemi di *sintesi vocale* (*Speech Synthesis)* scendendo fino ai dettagli implementativi del sistema di sintesi vocale realizzato e validato da risultati sperimentali. Sebbene siano state studiate diverse tecniche per la sintesi di segnali vocali, l'approccio basato sui modelli Hidden Markov Models (HMM) si è dimostrato tra i più vantaggiosi in termini di performance ottenute [13, 14, 15]. Ciò nonostante un problema ancora aperto è rappresentato dalla ricostruibilità del segnale mediante delle feature in grado di assicurare la perfetta ricostruzione del segnale vocale e superare quindi i limiti imposti dal modello HMM. Il metodo di sintesi implementato combina un learning HMM delle sequenze di stati associate ai fonemi (context-dependent) ad una trasformazione Modified Discrete Cosine Transform (MDCT) che assicura la perfetta ricostruzione del segnale, unendo all'approccio HMM la tecnica unit-selection per la generazione di forme d'onda.

Il Capitolo 6 conclude la tesi, facendo il punto sul grado di innovazione introdotto nel lavoro proposto.

Capitolo 2

Sistemi di interazione vocale per la domotica

In questo capitolo verrà brevemente illustrato lo stato dell'arte relativo ai sistemi di interazione vocale per la domotica. Lo schema generale di un sistema domotico è illustrato in Fig. 2.1.

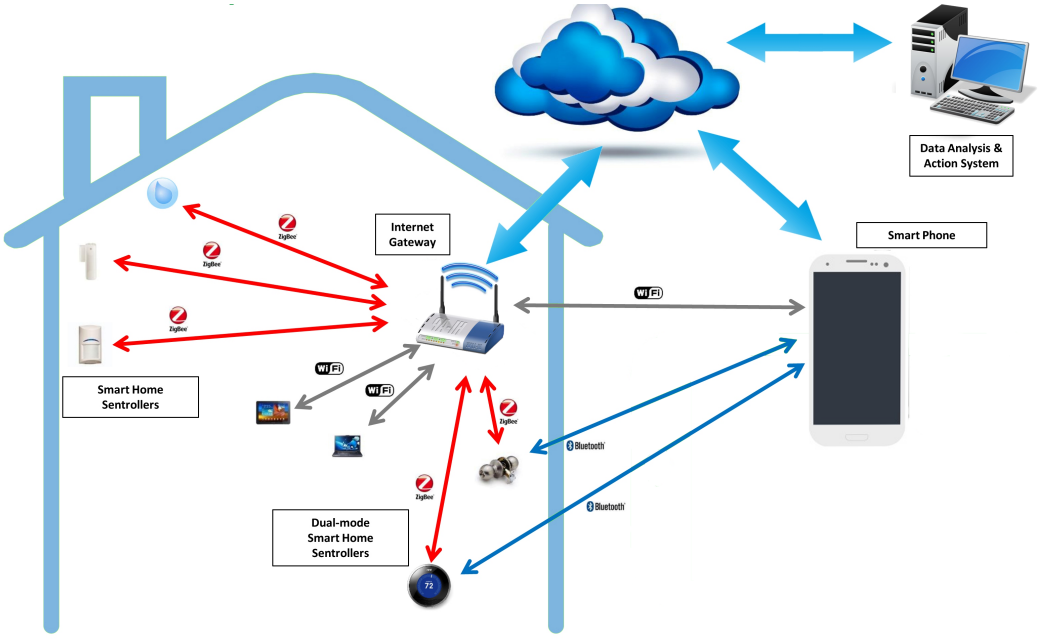


Figura 2.1: Schema generale di un sistema domotico.

La voce, come già detto nel capitolo precedente, rappresenta il mezzo più naturale per comunicare richieste e comandi, offrendo quindi all'utente una modalità

di interazione molto più semplice ed accessibile rispetto alle convenzionali soluzioni tattili (interruttori, touch-screen, ...). Questo ha spinto la ricerca nell'ambito delle smart homes verso lo studio ed implementazione di interfacce di controllo basate su voce. Si farà di seguito una panoramica sui sistemi ad interazione vocale attualmente oggetto di ricerca.

2.1 Stato dell'arte

Oggigiorno, le voice-user interfaces (VUIs) sono frequentemente impiegate in molte applicazioni (es., smartphones, desktop applications) poiché offrono un'interazione per mezzo del linguaggio naturale così che l'utente non debba imparare procedure complesse [16, 17].

Un largo numero di progetti relativi alle smart homes è finalizzato alla tecnologia assistiva come: CASAS [18], AGING IN PLACE [19], DESDHIS [20]. Ma un numero sempre maggiore di smart home projects stanno considerando l'interazione vocale nei loro sistemi: SWEET-HOME [21, 22, 23, 24], CIRDO [25, 26], PERS [27]. Alcuni di questi progetti sono focalizzati su patologie della voce (es., Alzheimer) come ALADIN [28], HOMESERVICE [29], e PIPIN [30]. Lo scopo del DIRHA [31] è invece quello di trovare soluzioni ottimali nell'ambito del "distant speech" recognition. Tuttavia, tali tecnologie devono essere convalidate in situazioni reali e SWEET-HOME è il primo sistema ad interazione vocale che è stato valutato online in una reale smart home con potenziali utenti [32]. I progetti certamente più noti, dal punto di vista commerciale, sono Amazon Echo [33], e Google Home [34]. Questi sistemi offrono sicuramente ottime performance ma si basano su sistemi cloud-based processing che possono comportare problemi relativamente alla privacy.

Si illustrano di seguito le caratteristiche dei più noti sistemi sopra citati.

SWEET-HOME L'architettura del sistema è illustrata in Fig. 2.2; per una descrizione completa si rimanda a [23].

L'ingresso è composto dall'informazione resa disponibile dalla rete del sistema domotico e derivante da 7 microfoni opportunamente disposti nell'ambiente. I segnali audio sono analizzati in real-time per mezzo del framework PATSH [5]. Il toolkit Speeral viene utilizzato come ASR [35]. Speeral permette un modeling acustico HMM-based context-dependent ed un modeling linguistico a trigram-

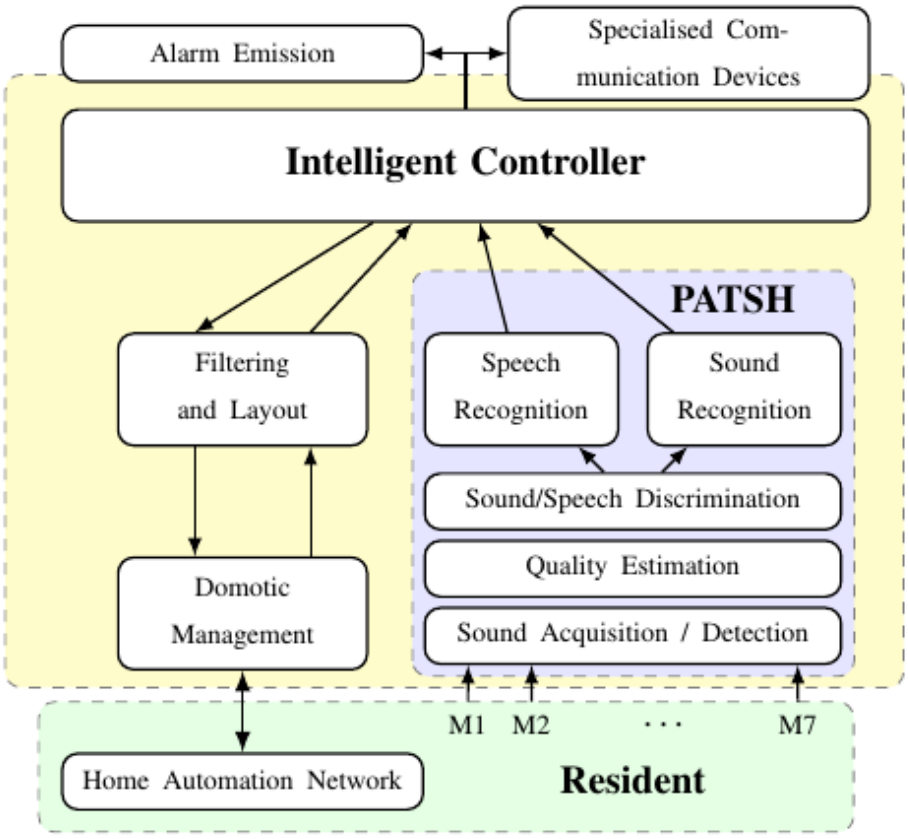


Figura 2.2: Schema a blocchi del sistema SWEET-HOME.

mi. Al termine dell'audio processing, il testo più probabilmente riconosciuto viene inviato al modulo Intelligent Controller. Quindi, le informazioni possono essere fornite direttamente dall'utente (comandi vocali) o via sensori d'ambiente (temperatura). Le informazioni provenienti dal sistema di automazione sono trasmesse on-line all'Intelligent Controller (attraverso una serie di processamenti intermedi); il controller cattura lo stream di dati, li interpreta ed esegue le azioni corrispondenti inviando i comandi riconosciuti all'attuatore attraverso la home automation network. Inoltre, il controller può inviare messaggi di alert o informazione ad un sintetizzatore vocale, in caso di emergenza o su richiesta dell'utente [36]. Questo progetto di ricerca si avvale della DOMUS smart home [37] per eseguire i test su uno scenario reale di utilizzo. La DOMUS smart home, illustrata in Fig. 2.3, è una vera è propria casa di una trentina di metri quadrati circa e composta da un bagno, una cucina, una camera da letto ed uno studio, il

tutto dotato di sensori ed attuatori. L'appartamento è pienamente utilizzabile.

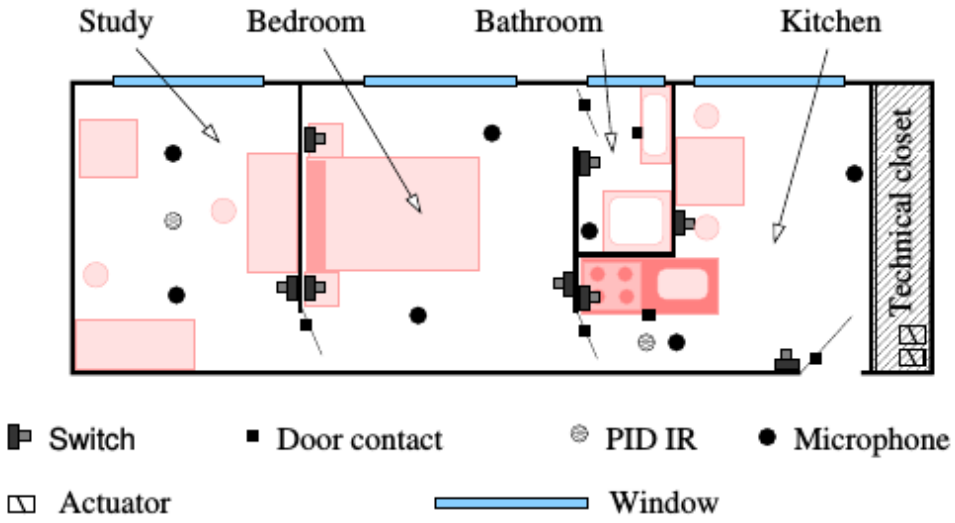


Figura 2.3: DOMUS smart-home.

AMAZON ECHO Amazon Echo è uno smart speaker sviluppato da Amazon.com, all'interno dei laboratori Lab126 in Silicon Valley e Cambridge, Massachusetts a partire dall'anno 2010. Il dispositivo è in grado di fornire: interazione vocale, riproduzione musicale, impostazione di allarmi, riproduzioni di audiolibri, informazioni su traffico e meteo; può inoltre controllare gli smart devices di un ambiente domotico se collegato ad un home automation hub. L'interazione vocale è realizzata tramite l'assistente vocale Amazon Alexa. Nella modalità di default, il dispositivo è in continuo ascolto, in attesa della "wake word". Echo richiede una connessione Wi-Fi per poter operare. Le funzioni di riconoscimento del dispositivo si basano sugli Amazon Web Services e sulla piattaforma Amazon common voice. Echo utilizza le passate registrazioni vocali che l'utente ha inviato al servizio cloud per migliorare la risposta alle domande future. Per far fronte a problemi di privacy, l'utente può cancellare le registrazioni vocali che sono attualmente associate all'account dell'utente, ma ciò potrebbe compromettere le performance del sistema.

GOOGLE HOME Anche il Google Home è uno smart speaker, che offre principalmente funzionalità per la riproduzione del suono ma, data la compatibilità con alcuni dispositivi in commercio, è in grado di offrire anche funzionalità

di controllo vocale dei dispositivi domotici. La caratteristica più importante è quella di fornire un assistente vocale, Google Assistant, basato sul Google's natural language processing algorithm che fornisce funzionalità di conversazione bidirezionale.

Capitolo 3

Speech Recognition

Integrare in un sistema domotico la funzionalità di riconoscimento vocale (*speech recognition*) permette il controllo vocale delle apparecchiature domestiche e consente quindi di aumentare il ventaglio di possibilità di interazione verso i dispositivi che usiamo quotidianamente. Questa integrazione può essere effettuata realizzando interfacce ad interazione vocale lato client sfruttando la tecnologia Distributed Speech Recognition (DSR).

Tramite la tecnologia DSR è possibile inviare il segnale vocale acquisito da vari dispositivi dislocati nello spazio attraverso un link radio di bassa qualità e convertirlo successivamente in testo per interagire con il sistema di automazione domestica. La degradazione nella qualità del segnale dovuta a tale link rende necessario eseguire un certo numero di elaborazioni preliminari direttamente nel front-end posto sul terminale mobile ed inviare i risultati ad un back-end remoto per successivi processamenti e per il riconoscimento vocale. Il DSR fornisce ad un sistema vocale per il controllo domotico notevoli vantaggi in termini di robustezza del riconoscimento rispetto all'impiego di un canale vocale convenzionale, dove la compressione dovuta al codec audio e gli errori sul canale di trasmissione influenzano negativamente l'accuratezza del riconoscimento. Il DSR, inoltre, garantisce intrinsecamente l'impiego di nuove interfacce multimodali permettendo l'invio di informazioni collaterali simultaneamente alle features del segnale vocale utilizzando lo stesso canale di comunicazione. In questo capitolo, verranno illustrate le tecniche e gli standard esistenti riguardanti i sistemi DSR, fino all'implementazione di un sistema di riconoscimento vocale distribuito robusto sia a livello di infrastruttura sia a livello di improvement del rate del parlato continuo anche tramite l'ausilio di modelli "garbage" che permettono di distinguere i comandi dal parlato continuo. Al fine di integrare il sistema di controllo vocale

con tutte le possibili aree di automazione della casa (climatizzazione, apparati, sicurezza, comunicazione e informazione) sono state sviluppate interfacce lato client come soluzioni versatili che permettano una gestione centralizzata di tutte le aree domotiche di afferenza e che siano in grado di interoperare con esse.

Un requisito essenziale di una interfaccia vocale per l'home automation è rappresentata dalla capacità di riconoscere più parlatori. A tal fine, in questo capitolo, verranno illustrate tecniche di speaker adaptation, per adattare il modello acustico ad un qualsiasi parlatore [10, 11]. In particolare, saranno descritti i risultati ottenuti dall'applicazione di queste tecniche sia al lato back-end che al lato front-end dell'elaborazione.

Un altro problema affrontato nell'ambito dello speech recognition è legato all'utilizzo del modello linguistico per l'estrazione di parole dal flusso di fonemi[1]. L'utilizzo dei modelli linguistici richiede una larga quantità di materiale di training. Per questo motivo, un modello linguistico N-gram è usato con successo quando si ha a disposizione un ampio data-set di training, mentre fallisce in modo drammatico con un set limitato di training data. In questo ambito verrà presentata una tecnica denominata "dictionary-based large vocabulary speech recognition" (DB-LVCSR), nella quale il task del decoder è quello di determinare la sequenza più probabile di fonemi, anziché ricercare la sequenza di parole che massimizza il prodotto di modelli acustici e linguistici, come di solito avviene negli attuali sistemi ASR.

[1]Un fonema è una unità linguistica dotata di valore distintivo, ossia una unità che può produrre variazioni di significato se scambiata con un'altra unità: ad esempio, la differenza di significato tra l'italiano tetto e detto è il risultato dello scambio tra il fonema /t/ e il fonema /d/. In ciascuna lingua, le lettere dell'alfabeto sono la rappresentazione o trascrizione grafica di suoni (detti foni). I foni individuano la "qualità sonora delle parole", ma in quanto permettono di distinguere una parola da ogni altra essi hanno un carattere astratto e una funzione distintiva. Intesi in tal modo sono indicati come "fonemi". L'analisi del segnale vocale è iniziata con lo studio dell'apparato fonatorio umano e con la classificazione dei suoni, che tale apparato (tratto vocale) può emettere, in categorie astratte chiamate fonemi. A ogni fonema corrispondono un particolare assetto del tratto vocale e una particolare modalità di emissione del suono. Essendo le caratteristiche dell'apparato vocale umano indipendenti dalla lingua dei parlanti, l'insieme di tutti i fonemi classificati costituisce un alfabeto fonetico internazionale (IPA, International Phonetic Alphabet). Ogni lingua ne utilizza un particolare sottoinsieme.

3.1 Cenni storici

I primi tentativi di riconoscimento vocale vennero effettuati negli anni 50 negli Stati Uniti, con lo scopo di realizzare sistemi controllabili con la voce. I maggiori finanziatori della ricerca in questo campo furono la National Security Agency (NSA) e il dipartimento della difesa Defense Advanced Research Projects Agency (DARPA). Nel 1952 i Bell Laboratories svilupparono un sistema in grado di riconoscere i numeri da 0 a 9. Il sistema poteva riconoscere solo parole singole: solo negli anni 70, presso la Carnegie Mellon University venne messo a punto un prototipo con il quale era possibile riconoscere frasi complete, ma con un dizionario e una struttura grammaticale limitati. La potenza di calcolo richiesta per elaborare il riconoscimento era prodigiosa, prevedeva l'utilizzo contemporaneo di 50 computer.

Negli anni 80 comparvero i primi dispositivi commerciali per il riconoscimento vocale. Nel 1982 la Covox commercializzò il Voice Master per Commodore 64 e successivamente per PC, in grado di realizzare una rudimentale sintesi vocale e un riconoscimento a parola singola in base ad un dizionario ristretto. Nel 1982 Dragon Systems iniziò a produrre software per il riconoscimento vocale, seguita da IBM e Kurzweil. Da allora il mercato è stato invaso di applicativi, spesso venduti con un microfono a corredo, in grado di trasformare il proprio PC in una "stazione vocale".

Negli anni novanta i PC conquistarono la "multimedialità" che ha trasformato il personal computer in uno strumento versatile in grado di riprodurre CD e DVD ed elaborare segnali audio/video. Anche i notebook dagli anni novanta in poi diventano multimediali, offrendo prestazioni analoghe ai modelli "da scrivania" (desktop). Lo sviluppo degli algoritmi per realizzare applicazioni di riconoscimento vocale su PC ebbe una grande accelerazione proprio negli anni novanta.

Oggi gli impieghi di questa tecnologia sono molteplici: si può comandare con la voce il proprio PC, il telefono cellulare, o il computer di bordo di un auto. Il forte sviluppo della ricerca in questo campo ha permesso di realizzare software anche per il mercato consumer. Con questi programmi, trascorso un periodo di addestramento sulla voce dell'utente, si può dettare un testo parlando in modo naturale. La precisione di riconoscimento di questi software è del 95 e 98%.

Più di recente, la ricerca ha beneficiato di progressi importanti nel campo del deep learning e big data. I progressi sono evidenziati dall'adozione di queste

tecniche da parte delle maggiori industrie del settore (Google, Microsoft, Hewlett Packard Enterprise, IBM, Nuance, Apple, Amazon, Baidu) nella progettazione ed implementazione dei propri sistemi di riconoscimento vocale.

3.2 Sistemi di riconoscimento vocale automatico (ASR)

Il riconoscimento vocale è il sotto-campo interdisciplinare della linguistica computazionale che integra conoscenze da diverse aree [38], quali processamento dei segnali, fisica (acustica), riconoscimento di patterns, teoria dell'informazione e comunicazioni, linguistica e informatica, al fine di sviluppare metodologie e tecnologie che consentano il riconoscimento e la traduzione del linguaggio parlato in testo per mezzo di computer e dispositivi computerizzati appartenenti al campo delle smart technologies e della robotica. È noto anche come riconoscimento vocale automatico "automatic speech recognition" (ASR), "computer speech recognition", o semplicemente "speech to text" (STT).

Il riconoscimento vocale automatico può essere definito come la trascrizione automatica in real-time di linguaggio parlato in testo leggibile [39], è il processo mediante il quale il linguaggio orale umano viene riconosciuto e successivamente elaborato attraverso un computer o più specificatamente attraverso un apposito sistema di riconoscimento vocale.

L'obiettivo di realizzare una macchina in grado di capire fluentemente il parlato, ha guidato la ricerca per più di 50 anni e sebbene questo obiettivo non sia stato ancora pienamente raggiunto, questa tecnologia è utilizzata correntemente da un largo numero di applicazioni e servizi.

L'obiettivo finale della ricerca nel campo ASR è quello di consentire ad un computer di riconoscere in tempo reale, con una precisione del 100%, tutte le parole che sono intelligibili pronunciate da qualsiasi persona, indipendentemente dalle dimensioni del vocabolario, il rumore, le caratteristiche del parlatore o dell'accento. Attualmente, se il sistema è addestrato per imparare la voce di uno singolo speaker, la precisione può essere superiore al 90%. I sistemi attualmente in commercio, sono in grado, con un breve periodo di addestramento, di catturare il parlato continuo con un ampio vocabolario ed una accuratezza molto alta (intorno 98%), sotto "ottime condizioni". Per "ottime condizioni" si intende: utenti che

abbiano caratteristiche vocali che corrispondono ai training data ed ambiente privo di rumore. Questo spiega perché, in certe condizioni, il rate si abbassa drasticamente rispetto al valore atteso. L'obiettivo di un sistema ASR è quindi quello di convertire il segnale vocale in testo scritto con elevata accuratezza, indipendentemente dal parlatore, dall'ambiente e dal dispositivo utilizzato per l'acquisizione del segnale.

Le principali problematiche sono stati classificati in sei punti [40]:

- Continuità. Nel linguaggio naturale non ci sono delle sospensioni tra le unità, inoltre le pause ed i silenzi possono essere molto brevi e mascherati da rumore di sottofondo.

- Dipendenza dal Contesto. Ogni suono in cui si può dividere il parlato (fonema) è modificato dal contesto in cui si trova. La produzione del fenomeno di coarticolazione è legata alla vicinanza dei fonemi; due fonemi uno precedente ed uno successivo possono modificare l'aspetto di un fonema adiacente.

- Variabilità. La variabilità può essere intra-parlatore oppure inter-parlatore. La variabilità intra-parlatore è prodotta per le modifiche introdotte dallo stesso parlatore (ad esempio, quando si pronunciano in maniera diversa gli stessi fonemi). La variazione di tipo inter-parlatore è prodotta dall'interazione del parlatore con l'ambiente; il segnale ottenuto dipenderà dai dispositivi usati per la sua registrazione.

- Salvataggio dell'informazione. Il numero di dati processati e salvati cresce notevolmente all'aumentare della durata dell'eloquio, perciò è necessario avere a disposizione risorse di memoria che possano supportare queste esigenze.

- Organizzazione dell'informazione. Il segnale contiene informazione a diversi livelli di descrizione. Nel processo di analisi ci sono dati di tipo semantico, sintattico e fonetico. Dall'altra parte il segnale vocale contiene anche l'informazione che descrive il parlatore.

- Assenza delle regole di descrizione e ridondanza.

3.2.1 Caratteristiche

Un sistema di riconoscimento automatico avviene secondo le seguenti fasi:

- Conversione analogica/digitale del segnale.
- Trattamento o analisi del parlato (elaborazione front-end). In questa fase viene fatta un'analisi preliminare della voce. Durante questo stadio, si esamina il segnale vocale e si estraggono i parametri spettrali.
- Realizzazione del modello acustico (elaborazione back-end). Questa è la cosiddetta fase di "addestramento" o training. In questa fase avviene la classificazione delle unità fonetiche: i segmenti di voce già processati vengono classificati ed identificati con dei simboli fonetici. A volte si può associare una probabilità ai fonemi mediante una simbologia che permette di ampliare l'informazione trasmessa allo stadio seguente.
- Realizzazione del modello linguistico (elaborazione back-end). In quest'ultima fase si sfruttano le regole del parlato (ortografia, sintassi, prosodia) per codificare il messaggio contenuto nel segnale, con lo scopo di migliorare le performance del sistema.

Il processo di riconoscimento è eseguito da un software chiamato motore di riconoscimento vocale; in particolare se l'applicazione è orientata al riconoscimento di comandi, viene detta "command-control", se orientata invece al riconoscimento di testo, l'applicazione è di dettatura. Nelle applicazioni di riconoscimento, il silenzio (unvoiced) è un elemento molto importante, al pari del segnale voiced, perché delimita l'inizio e la fine di una enunciazione o utterance [2]. Le utterances sono inviate al motore di riconoscimento sotto forma di features per il loro trattamento; questo motore utilizza un insieme di dati, modelli statistici ed algoritmi per convertire il segnale vocale in testo. Nelle applicazione "command-control" è opportuno definire l'insieme delle parole e delle frasi che gli utenti possono pronunciare, ossia è consigliabile, per migliorare le prestazioni del riconoscitore, definire una grammatica, un gruppo di regole per specificare parole e frasi che possono essere riconosciute dal motore. La specifica della grammatica può essere fatta semplicemente mediante un elenco di parole, cioè sfruttando le proprietà

[2] Una utterance è un flusso di parlato fra due intervalli di silenzio. Pertanto può essere una singola sillaba, una singola parola oppure una serie di parole che formano una frase.

del linguaggio naturale. Il World Wide Web Consortium (W3C) ha definito degli standard per le tecnologie vocali, che sono il VoiceXML e il CCXML. Per la specifica di grammatiche vocali ha introdotto la Speech Recognition Grammar Specification.

I sistemi di riconoscimento vocale, si dividono in due categorie: *Speaker Dependent* e *Speaker Independent*:

- *Speaker Dependent*: in questo caso il modello vocale viene adattato alla voce dell'utente. In pratica durante la fase di installazione, viene chiesto all'utente di leggere un testo con voce e velocità naturali. Il sistema si adatta così alle caratteristiche della voce e della pronuncia dell'utilizzatore. Questi sistemi offrono i migliori risultati in termini di precisione. I sistemi speaker dependent possono riconoscere correttamente oltre cento parole al minuto, confrontando quello che viene detto con un vocabolario di almeno 200.000 lemmi. Grazie al training sul parlatore, un normale PC è in grado di effettuare questa operazione in tempo reale, in background, e consentire all'utente di dettare un testo, estendendo le possibilità degli applicativi di acquisizione e trattamento testi.

- *Speaker Independent*: permettono il riconoscimento di un parlato generico, senza essere legati ad un determinato timbro di voce. Ogni individuo ha un proprio timbro vocale e un modo diverso di pronunciare le parole; la precisione di questi sistemi è inferiore quindi rispetto a quelli dipendenti dal parlatore, non disponendo del modello vocale del parlatore. I sistemi speaker independent offrono buoni risultati in situazioni in cui quello che viene detto dall'utente fa parte di una ristretta lista di parole oppure è prevedibile, come nel caso di risposte a scelta multipla. Per aumentare la precisione è necessario "insegnare" al sistema tutti i modi diversi in cui ogni singola parola può essere pronunciata. In pratica non potendo effettuare il training sul parlatore, la complessità si sposta verso il database che diventa molto grande e oneroso da costruire. Devono essere elaborate molte migliaia di ore di materiale audio con parole note, pronunciate da persone diverse. Ovviamente il riconoscimento speaker independent richiede un computer estremamente potente oppure una elaborazione off-line che può comportare, per una singola CPU, ore di elaborazione per riconoscere e trascrivere un minuto di audio.

Gli ASR si dividono in due classi: *"direct voice input" (DVI)* e *"large vocabulary continuos speech recognition" (LVCSR)*:

- *DVI* o VIC (voice input control): sistema di interazione uomo-macchina nel quale l'utente utilizza i comandi vocali per impartire istruzioni alla macchina. Utilizzato nelle applicazioni "command-control".
- *LVCSR*: sistema orientato al riconoscimento del parlato continuo ed applicazioni di dettatura.

3.2.2 Algoritmi

3.2.2.1 Hidden Markov model (HMM)

I riconoscitori vocali sono basati sui modelli di Markov nascosti [41]. Gli Hidden Markov Models (HMM) permettono di modellare processi stocastici come il parlato. È stata la prima tecnica utilizzata per il riconoscimento vocale e ancor oggi è ampiamente utilizzata.

Gli HMM rappresentano il parlato come una sequenza di vettori di osservazione derivati da una funzione di probabilità di primo ordine chiamata "catena di Markov". Gli stati del sistema sono identificati in uscita da una funzione probabilistica che descrive le variazioni e sono collegati mediante "transizioni" probabilistiche. Si tratta di comuni catene markoviane [42], con la solo differenza che gli stati non sono direttamente osservabili, da cui il termine "nascosti", in particolare:

- la catena ha un certo numero di stati;
- ogni stato genera un evento con una certa distribuzione di probabilità che dipende unicamente dallo stato attuale;
- l'evento è osservabile ma lo stato no.

I modelli HMM, che sono processi doppiamente stocastici (infatti sono previste due probabilità: una di transizione tra stati della catena e una distribuzione di probabilità tipica di ogni singola unità), descrivono le variazioni intrinseche del segnale vocale (e delle sue features), dando vita ad un modello statistico del parlato. I modelli di Markov nascosti restituiscono una misura probabilistica che sfrutta una catena markoviana per rappresentare la struttura linguistica del

parlato e un set di distribuzioni di probabilità per tenere in considerazione la variabilità del suono che dà luogo a quella particolare parola. Dato un insieme di termini noti, in grado di approssimare tutte le possibili variazioni delle parole di interesse, necessarie per l'addestramento (cioè il training set), è possibile determinare il miglior modello (set di parametri) che identifica una ben precisa parola o frase (in generale, una utterance) del vocabolario; questo modello verrà poi impiegato in fase di riconoscimento per valutare la verosimiglianza di un suono sconosciuto (segnale in ingresso) con tutti gli elementi del training set, consentendo quindi di determinare quale sia la realizzazione che meglio rappresenta il suono stesso.

3.2.2.2 Reti neurali

Una rete neurale artificiale (Artificial Neural Network - ANN) è un modello computazionale parallelo, costituito da numerose unità elaborative omogenee, raggruppate in differenti livelli, fortemente interconnesse mediante collegamenti di diversa intensità. Ogni neurone artificiale esegue la somma degli input pesati con il valore delle interconnessioni, per poi effettuare una trasformazione con una funzione spesso non lineare. Uno degli aspetti più interessanti è che una rete neurale non viene progettata per compiere una particolare attività, ma essa dipende dal particolare algoritmo di apprendimento automatico adottato. L'aspetto più importante e difficile da affrontare consiste nell'addestramento della rete: nel cercare cioè i pesi e i bias che le permettano di essere il più possibile aderente al fenomeno che deve modellare. Senza questa fase, la rete non potrebbe essere utilizzata per il riconoscimento. A tale scopo esistono degli appositi algoritmi, di cui il più importante è l'algoritmo di back-propagation. Per le ANN esistono due possibili tipi di training: supervised learning (addestramento supervisionato) e unsupervised learning (addestramento non supervisionato).

Negli ultimi 30 anni, l'approccio GMM-HMM è stato lo standard de facto nella modellazione acustica [43]. Il predominio del metodo GMM-HMM nella modellazione acustica ha, nel tempo, portato a tecniche di processing ed adattamento specificatamente ottimizzate per migliorare le performance con questo approccio. Negli ultimi anni, gli sviluppi nel campo del machine learning, hanno portato al concetto di deep neural networks (DNNs). L'utilizzo delle DNNs con successo nell'ambito vocale è stato dimostrato per la prima volta in [44]. Ci sono diverse motivazioni che hanno portato il Deep Learning ad essere stato sviluppato e po-

sto al centro dell'attenzione nell'ambito del Machine Learning solo negli ultimi decenni. Una di queste ragioni, forse la principale, è rappresentata sicuramente dal progresso nell'hardware, con la disponibilità di nuove unità di elaborazione, quali le graphics processing unit (GPUs), che hanno ridotto notevolmente i tempi di addestramento delle reti. Un' altra ragione è sicuramente la sempre crescente facilità di reperire dataset sempre più numerosi sui quali addestrare un sistema, necessari per addestrare architetture di una certa profondità e con un'alta dimensionalità dei dati di input. Il Deep Learning consiste in un insieme di metodi che permettono ad un sistema di ottenere una rappresentazione dei dati di tipo gerarchico, su molteplici livelli.

3.3 Sistemi di riconoscimento vocale distribuito (DSR)

Un sistema di riconoscimento vocale distribuito, fornisce ad un sistema vocale per il controllo domotico notevoli vantaggi in termini di robustezza del riconoscimento rispetto all'impiego di una canale vocale convenzionale, dove la compressione dovuta ai codec audio e gli errori nel canale di trasmissione influenzano negativamente l'accuratezza del riconoscimento. Un sistema DSR risolve i problemi indicati, mediante l'eliminazione della connessione voce e la sua sostituzione con una connessione dati a pacchetto per l'invio di una rappresentazione parametrica della voce ottimizzata per il riconoscimento e non soggetta alle limitazioni e alle degradazioni causate dai codec vocali, il cui scopo è la sola compressione massima di banda. In un sistema distribuito, il riconoscimento è ripartito tra il terminale e la rete. Il terminale (front-end) esegue l'identificazione degli intervalli di voce o di rumore ed il calcolo di parametri rappresentativi del parlato (features), che sono trasmessi tramite un canale dati ed arrivano al server per il riconoscimento automatico (back-end).

Per facilitare lo sviluppo delle applicazioni che supportano il riconoscimento distribuito è stato proposto uno standard che garantisce la compatibilità tra il terminale (client) e il riconoscitore remoto (server).

3.3.1 Standard ETSI ES 202-212

Il primo standard ETSI [45] è basato sulla rappresentazione tramite Mel-Cepstral coefficients (MFCCs), data la diffusione di questa tecnica nell'industria del riconoscimento vocale. Il secondo standard [46], precisa gli aspetti per la costruzione di un DSR resistente al rumore di sottofondo. Lo standard preso in esame in questa tesi è lo standard ETSI ES 202 212 [47].

La Fig. 3.1 mostra l'architettura generale di un sistema DSR secondo lo standard.

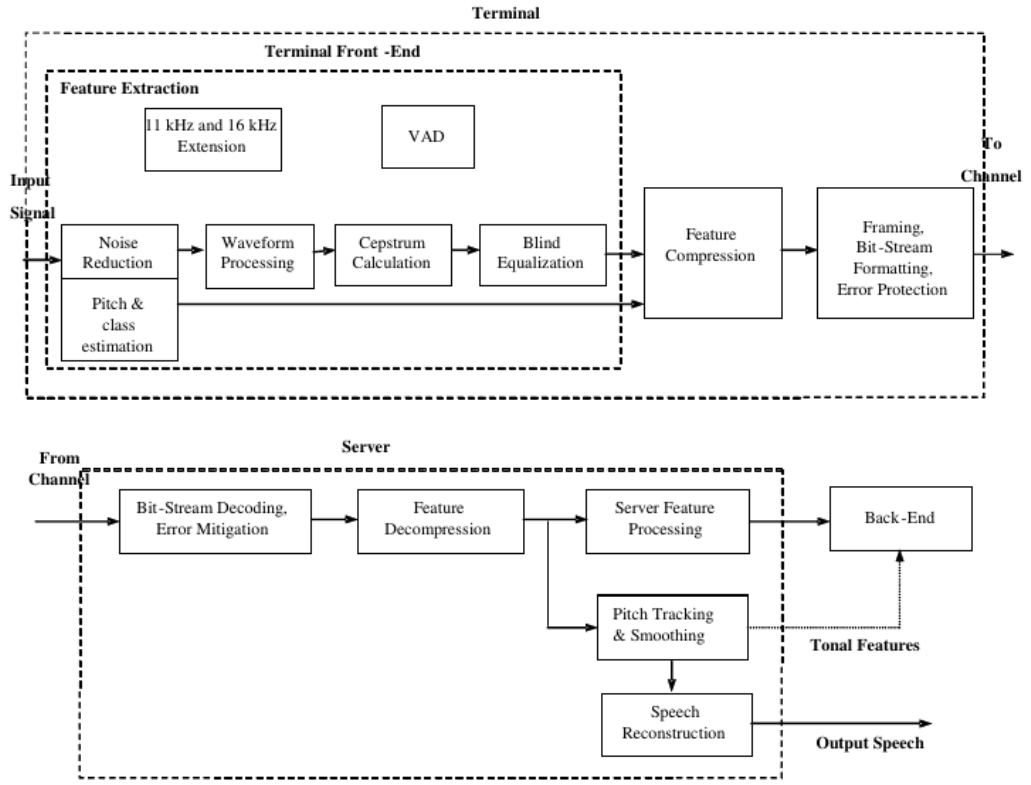


Figura 3.1: Schema a blocchi di un sistema DSR. (a) raffigura il lato client. (b) raffigura il lato server.

In accordo con le specifiche dello standard ETSI ES 202 212, un DSR è composto da un terminale mobile, o front-end, e da un server o back-end. Il front-end si occupa di estrarre le features della voce (coefficiente dell'energia c(0) + 12 coefficienti cepstrali statici), implementando il denoising del segnale (Fig. 3.2) ed il calcolo dei coefficienti cepstrali (Fig. 3.3). Nel terminale, il segnale del parlato

è campionato e parametrizzato usando l'algoritmo Mel-Cepstrum: come risultato sono generati 12 coefficienti cepstrali ed il valore dell'energia (log energy), che vengono compressi e formattati per comporre il bitstream di trasmissione. Il bitstream è inviato su una linea di trasmissione di tipo cablata oppure senza fili al server remoto; dal lato server si esegue il processo di decodifica estraendo i parametri Mel-Cepstrum. La definizione dell'architettura per il server di riconoscimento back-end non è parte dallo standard, in quanto l'interoperabilità tra il terminale e la rete è garantita dalla definizione del solo bitstream trasmesso. I canali usati per trasportare il bitstream sono sensibili agli errori, per questo motivo nel codificatore (terminale mobile) sono aggiunti alcuni bit di codice correzione (CRC) ed è usato un algoritmo per l'identificazione e la correzione degli errori nel decodificatore.

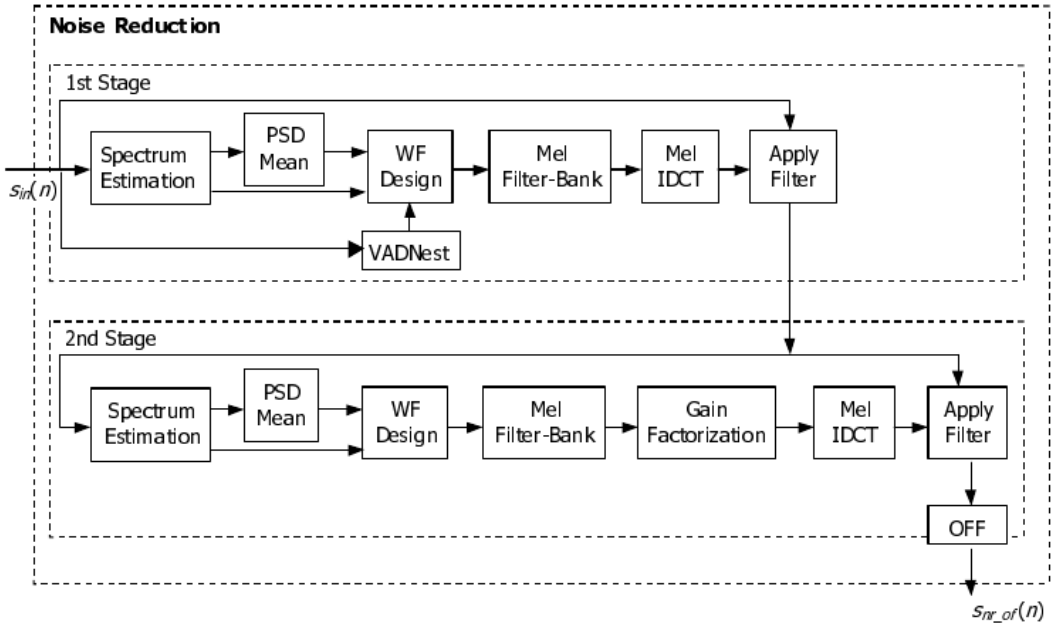


Figura 3.2: Schema a blocchi dell'algoritmo di noise reduction.

Il server svolge quattro operazioni: la ricostruzione del coefficiente dell'energia, l'estrazione dei parametri dai coefficienti cepstrali statici, la selezione dei vettori di parametri e infine il processamento di questi parametri per l'operazione di riconoscimento del parlato. L'energia ed i coefficienti cepstrali offrono delle informazioni sull'energia totale di tutta la banda di un frame. L'informazione

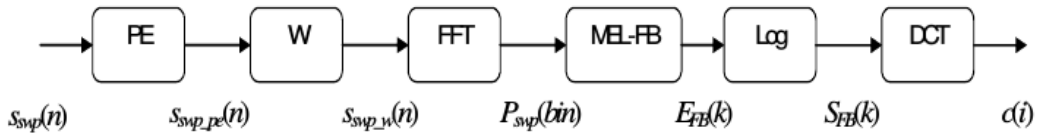


Figura 3.3: Schema a blocchi del calcolo dei coefficienti cepstrali.

aggiuntiva sulla dinamica dei parametri migliora la robustezza della rappresentazione del parlato, pertanto sono calcolate velocità ed accelerazione (delta e double delta) a partire dalle 13 caratteristiche statiche ricevute dal back-end. In conclusione sono usati 39 parametri per il riconoscimento. Negli ambienti rumorosi, i periodi lunghi di assenza di parlato del segnale incrementano il numero degli errori, causati dalla mancanza di corrispondenza tra le caratteristiche delle regioni di pausa-silenzio e il modello del silenzio. La soluzione a questo problema è l'eliminazione degli intervalli di pausa-rumore individuati dal Voice Activity Detector (VAD).

3.3.2 Frameworks per il riconoscimento vocale

La Tab. 3.1 mostra un elenco dei frameworks di riconoscimento vocale maggiormente utilizzati.

Tabella 3.1: Speech recognition software engine.

Application name	Description	Website	Open Source	License	Operating System	Programming Language	Supported Language
CMU Sphinx	HMM	CMU: Sourceforge	Yes	BSD style	Multi-platform	Java	English
HTK	HMM	HTK Web Site	No	HTK Specific License	Multi-platform	C	English
Julius	HMM trigrams	Julius Home page	Yes	BSD-like	Multi-platform	C	English
Kaldi	Deep neural net.	Kaldi Web Site	Yes	Apache	Multi-platform	C++	English

In questo lavoro di tesi, si farà riferimento al framework CMU Sphinx descritto di seguito. La caratteristica di essere indipendente dalla piattaforma e l'assenza di restrizioni sull'utilizzo unitamente alla possibilità di interagire attivamente con la community di sviluppo ed alla bontà dei modelli acustici ottenuti [48], rendono questo system engine molto competitivo.

3.3.2.1 CMU Sphinx

Il framework CMU Sphinx è stato creato attraverso una collaborazione tra Carneige Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), con il contributo dell'University of California at Santa Cruz (UCSC) ed il Massachusetts Institute of Technology (MIT). CMU Sphinx, chiamato anche brevemente Sphinx, è il termine generale per descrivere un insieme di sistemi di riconoscimento vocale open-source sviluppati presso la Carnegie Mellon University che include una serie di riconoscitori (Sphinx 2 - 4 & PocketSphinx), un acoustic model trainer (SphinxTrain), insieme a tools per la realizzazione del modello linguistico e dizionari. Sphinx si basa su HMM per determinare il path migliore attraverso i vincoli combinati di modello acustico, lessicale e linguistico, dato l'audio in ingresso. Di seguito una breve descrizione dei tools che lo Sphinx incorpora:

- **Sphinx**
  Sphinx è un sistema di riconoscimento vocale continuous-speech, speaker-independent che utilizza modelli acustici HMM ed un modello linguistico a n-grammi [49]. Sphinx è di solo interesse storico; esso è stato sostituito in termini di prestazioni dalle successive versioni.

- **Sphinx-2**
  Sphinx-2 si concentra sul riconoscimento in tempo reale adatto per essere integrato in applicazioni di riconoscimento vocale [50] . É utilizzato in sistemi di dialogo e sistemi di apprendimento delle lingue ed è stato incorporato in un certo numero di prodotti commerciali. Non è più in fase di sviluppo attivo. L'attuale sviluppo del real-time decoder si svolge nel progetto PocketSphinx.

- **Sphinx-3**
  Sphinx-3 è stato sviluppato prevalentemente per un riconoscimento vocale ad elevata precisione e non-real-time; recenti sviluppi hanno reso lo Sphinx-3 "vicino" al real-time ma non così tanto da renderlo utilizzabile in applicazioni interattive critiche. Sphinx-3 è ancora sotto sviluppo ed in congiunzione con il tool SphinxTrain permette l'accesso ad un largo numero di tecniche di modeling, come LDA/MLLT, MLLR e VTLN, per migliorare la recognition accuracy.

- **Sphinx-4**
  Sphinx-4 consiste nella completa "riscrittura" (in linguaggio di programmazione Java) dello Sphinx engine con l'obiettivo di fornire un framework più flessibile per la ricerca nell'ambito del riconoscimento vocale [51]. Sun Microsystems ha sostenuto lo sviluppo di Sphinx-4 e insieme a MERL, MIT e CMU. In Fig. 3.4 lo schema a blocchi dell'architettura del framework Sphinx-4. La maggior parte dei componenti del riconoscitore sono interfacce; le principali sono: il search manager, l'active list, lo scorer, il pruner, and search graph. Per quanto riguarda il search manager, l'utente specifica attraverso un file di configurazione in XML quale delle diverse implementazioni realizzate utilizzare. In questo file di configurazione l'utente può specificare anche altre opzioni (es. frequenza di campionamento). L'active list tiene traccia di tutti i percorsi correntemente attivi attraverso il grafo di ricerca per memorizzare l'ultimo token di ogni percorso, dove ogni token contiene l'informazione sulla probabilità del percorso in quel particolare punto della ricerca. Per effettuare il pruning è sufficiente potare i token nell'active list. Quando viene effettuato il processo di riconoscimento, il search manager chiede allo scorer la probabilità di ogni token nell'active list in previsione del futuro vettore di features estratto dal front-end. Infine il tool analizza tra tutti i percorsi che hanno raggiunto lo stato finale quello che ha la probabilità maggiore. Tale percorso rappresenta il risultato del riconoscimento.

- **Pocketsphinx**
  Pocketsphinx realizza il porting in C dello Sphinx, per consentirne un efficiente utilizzo nei sistemi embedded (basato su architettura ARM). È attualmente sotto sviluppo attivo ed incorpora features come fixed-point arithmetic ed algoritmi efficienti per la computazione GMM.

- **Sphinxbase**
  Libreria di supporto richiesta dal Pocketsphinx.

- **Sphinxtrain**
  Strumento per il training del modello acustico [52] come mostrato in Fig. 3.5. Il trainer, per estrarre le informazioni statistiche che determineranno il modello acustico, richiede in ingresso un database, chiamato training database, di campioni audio del segnale vocale. Il trainer deve

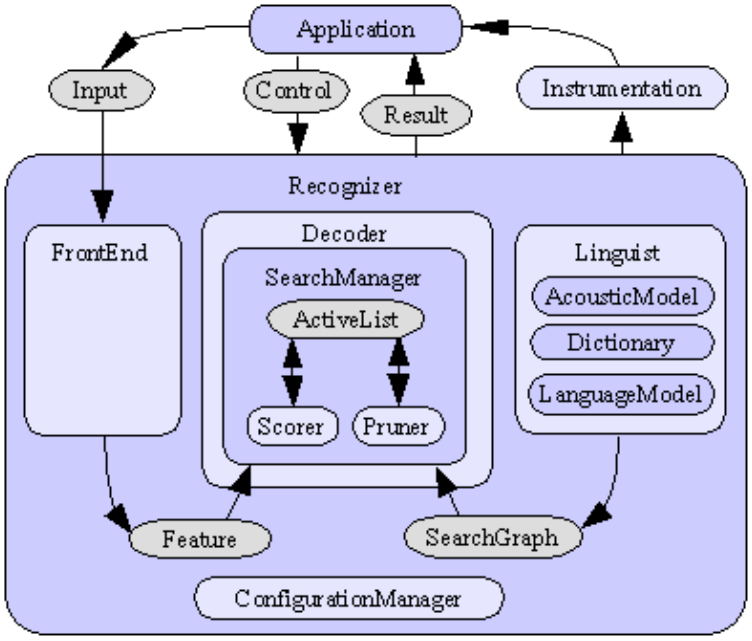


Figura 3.4: Architettura dello Sphinx-4.

essere informato delle unità sonore di cui si vuole imparare i parametri. Queste informazioni sono fornite al trainer attraverso un file denominato "transcript file", nel quale le sequenze di parole e non-speech sounds sono scritte esattamente come si sono verificate nel segnale vocale, seguite da una variabile che può essere utilizzata per associare questa sequenza con il segnale vocale corrispondente. A questo punto il trainer cerca nel dizionario, che mappa ogni parola in una sequenza di unità sonore, per derivarne la sequenza associata ad ogni segnale.

3.4 Implementazione di un sistema DSR per il controllo domotico

In questo capitolo verrà analizzato lo sviluppo di un sistema di riconoscimento vocale distribuito nel quale la ricostruzione del messaggio trasmesso è basata sull'elaborazione dello stream di features. Il sistema proposto è stato sviluppato in [53] e di seguito se ne riportano caratteristiche e risultati ottenuti.

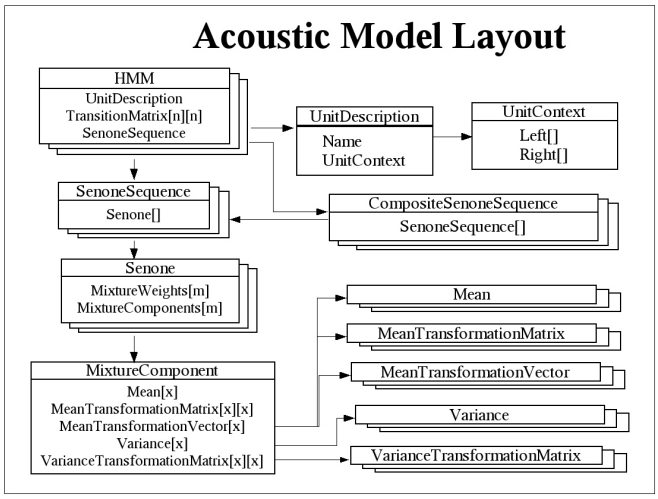


Figura 3.5: Architettura del tool Sphinxtrain.

In Fig. 3.6 è mostrato uno schema ad alto livello di un sistema di riconoscimento vocale distribuito (DSR).

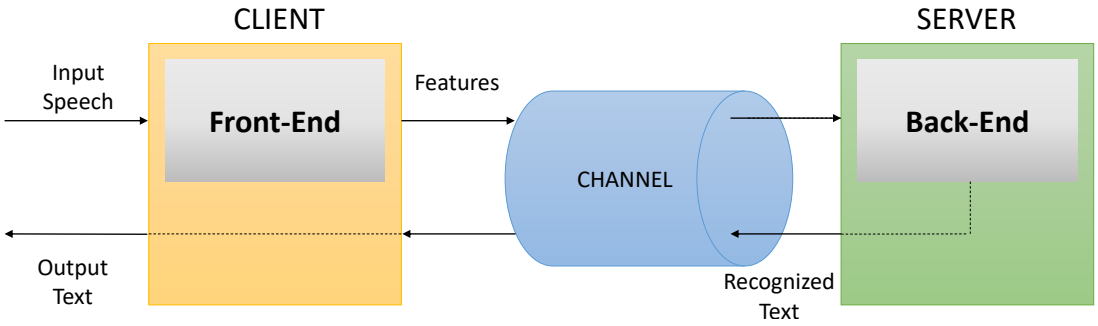


Figura 3.6: Schematizzazione ad alto livello di un sistema DSR.

L'architettura di un sistema DSR, come mostrato in Fig. 3.6, è riconducibile ad uno schema di tipo client-server [54]. L'idea base è quella di distribuire l'elaborazione tra front-end e back-end. Il front-end (FE), ospitato nel client, elabora l'audio in ingresso estraendone una rappresentazione parametrica codificata (features). Le features estratte vengono trasmesse attraverso un canale dati al back-end remoto (BE). L'onere computazionale maggiore è destinato al back-end, in quanto è responsabile dei successivi processamenti ed elaborazione delle features al termine dei quali invia al front-end il testo riconosciuto.

Il sistema prevede una modalità di comunicazione HTTP tra front-end e back-end in modalità chunked-encoding[3]. In particolare è stato sviluppato un server web minimale in grado di gestire la comunicazione HTTP tra il front-end ed il back-end. Nel server web è stato implementato un canale in POST per l'invio delle features dal front-end al back-end, ed un canale di ritorno in GET per il feedback testuale inviato dal back-end al front-end.

In Fig. 3.7 è riportato lo schema a blocchi dettagliato del sistema DSR.

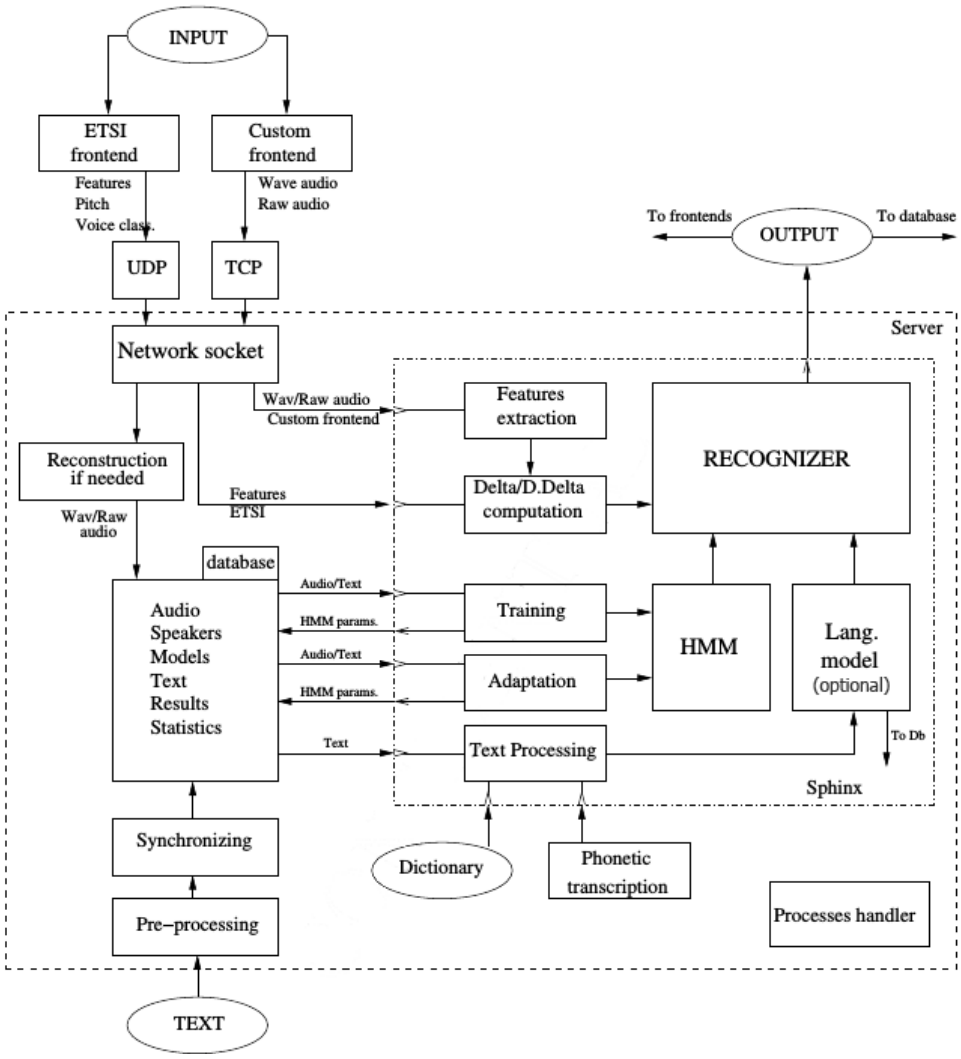


Figura 3.7: Schema a macro-blocchi del sistema DSR.

[3]modalità di trasferimento di dati in cui i dati stessi vengono inviati in una serie di chunks (pacchetti).

3.4.1 Case Study Architecture

Il sistema DSR realizzato consente il controllo vocale dell'illuminazione in un ambiente domestico o di lavoro. La voce infatti, come già detto, si presta a questo tipo di applicazioni, essendo uno dei mezzi più semplici da utilizzare per il controllo degli home automation systems [8, 4]. In questo sistema il back-end processa il segnale audio e lo trasforma in comandi testuali che posso essere inviati e processati dall'attuatore del sistema di illuminazione.

In questo tipo di configurazione, diverse sono le problematiche che ci si trova a dover affrontare:

- attivazione del sistema,
- speech capture: il sistema deve essere in grado di distinguere la voce dal rumore, dato che il microfono è sempre attivo, di riconoscere "distant speech", ed individuare errori di pronuncia. Problemi legati all'interpretazione dei comandi nascono poiché l'utente potrebbe non aderire perfettamente alla grammatica e la conversazione potrebbe essere confusa con comandi di controllo.
- eliminazione del testo che non corrisponde alla grammatica dei comandi,
- interpretazione ed attuazione dei comandi,
- sistema in ascolto continuo.

Il sistema proposto, si pone l'obiettivo di realizzare uno strumento di controllo che sia il meno invasivo possibile e di mantenere una buona precisione nel riconoscimento anche in condizioni di distant speech e background noise [55], come può accadere in una situazione reale di utilizzo. Per quanto riguarda il primo obiettivo, questo è stato raggiunto realizzando una configurazione ad-hoc che richiede al lato utente la semplice installazione di un piccolo ed economico front-end, dotato di un microfono panoramico e connessione a Internet. L'installazione non richiede altri dispositivi user side, dato che il processamento del segnale è demandato all'unità esterna come previsto dall'architettura del framework DSR [46]. Per realizzare il secondo obiettivo (robust ASR system) il sistema integra due semplici tecniche: un *i)* meccanismo adattativo di automatic voice activity detection (VAD) basato su soglia che abilita il sistema DSR solo quando è individuato uno spoken command, un *ii)* meccanismo che scarta utterances non

corrispondenti a comandi, background noise e suoni estranei. Questa seconda tecnica si basa sulla generazione di un *garbage model* [56], che include delle parole "esca" opportunamente collocate [57] aiutando il sistema ASR ad identificare le out-of-vocabulary words, consentendo così di eliminare le utterances che non corrispondono ai comandi.

Queste strategie, insieme a un protocollo ad hoc per ottimizzare la comunicazione client / server, hanno portato ad un sistema DSR robusto, come mostrano i risultati sperimentali.

La Fig. 3.8 mostra il sistema di illuminazione di riferimento [55]: è un sistema wireless per il controllo digitale delle luci, con una GUI di controllo su touch panel (DSR FE) ed un sistema di riconoscimento vocale (DSR BE); questo sistema realizza una rete wireless point-to-point costituita da diversi dispositivi DALI (control gears (CGs) connessi al DALI bus), ed include:

- un FE audio costituito da una low cost single-board computer (BeagleBoard® / Raspberry Pi®) dotata di un microfono USB standard e GUI realizzata, o uno smartphone con installata la custom app sviluppata per il controllo luci;

- uno o più DALI bridges. Ogni bridge può essere connesso ad uno o più CG attraverso il DALI bus;

- un touch panel DALI master per il controllo e la configurazione della rete di illuminazione. Il master è anche in grado di interoperare con differenti sistemi di home automation, inoltrando richieste e comandi al control bus del sistema domotico;

- una applicazione Android custom, che consente il controllo vocale dei punti luce nella rete domotica.

3.4.2 Front-End

Nel framework descritto, il front-end, ovvero il modulo client del sistema DSR [58], corrisponde all'estrattore di features, la cui struttura rispecchia quella dello standard ETSI ES 202-212 descritto in 3.3.1. Il front-end sviluppato in Java nella tesi di dottorato del Dott. Massimo Mercuri [59], cattura i campioni dalla

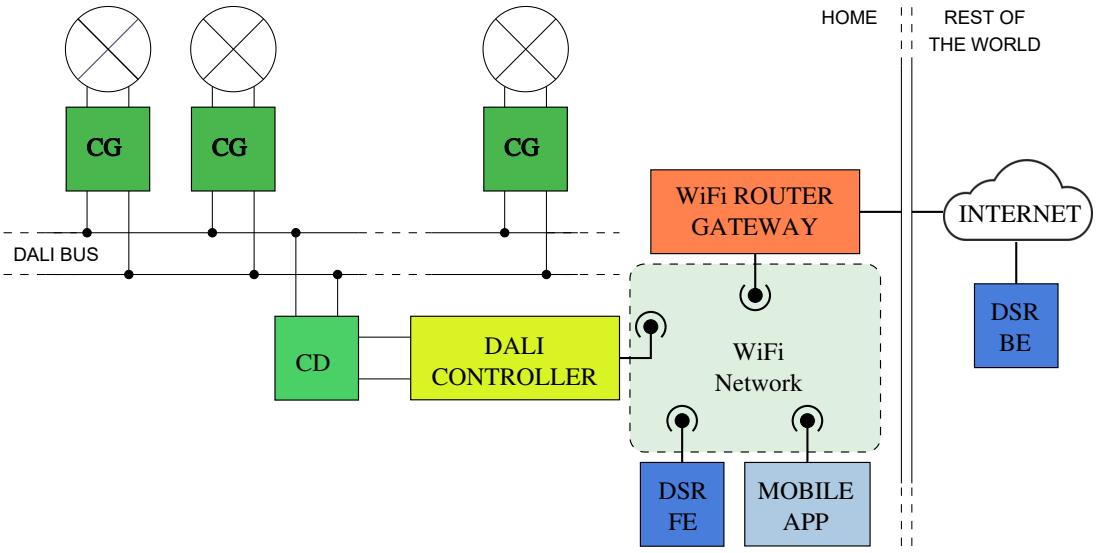


Figura 3.8: Sistema di controllo luci.

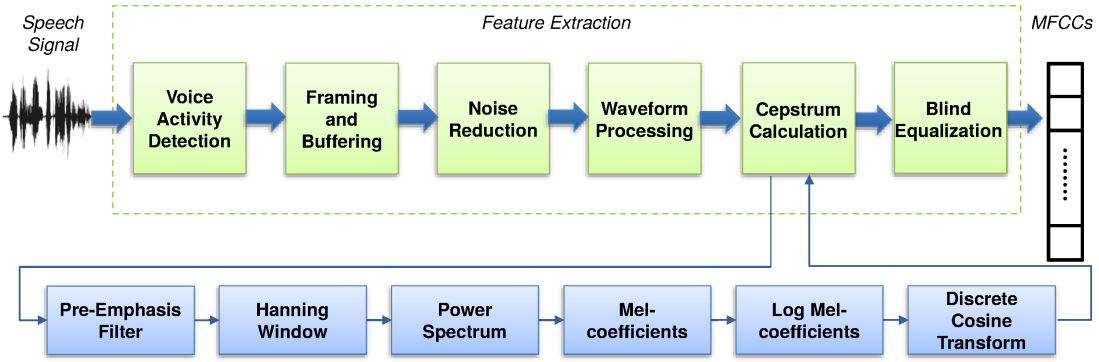


Figura 3.9: Schema dell'estrattore delle features conforme allo standard ETSI.

linea audio, li bufferizza ed estrae le features dall'audio catturato. Lo schema a blocchi dell'elaborazione è riportato in Fig. 3.9.

Lo standard descrive gli algoritmi per la computazione delle Mel-cepstral features (MFCCs), a partire da un segnale audio campionato a diversi rates (8 kHz, 11 kHz and 16 kHz). Il vettore delle features consiste in 12 coefficienti cepstrali ed un coefficiente che rappresenta la log-energy. In aggiunta, 26 dynamic features, cioè i coefficienti cepstrali delta e delta-delta, sono calcolate al lato back-end, per un vettore finale di 39 componenti. I coefficienti MFCC sono estratti da frames di 25 ms generati ogni 10 ms, quindi frame consecutivi di 25 ms overlappati di

15 ms. Prima di effettuare il calcolo dei coefficienti cepstrali, viene realizzata una noise reduction ed un successivo waveform processing. La blind equalization delle features risultanti è lo stadio finale dell'elaborazione lato front-end.

Il formato del frame previsto dallo standard ETSI non è però conforme al formato del frame accettato dallo Sphinx-4. Ciò viene messo in evidenza mediante le Fig. 3.10– 3.11. Per risolvere tale inconveniente è stata inserita, nell'implementazione in Java del front-end, una funzione che permetta di convertire il formato del frame ETSI in quello accettato dallo Sphinx.

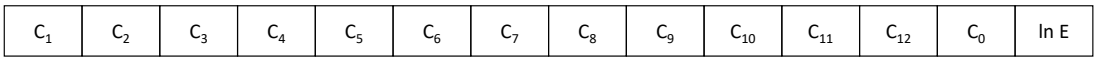


Figura 3.10: Formattazione del frame ETSI.

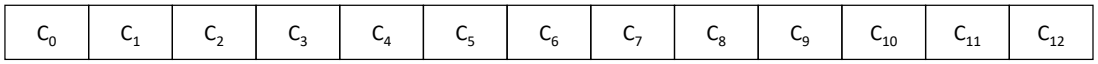


Figura 3.11: Formattazione del frame Sphinx.

In aggiunta, sono stati implementati algoritmi per la la stima del pitch e la classificazione dei frames (voiced/unvoiced) al fine di ridurre lo speaker mismatch e tools per l'elaborazione ed il coding/decoding di differenti formati di features.

3.4.3 Back-End

Il riconoscimento del messaggio vocale è realizzato lato BE, che come già detto, si appoggia alle Sphinx third-party libraries sviluppate dalla Carnegie Mellon University [51]. Gli steps del processo di riconoscimento sono:

- identificazione dei segmenti contenenti potenzialmente dei comandi e loro processamento (per ridurre rumore, riverbero e componenti estranee) fino alla loro compressione;
- processamento dei segmenti precedentemente identificati per estrarne i comandi che potrebbero contenere;
- interpretazione dei comandi per l'attuazione delle azioni corrispondenti.

Il BE restituisce quindi il testo corrispondente al comando pronunciato, se questo soddisfa le regole della grammatica stabilita. Sarà poi il dispositivo installato lato utente, a decidere quale azione deve corrispondere al comando riconosciuto.

Il framework illustrato dispone anche di un modulo, chiamato director, che funge da "assistente" alla generazione del modello acustico e linguistico, sempre servendosi degli strumenti forniti dallo Sphinx [52]. La Fig. 3.12 mostra tutte le operazioni gestite da questo modulo, nello specifico: estrazione delle features, generazione del modello acustico e linguistico, riconoscimento e testing.

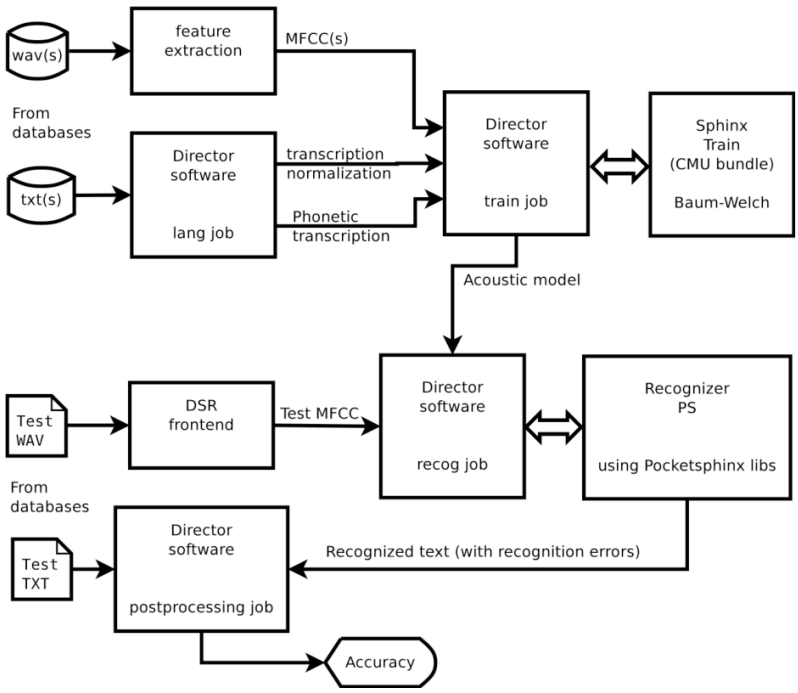


Figura 3.12: Director work-flow.

3.4.4 Protocollo di comunicazione

La Fig. 3.13 mostra lo schema della comunicazione tra le varie entità che costituiscono il sistema.

I dati sono scambiati tra l'audio FE ed il BE di riconoscimento. All'avvio del sistema avviene l'autenticazione dell'utente. Una volta che un comando è stato

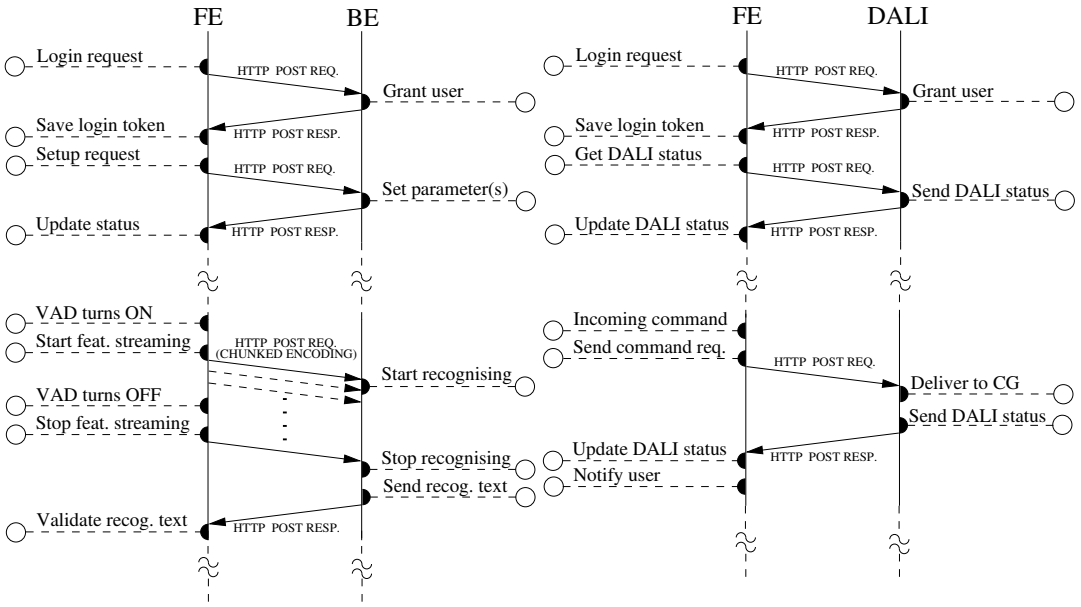


Figura 3.13: DSR-dialogue.

individuato e riconosciuto, avviene un nuovo scambio tra FE e controller (in questo caso la DALI command unit). A questo scopo, sono state implementate delle API ad-hoc che consentono all'utente di interagire con un sistema di riconoscimento in risposta agli ingressi incapsulati in un protocollo HTTP. La Tab. 3.2 elenca le API implementate.

Tabella 3.2: DSR HTTP API (low level API).

API FUNCTION	METHOD	DESCRIPTION
login	POST	obtains an access token (OTP) for authenticated requests
logout	POST	invalidates the current session on the server
echo	POST	tests the connection with the server
batch_adapt	POST	sends a file to the server in tar or compressed tar format
recog_frames	POST with query string	sends to the server a stream of frames for voice recognition
recog_utterance	POST with query string	sends the server a whole utterance for recognition
get_model_info	GET	obtains useful information on the acoustic model used

3.4.5 Interfacce per il controllo vocale

Integrare in un sistema domotico la funzionalità di riconoscimento vocale permette il controllo vocale delle apparecchiature domestiche e consente quindi di

aumentare il ventaglio di possibilità di interazione verso i dispositivi che usiamo quotidianamente. Questa integrazione può essere effettuata realizzando interfacce ad interazione vocale lato client sfruttando la tecnologia DSR. Per il raggiungimento di questo obiettivo, sono state implementate alcune interfacce ad interazione vocale lato client: una Web app che effettuata il riconoscimento on-line del parlato e un'interfaccia Android per smartphone che interagisce con il sistema wireless di controllo digitale della luce descritto precedentemente.

Web application La Fig. 3.14 mostra la schermata della web app di riconoscimento vocale realizzata secondo il protocollo di comunicazione e le APIs descritte nella sezione 3.4.4.

Android application Il front-end ETSI ES 202-212 precedentemente sviluppato in Java è stato portato su un dispositivo mobile smartphone, basato su piattaforma Android. L'applicazione Android si interfaccia con il sistema digitale per il controllo delle luci, con il sistema MyHome Bticino tramite protocollo OpenWebNet per il controllo domotico e con il sistema DSR per il controllo del sistema domotico tramite comandi vocali. Le Figs. 3.15, 3.16, 3.17 mostrano alcuni screenshots dell'applicazione.

3.4.6 Risultati sperimentali

Il set-up utilizzato per la fase sperimentale è il seguente:

- microfono USB panoramico connesso al FE e riconoscitore HMM-based in esecuzione sul BE;
- riconoscitore HMM-based in esecuzione sul BE, configurato per la lingua italiana e command-spotting;
- modello acustico addestrato per un generico large-vocabulary continuous speech recognition task [60] [61];
- modello linguistico grammar-based, con grammatica specificatamente definita per il controllo di un sistema di illuminazione.

L'abilità di command-spotting del sistema è raggiunta tramite l'implementazione di due differenti modelli garbage: *i)* phone loop-based generic word model

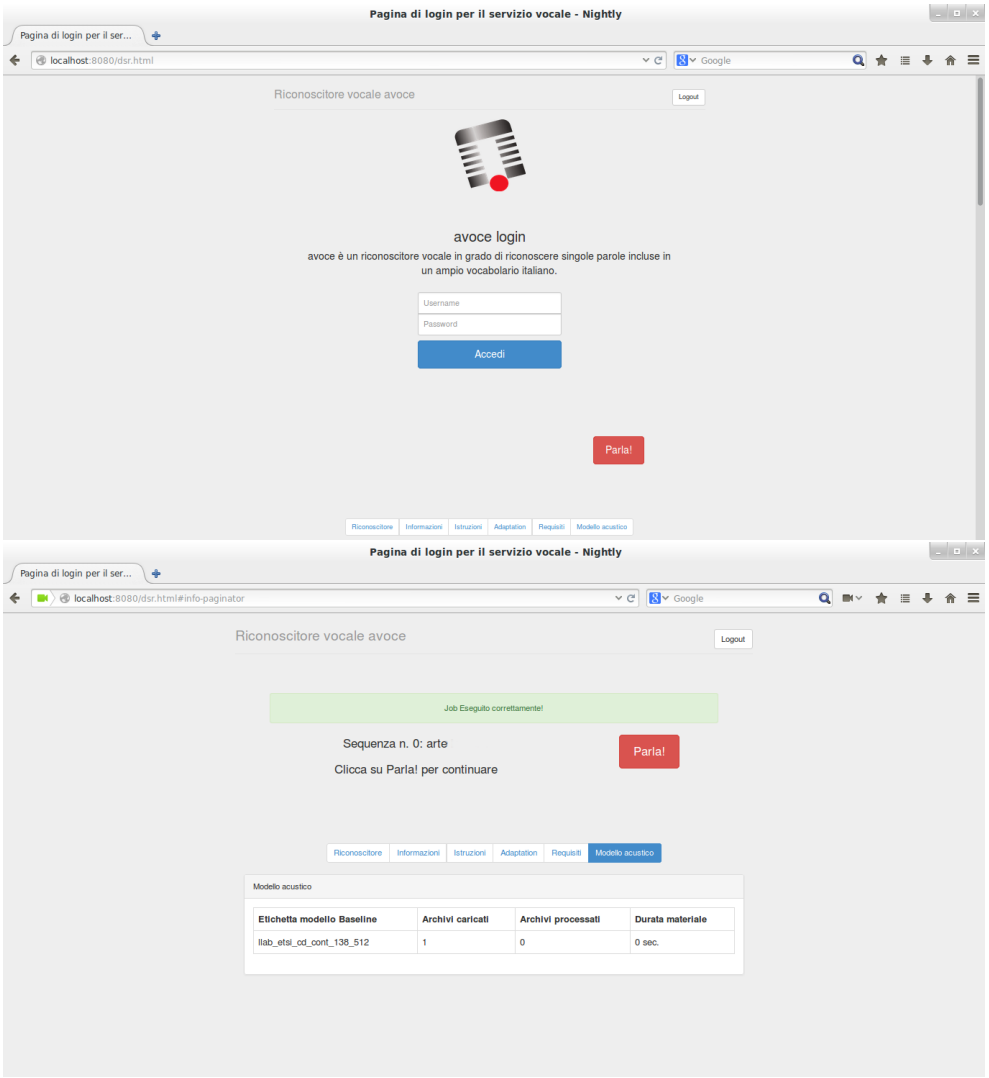


Figura 3.14: Web application.

[62], capace di catturare ogni sequenza di fonemi out-of-grammar (OOG) e *ii)* decoy-based garbage model [63] sempre finalizzato alla cattura delle sequenze OOG, che in modo semi-automatico trova le parole errate che l'ASR più spesso erroneamente sostituisce con parole corrette. Funziona iterativamente provando il motore ASR sulle parole pronunciate desiderate, ogni volta regolando la grammatica in modo che le parole che più probabilmente possono essere confuse con la parola target vengano identificate e rimosse.

I test sono stati condotti posizionando il microfono in un ufficio e registran-

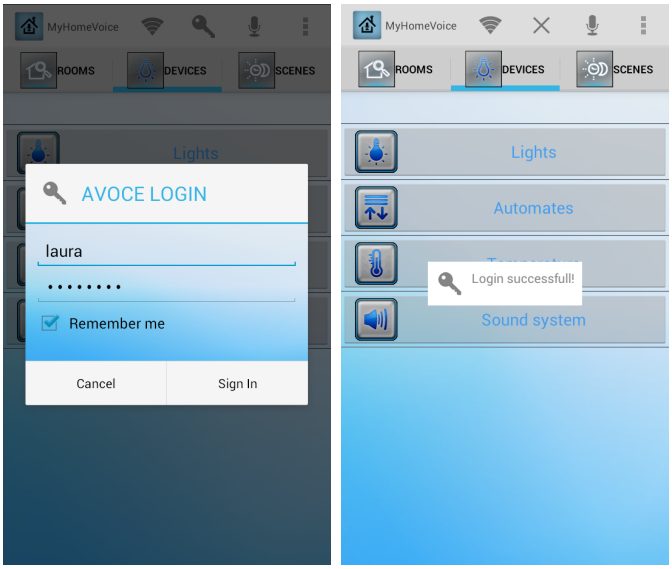


Figura 3.15: Android application: login.

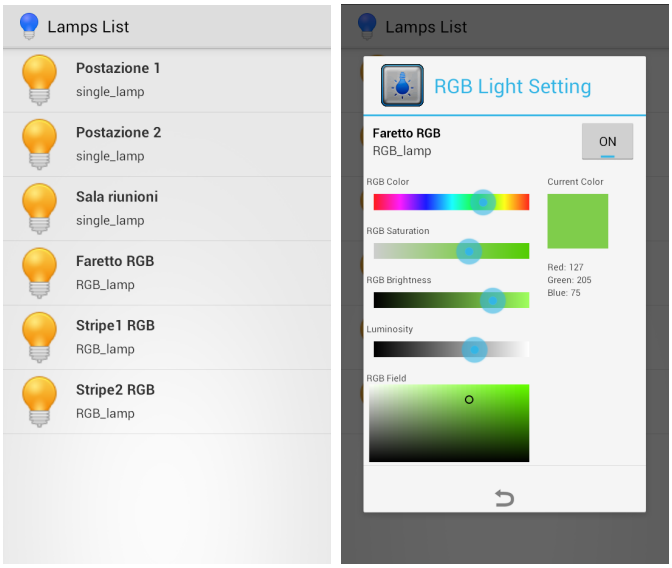


Figura 3.16: Android application: setting.

do diverse sessioni della durata di circa 30 minuti: nel primo tipo di sessione, viene registrata la conversazione che ha luogo tra due persone che si trovano a lavorare nel proprio ufficio, parlando in maniera completamente naturale (continuous speech), agendo, a bisogno, sul sistema di controllo luci secondo i comandi appartenenti alla grammatica; nel secondo tipo di sessione un singolo utente

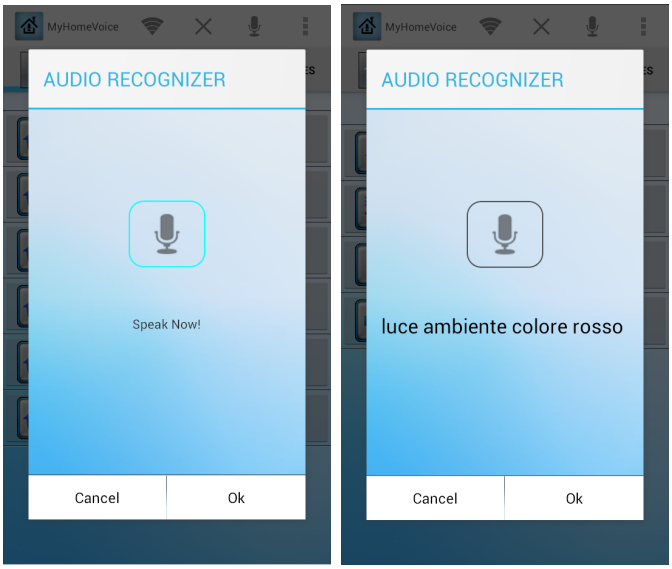


Figura 3.17: Android application: riconoscimento comandi.

ripete periodicamente i diversi comandi del sistema di controllo senza interporre parole diverse dalla grammatica dei comandi (chiameremo questa modalità round-robin). In particolare, il primo tipo di sessione è stato analizzato sia in modalità on-line che off-line per i seguenti tipi di VAD:

- threshold VAD: basato su soglia di energia adattiva,
- ETSI VAD: basato sullo standard ETSI.

Le performance del sistema sono state valutate analizzando i seguenti parametri:

- Sensitivity = TP / (TP + FN)
- Specificity = TN / (TN + FP)
- Precision = TP / (TP + FP)
- Accuracy = (TP + TN) / (TP + TN + FP + FN)

dove TP sono i veri positivi (elementi che appartengono alla grammatica e che sono riconosciuti come tali), FN sono i falsi negativi (elementi che appartengono alla grammatica e sono scartati), FP i falsi positivi (elementi che appartengono

al garbage e sono riconosciuti come appartenenti alla grammatica), TN i veri negativi (elementi che appartengono al garbage e che sono riconosciuti come tali). Si è scelto di valutare anche l'accuratezza del VAD secondo la sua capacità di segmentazione, intesa come il numero di utterances individuate correttamente dal VAD rispetto al numero totale di utterances pronunciate. Solo per il threshold VAD, le performance sono state espresse in funzione della soglia e per valutare la bontà del garbage model, è stato fatto variare il valore della OOG probability.

Di seguito i risultati ottenuti per i diversi tipi di sessione descritti in precedenza.

Continuous speech (off-line mode) Le Tabs. 3.3–3.4 mostrano i risultati ottenuti. Le performance migliorano con un valore della OOG probability maggiore di 0.1 e soglia uguale a 7; per questo motivo i test successivi manterranno questa configurazione.

Tabella 3.3: Threshold VAD performance per differenti soglie e valori della OOG probability, in off-line mode. Traccia audio in esame - Durata: 0h 36m 58s, rate: 8 kHz, SNR mean: 13.860 dB, SNR variance: 13.025 dB.

OOG prob	Threshold	Sensitivity [%]	Specificity [%]	Precision [%]	Accuracy [%]	VAD accuracy
0.001	5	82	60	92	78	372/418
	7	85	42	85	77	461/418
	10	67	80	95	69	431/418
	13	57	57	84	57	289/418
0.01	5	67	60	90	66	376/418
	7	64	60	92	81	463/418
	10	64	80	94	66	436/418
	13	50	66	87	53	290/418
0.1	5	64	50	85	61	381/418
	7	82	75	95	81	470/418
	10	60	80	94	63	437/418
	13	46	66	86	60	291/418
1	5	79	75	95	68	383/418
	7	82	75	95	81	471/418
	10	50	100	100	60	438/418
	13	50	60	87	51	292/418

Continuous speech (on-line mode) Risultati descritti in Tab. 3.5.

Tabella 3.4: ETSI VAD performance per diversi valori della OOG probability, in off-line mode. Traccia audio in esame - Durata: 0h 36m 58s, rate: 8 kHz, SNR mean: 13.860 dB, SNR variance: 13.025 dB.

OOG prob	Sensitivity [%]	Specificity [%]	Precision [%]	Accuracy [%]	VAD accuracy
0.001	82	66	92	79	383/418
0.01	71	100	100	75	390/418
0.1	85	66	92	87	397/418
1	75	80	95	75	403/418

Tabella 3.5: Threshold VAD performance in on-line mode. Traccia audio in esame - Durata: 0h 29m 26s, rate: 8 kHz, SNR mean: 19.673 dB, SNR variance: 22.580 dB.

OOG prob	Th	Sensitivity [%]	Specificity [%]	Precision [%]	Accuracy [%]	VAD accuracy
0.1	7	60	83	98	61	138/122

Round-robin (on-line mode) Risultati riportanti in Tab. 3.6. Naturalmente questo test non contiene i true negative checks; per questo motivo il valore della *specificity* è nullo.

Tabella 3.6: Round-robin performance. Traccia audio in esame - Durata: 0h 8m 8s, rate: 8 kHz, SNR mean: 17.918 dB, SNR variance: 15.409 dB.

OOG prob	Th	Sensitivity [%]	Specificity [%]	Precision [%]	Accuracy [%]	VAD accuracy
1	7	89	0	98	87	135/123

Conclusioni In conclusione, il sistema descritto, sebbene necessiti di ulteriori accorgimenti per un improvement del rate di riconoscimento, sembra essere già in grado di offrire una interfaccia vocale efficiente per il controllo dell'illuminazione.

3.5 Tecniche di adattamento per ASR robusto

Come già detto nella sezione 3.2.1, i sistemi di riconoscimento vocale, si dividono in due categorie: *speaker dependent* (SR) e *speaker independent* (SI). I modelli SI sono addestrati con dati di diversi parlatori e funzionano ragionevolmente bene quando vengono utilizzati per il riconoscimento del parlato di un

esteso range di parlatori diversi ma il rate di riconoscimento sul singolo parlatore non è paragonabile a quello ottenuto con un modello SR addestrato specificamente per un parlatore. Nonostante i progressi nello sviluppo di modelli SI, il tasso di errore che si ha nel riconoscimento vocale usando questo tipo di modello acustico è tipicamente 2-3 volte superiore rispetto al tasso di errore che si otterrebbe utilizzando nel riconoscimento vocale un modello acustico SR [64]. In questo caso, le performance che si ottengono nel riconoscimento del parlato sulla voce del medesimo parlatore sono notevolmente migliori rispetto a quelle che si ottengono con un modello SI ma le performance che si ottengono con altri parlatori diversi dal parlatore per cui il modello SD è stato addestrato sono generalmente insoddisfacenti.

Il problema è quindi quello di ottenere un modello SI ad elevate prestazioni o un modello che si "adatti" alle caratteristiche dei vari parlatori, senza che la quantità di materiale richiesto aumenti esponenzialmente in entrambi i casi.

Un metodo efficace usato in letteratura per ridurre il mismatch tra l'iniziale modello acustico SI ed il parlatore target è quello di utilizzare una piccola quantità di speech del nuovo parlatore (adaptation data) per fare il "tuning" del modello SI al nuovo parlatore [65]. Tale metodo è conosciuto come *speaker adaptation*. Le tecniche di adattamento al parlatore (o speaker adaptation) ricadono in due categorie: tecniche in cui si cerca di adattare la voce del nuovo parlatore a quella del parlatore con cui è stato addestrato il modello acustico esistente e tecniche in cui si cerca di migliorare la modellizzazione del nuovo parlatore adattando i parametri del modello acustico esistente.

Una delle più note tecniche di adattamento del modello acustico è la Maximum-Likelihood Linear Regression (MLLR) [66]. L'approccio MLLR prevede che l'iniziale modello acustico SI sia un HMM addestrato in precedenza. Il metodo MLLR aggiorna solamente i valori delle medie delle misture che compongono la distribuzione dello stato delle uscite del modello iniziale. Il fondamento logico alla base di questa trasformazione è che la differenza tra parlatori diversi si assume essere caratterizzata dai valori delle medie delle misture. Pertanto i valori delle medie delle misture del modello SI vengono trasformati per migliorare la modellazione del nuovo parlatore. La stima della trasformazione al nuovo parlatore viene effettuata facendo uso di un modello di regressione lineare. Non vengono adattate né la probabilità di transizione degli stati, né il peso delle misture, né le matrici di covarianza. Questi parametri mantengono i rispettivi valori propri

del modello SI.

3.6 Implementazione di tecniche di compensazione del mismatch tra parlatori

Un requisito essenziale di una interfaccia vocale per l'home automation è rappresentata dalla capacità di riconoscere più parlatori. Questo obiettivo, consiste nella ricerca di tecniche di speaker adaptation per adattare il modello acustico a qualsiasi parlatore, e tecniche basate sulla definizione di modelli acustici multi-speaker.

In prima analisi, l'attività di ricerca si è concentrata sull'applicazione di algoritmi di adaptation lato back-end, integrati anche nel sistema DSR che hanno portato ad un incremento del 10% del rate di riconoscimento.

Successivamente l'attività di ricerca si è spostata verso l'implementazione di tecniche di compensazione del mismatch tra parlatori lato front-end. Attualmente le tecniche più utilizzate per compensare tale mismatch in modelli multi-speaker sono quelle basate su MLLR, Maximum A Posteriori (MAP) ed Eigenvoice. Tali tecniche richiedono l'acquisizione di parlato più o meno lungo, richiedono altresì elaborazione server side per la ristima dei parametri del modello e spesso necessitano della trascrizione fonetica dell'audio utilizzato per l'adattamento. L'obiettivo è stato quello di realizzare un'adaptation lato front end, meno gravosa in termini computazionali ed in termini di materiale acustico necessario per l'elaborazione. L'adaptation viene realizzata individuando una trasformazione lineare che porti dallo spettro del parlatore A a quello del parlatore B per compensare il mismatch esistente tra i due parlatori; e questa trasformazione deve essere applicata prima del calcolo delle features. Le prestazioni confrontate con i risultati ottenuti dall'applicazione dell'adaptation lato back end non hanno portato improvement significativi del rate del riconoscimento. Dall'analisi della risultati sembra che un problema posso essere il fatto che la matrice dei parlatori viene calcolata mediando su tutto il segnale senza distinguere gli stati dei fonemi. Infatti le medie delle features dei singoli parlatori non presentano grandi differenze se non nell'energia e le features trasformate si discostano dal valore atteso. Anche analizzando la distribuzione delle features non si riscontrano cluster separati per i diversi parlatori. Passi successivi per la possibilità di applicare

l'adaptation lato FE sono orientati al calcolo delle varianze associate agli stati di ciascun fonema.

3.6.1 Riconoscimento speaker dependent

Di seguito i risultati dei test condotti per valutare la bontà del modello acustico speaker dependent. Il materiale audio utilizzato consiste in un set di audiolibri liberamente scaricati dal sito *Liber Liber* [67]. L'audio è mono, 8 kHz ed è stato addestrato con il tool Sphinxtrain impostato con 138 tied states e 512 gaussiane. Per ogni modello, è stato effettuato il riconoscimento dello stesso audiolibro, quindi dello stesso parlatore, tramite il tool Pocketsphinx utilizzando l'intero corpus di training e l'ultimo capitolo di ogni audiolibro come materiale di testing (non appartenente quindi al materiale utilizzato per il training). Si osserva infatti in ogni plot, un lieve degrado delle prestazioni in corrispondenza dell'ultimo punto del grafo.

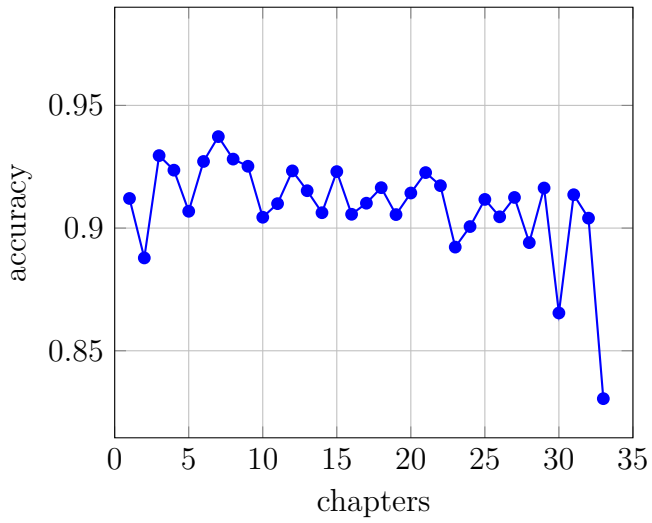


Figura 3.18: Accuratezza del modello acustico Calamita mediante riconoscitore Pocket Sphinx. Parlatore training: Calamita (F). Corpus training: gianburrasca {1 to 32} [466 min]. Parlatore riconoscimento: Calamita (F). Corpus riconoscimento: gianburrasca {1 to 33} [477 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

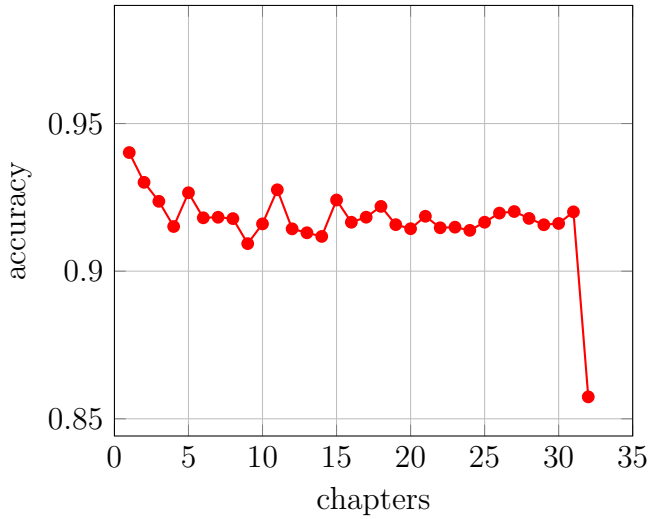


Figura 3.19: Accuratezza del modello acustico Carini mediante riconoscitore Pocket Sphinx. Parlatore training: Carini (M). Corpus training: mattiapascal {1 to 17} [468 min]. & senilita {1 to 13} [473 min]. Parlatore riconoscimento: Carini (M). Corpus riconoscimento: mattiapascal {1 to 18} [500 min]. & senilita {1 to 14} [487 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

3.6.2 Riconoscimento speaker independent

Test riconoscimento speaker independent senza adattamento La Tab. 3.8 mostra l'accuracy media ottenuta testando il modello di ogni parlatore sullo stesso parlatore (valori sulla diagonale) e su parlatori diversi. Il degrado delle prestazione del modello speaker dependent applicato ad un parlatore diverso è notevole.

Il set up dei test condotti è illustrato in Tab.3.7.

Questo spinge ad effettuare ulteriori test applicando le tecniche note di adattamento.

Test riconoscimento speaker independent con adattamento MLLR+MAP. La Tab. 3.9 mostra i nuovi risultati. I valori sulla diagonale principale restano ovviamente inalterati ma si osserva un incremento di circa il 10% sulle prestazioni dovuto all'applicazione degli algoritmi noti MLLR+MAP di adattamento lato Back-End.

training material:

language: Italiano

speaker: Calamita(F), Cecchini (F), Previati (M), Marangoni (M), Carini (M)

corpus: $n-1$ capitoli di ogni parlatore

- Calamita: gianburrasca 01..32
- Cecchini: alice 01..11, coscienzadizeno 01..23, promessisposi 01..37, vicere 01..49
- Previati: malavoglia 01..14
- Marangoni: tigri 01..31
- Carini: mattiapascal 01..17, senilita 01..13

source: Liber Liber
(`http://www.liberliber.it/`)

features:

feature type: MFCCs
audio: 8 kS/s, mono, 16 bit

acoustic model:

gaussians per state: 512
tied states: 138

testing material:

language: Italiano

speaker: Calamita(F), Cecchini (F), Previati (M), Marangoni (M), Carini (M)

corpus: $n-esimo$ capitolo do ogni parlatore

- Calamita: gianburrasca 33
- Cecchini: promessisposi 38
- Previati: malavoglia 15
- Marangoni: tigri 32
- Carini: mattiapascal 18

source: Liber Liber
(`http://www.liberliber.it/`)

recognizer:

tool: Pocket Sphinx (v. 0.1.10)

Tabella 3.7: Parametri usati negli esperimenti.

Tabella 3.8: Accuratezza dei modelli speaker dependent rispetto al riconoscimento fatto con materiale dello stesso parlatore e di parlatori differenti.

Modelli	Materiale di riconoscimento				
	Calamita	Cecchini	Previati	Marangoni	Carini
Calamita	83 %	66 %	42 %	43 %	43 %
Cecchini	62 %	88 %	48 %	50 %	47 %
Previati	47 %	56 %	84 %	73 %	72 %
Marangoni	42 %	51 %	72 %	87 %	70 %
Carini	43 %	53 %	70 %	72 %	85 %

Tabella 3.9: Accuratezza di riconoscimento dei modelli speaker dependent adattati con materiale di parlatori differenti

Modelli	Materiale di adattamento				
	Calamita	Cecchini	Previati	Marangoni	Carini
Calamita	–	79 %	73 %	77 %	76 %
Cecchini	73 %	–	73 %	79 %	77 %
Previati	69 %	76 %	–	79 %	78 %
Marangoni	67 %	75 %	75 %	–	78 %
Carini	70 %	77 %	75 %	80 %	–

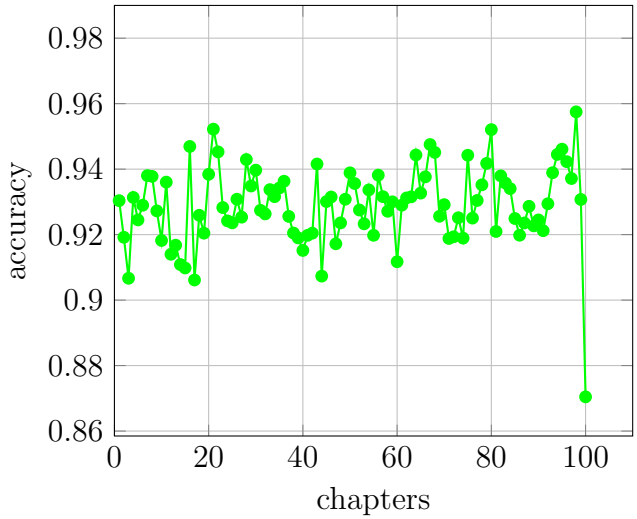


Figura 3.20: Accuratezza del modello acustico Cecchini mediante riconoscitore Pocket Sphinx. Parlatore training: Cecchini (F). Corpus training:alice {1 to 11} [142 min] & coscienzadizeno {1 to 23} [864 min] & promessisposi {1 to 37} [1420 min] & vicere {1 to 24} [678 min]. Parlatore riconoscimento: Cecchini (F). Corpus riconoscimento: alice {1 to 12} [155 min] & coscienzadizeno {1 to 24} [905 min] & promessisposi {1 to 38} [1459 min] & vicere {1 to 25} [697 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

3.6.3 Implementazione di tecniche di adattamento lato Front-End

Il fatto che l'improvement non sia significativo unito all'effort computazionale, spinge verso il tentativo di implementare tecniche di adattamento da applicare al lato FE.

L'attività di ricerca si focalizza sull'implementazione di una tecnica di compensazione del mismatch tra parlatori lato front-end e sulla valutazione delle prestazioni dell'algoritmo in termini di improvement sul rate di riconoscimento. Di seguito, viene riportata la matematica dell'algoritmo di FE adaptation.

3.6.3.1 Algoritmo di Front-End adaptation

Sia x il segnale campionato in un frame T di lunghezza N e $X(\omega)$ lo spettro di x. $f_k(k;\theta)$ è il modello acustico, inteso come misura di probabilità sullo spazio

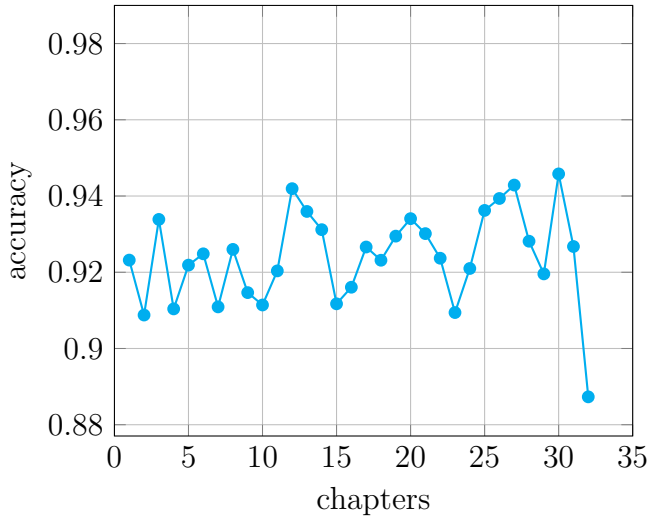


Figura 3.21: Accuratezza del modello acustico Marangoni mediante riconoscitore Pocket Sphinx. Parlatore training: Marangoni (M). Corpus training: tigri {1 to 31} [555 min]. Parlatore riconoscimento: Marangoni (M). Corpus riconoscimento: tigri {1 to 32} [574 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

delle features k e definita dal vettore dei parametri θ. Il modello statistico del segnale acustico associato ad un parlatore è determinato quindi dalla v.a. X nel front-end (FE) o dal vettore deterministico θ nel back-end (BE). Il mismatch tra due parlatori S, $\widehat{S}$, i cui modelli sono X, $\widehat{X}$ nel FE e $\theta, \widehat{\theta}$ nel BE, dipende dalle trasformazioni T_X, T_θ tali che

$$\widehat{X} = T_X\, X, \qquad \widehat{\theta} = T_\theta\, \theta \tag{3.1}$$

che è necessario stimare per compensarne l'effetto. Poiché, in generale, risulta $dim\theta >> dimX$ è conveniente stimare T_X.

Stima della trasformazione T_X

Le due v.a. $X, \widehat{X}$ si possono riscrivere come:

$$\widehat{X} = \widehat{\xi} + \widehat{\mu}_X, \qquad \widehat{\mu}_X = E[\widehat{X}] \tag{3.2}$$

$$X = \xi + \mu_X, \qquad \mu_X = E[X] \tag{3.3}$$

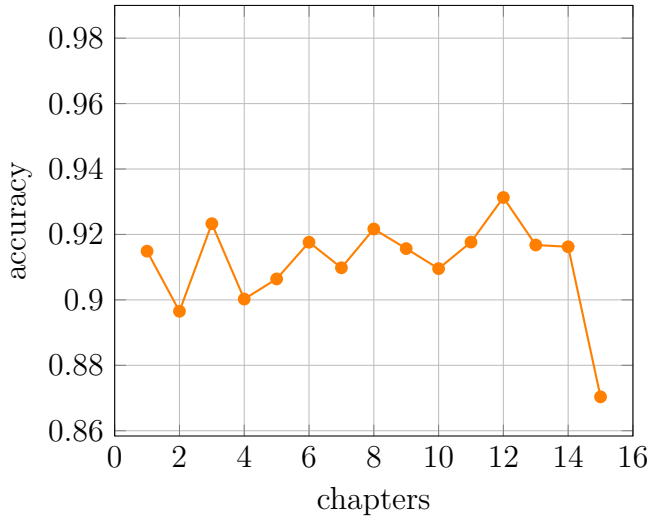


Figura 3.22: Accuratezza del modello acustico Previati mediante riconoscitore Pocket Sphinx. Parlatore training: Previati (M). Corpus training: malavoglia {1 to 14} [502 min]. Parlatore riconoscimento: Previati (M). Corpus riconoscimento: malavoglia {1 to 15} [550 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

dove

$$\widehat{\xi} = \widehat{X} - \widehat{\mu}_X, \qquad E[\widehat{\xi}] = 0 \tag{3.4}$$

$$\xi = X - \mu_X, \qquad E[\xi] = 0 \qquad . \tag{3.5}$$

Supponiamo che esista una trasformazione affine tale che

$$\boxed{\widehat{X} = AX + b} \qquad . \tag{3.6}$$

Allora si può scrivere

$$\widehat{X} = A(\xi + \mu_X) + b = \widehat{\xi} + \widehat{\mu}_X \tag{3.7}$$

da cui si deduce

$$\widehat{\xi} = A\xi, \qquad \widehat{\mu}_X = A\mu_X + b \qquad . \tag{3.8}$$

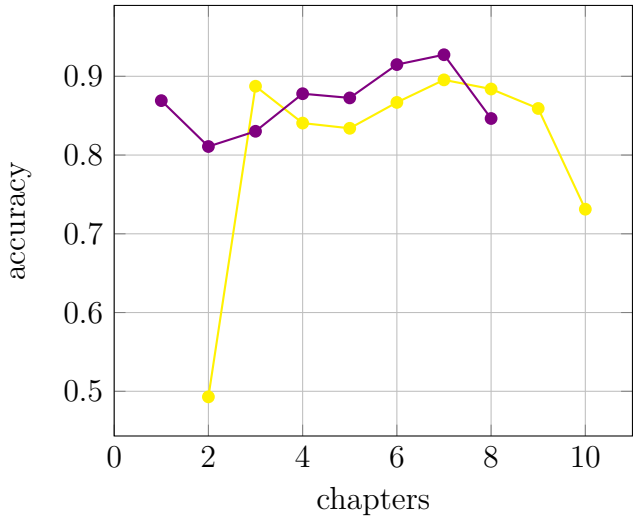


Figura 3.23: Accuratezza del modello acustico parlatore on-line mediante riconoscitore PocketSphinx. Parlatore training: M + F. Corpus training: M [105 min], F [312 min]. Parlatore riconoscimento: M + F. Corpus riconoscimento: M [117 min], F [355 min]. Topologia: Tied states 138 - Misture 512. Versione riconoscitore: Pocket Sphinx (versione 0.1.10).

Inoltre deve risultare

$$\begin{aligned} C_{\widehat{X}\widehat{X}} &= E[(\widehat{X} - \widehat{\mu}_X)(\widehat{X} - \widehat{\mu}_X)^T] = E[\widehat{\xi}\widehat{\xi}^T] \\ &= E[A\xi\xi^T A^T] = AE[\xi\xi^T]A^T \\ &= AE[(X - \mu_X)(X - \mu_X)^T] = AC_{XX}A^T \end{aligned} \tag{3.9}$$

ovvero

$$C_{\widehat{X}\widehat{X}} = AC_{XX}A^T \quad . \tag{3.10}$$

Per le proprietà di $C_{\widehat{X}\widehat{X}}, C_{XX}$ si ha:

$$C_{\widehat{X}\widehat{X}} = \widehat{\phi}\widehat{\Lambda}\widehat{\phi}^T, \qquad C_{XX} = \phi\Lambda\phi^T \tag{3.11}$$

con $\widehat{\phi}, \phi$ matrici unitarie e $\widehat{\Lambda}, \Lambda$ matrici diagonali di autovalori degli operatori $C_{\widehat{X}\widehat{X}}, C_{XX}$, e quindi

$$\widehat{\phi}\widehat{\Lambda}\widehat{\phi}^T = A\phi\Lambda\phi^T A^T \quad . \tag{3.12}$$

Poiché si può sempre scrivere

$$\widehat{\Lambda} = D\Lambda D^T \tag{3.13}$$

con $D = D^T$ diagonale, risulta

$$\widehat{\phi} D\Lambda D^T \widehat{\phi}^T = A\phi\Lambda\phi^T A^T \tag{3.14}$$

e quindi

$$\widehat{\phi} D = A\phi \tag{3.15}$$

da cui

$$\boxed{A = \widehat{\phi} D \phi^T} \tag{3.16}$$

che definisce la trasformazione A, sempre esistente. Infine b è univocamente determinata da

$$\boxed{b = \widehat{\mu}_X - A\mu_X} \quad . \tag{3.17}$$

3.6.3.2 Integrazione dell'algoritmo nel sistema ASR

L'algoritmo è stato inserito all'interno della suite di riconoscimento sviluppata. Uno schema di flusso di tale work-flow è visibile in Fig. 3.24.

3.6.3.3 Valutazione performance

Vengono riportati di seguito i plot significativi relativi al confronto tra i segnali dello speaker A e B e relativi alla successiva applicazione della trasformazione illustrata nei paragrafi precedenti. Nei test viene utilizzato come materiale di riferimento il Capitolo 1 dell'audiolibro "Gianburrasca" e parlatore Calamita di genere femminile per lo speaker A e il Capitolo 1 dell'audiolibro "Promessi Sposi" e parlatore Cecchini di genere femminile per lo speaker B estratti da [67]. Vengono calcolati i campioni dello spettro di potenza dei segnali in esame come mostrato in Fig. 3.25 - 3.26.

In prima analisi, la trasformazione 3.7 è stata applicata in frequenza. In Fig. 3.28 si osserva, dopo aver applicato la trasformazione, il perfetto matching tra la media del segnale del parlatore B (sui cui il parlatore A deve essere adattato) e la media di A trasformato. La matrice di trasformazione ottenuta viene integrata nel work-flow realizzato per essere applicata prima del calcolo delle

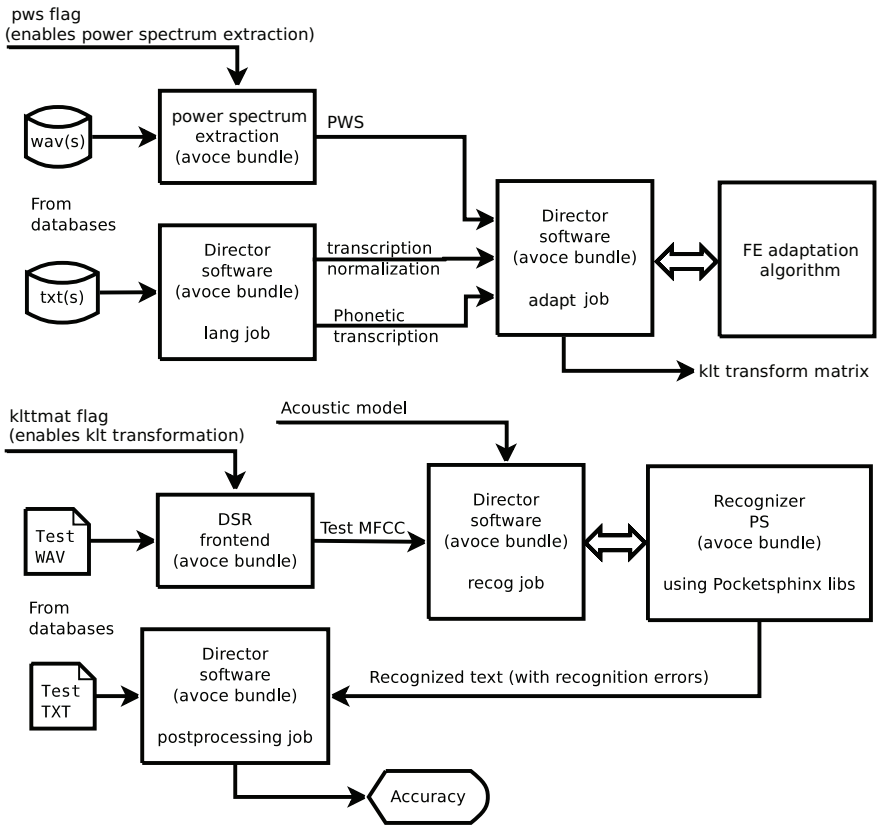


Figura 3.24: Schema a blocchi dell'integrazione nel flusso di estrazione delle feature ETSI dell'algoritmo di FE adaptation.

features. La presenza di valori negativi nel vettore b si riflette però in valori negativi in uscita dal blocco Mel-cepstrum e conseguente NaN all'applicazione del logaritmo per il calcolo delle MFCC. In conseguenza ai risultati ottenuti, che non consentono di portare a termine il riconoscimento del segnale in ingresso, si è deciso di applicare la trasformazione nel dominio del tempo. Con questa strategia, l'estrattore di features non fornisce più in uscita i campioni del power spectrum (128) ma i campioni (200) nel dominio del tempo che vengono processati dall'algoritmo. In Fig. 3.29 i risultati ottenuti applicando la trasformazione nel dominio del tempo. Si evidenziano risultati in frequenza comparabili al primo approccio (anche se non si ottiene una sovrapposizione esatta delle medie) ma in questo caso, il calcolo delle MFCCs va a buon fine.

In Fig. 3.30 - 3.31 l'effetto della trasformazione sulle features e sul power spectrum dello speaker A per un generico frame voiced di una utterance. Si

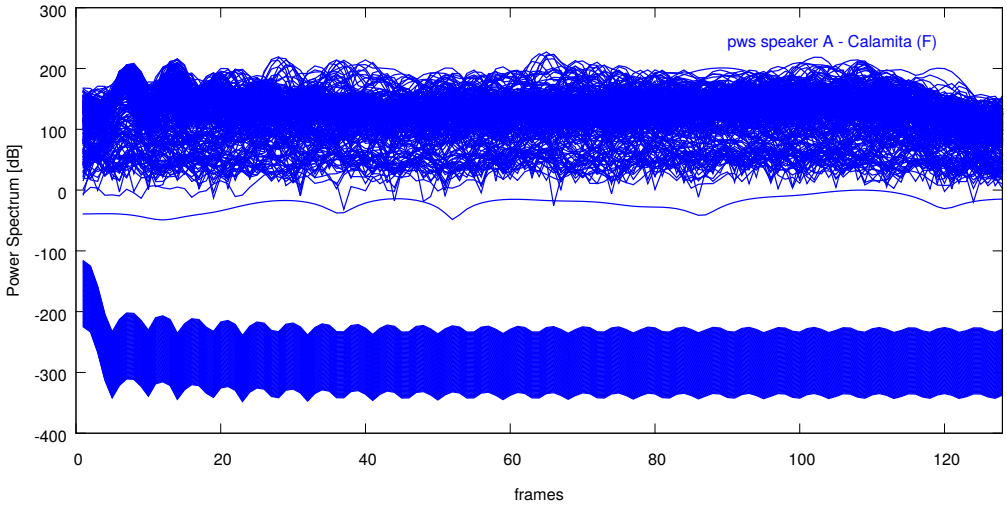


Figura 3.25: Campioni dello spettro di potenza del parlatore A.

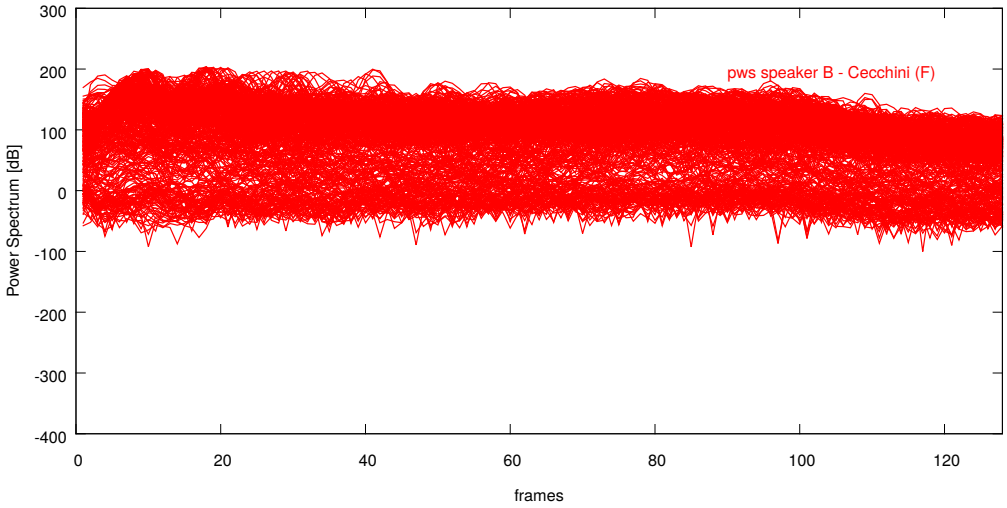


Figura 3.26: Campioni dello spettro di potenza del parlatore B.

noti in Fig. 3.32 come il SIL del parlatore A venga trasformato in segnale con conseguente degradazione nel riconoscimento.

Per ovviare al problema illustrato in Fig. 3.32, nella procedura di test viene incorporata l'informazione sul VAD modificando il front-end in modo tale da fornire in uscita il frame nel tempo e il valore corrispondente del VAD ed applicare la trasformazione solo ai campioni voiced. Con questa modifica i risultati migliorano ma non sono ancora da considerarsi positivi in termini di rate di

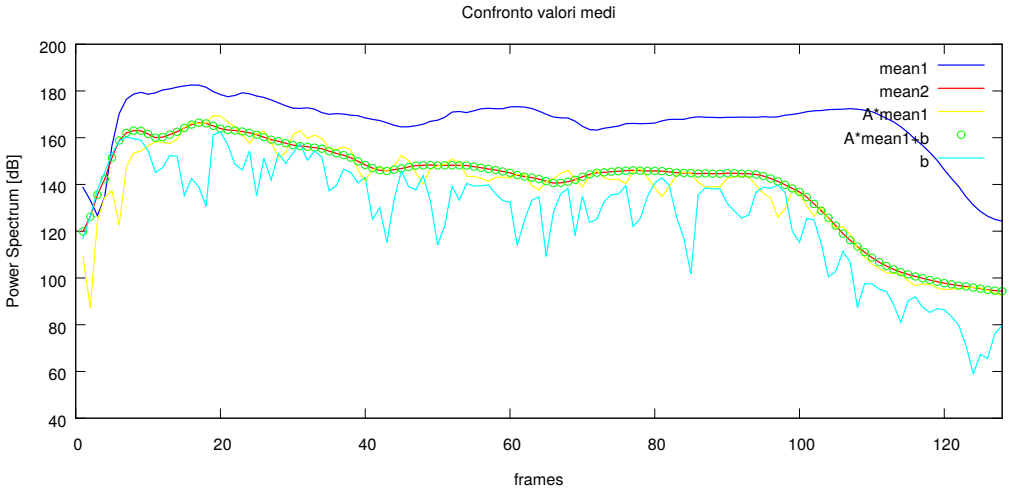


Figura 3.27: Confronto valori medi.

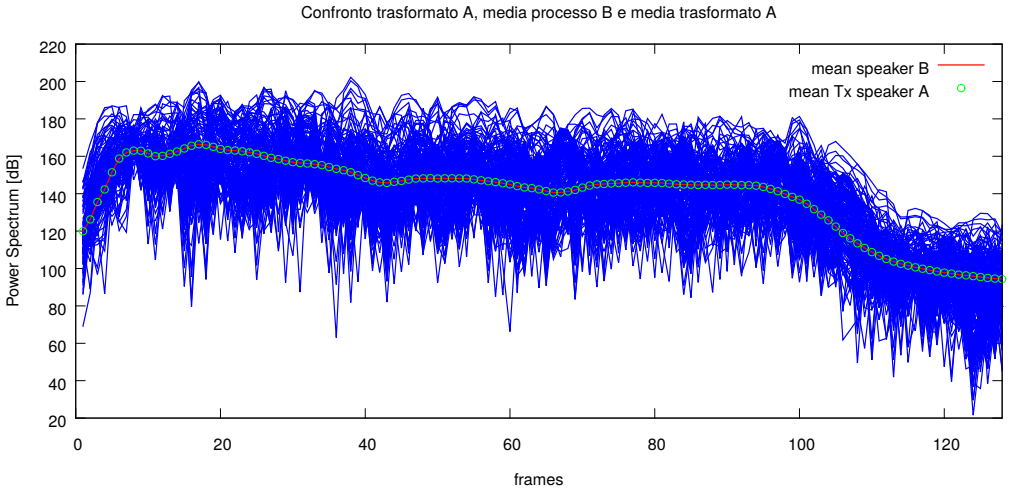


Figura 3.28: Confronto trasformato A, media processo B e media trasformato A - dominio della frequenza.

riconoscimento.

Nella tabella i risultati dei test effettuati per valutare l'accuracy del riconoscitore con l'introduzione dell'algoritmo di adaptation lato front-end. Purtroppo i risultati ottenuti mostrano rate bassi di riconoscimento che non riescono a superare ma anche ad eguagliare i risultati dell'accuracy ottenuta applicando l'adaptation lato back-end.

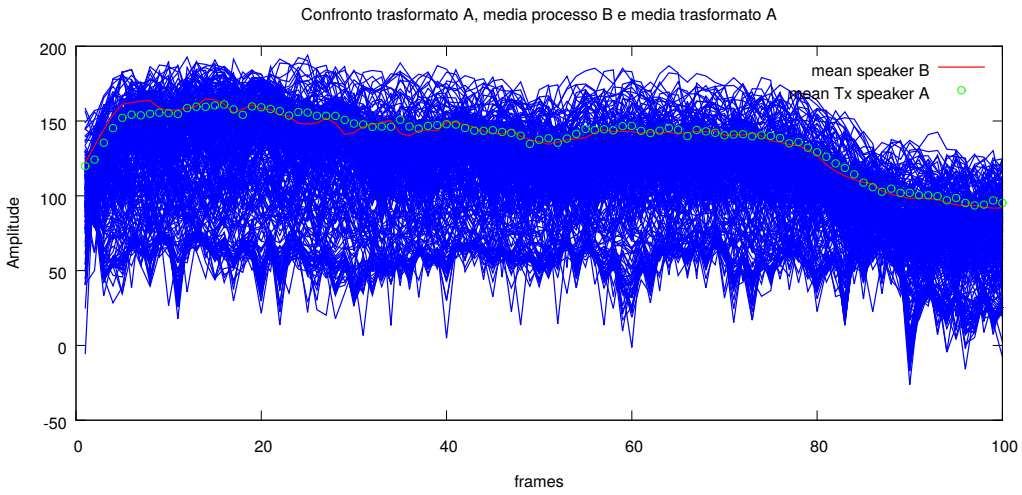


Figura 3.29: Confronto trasformato A, media processo B e media trasformato A - dominio del tempo.

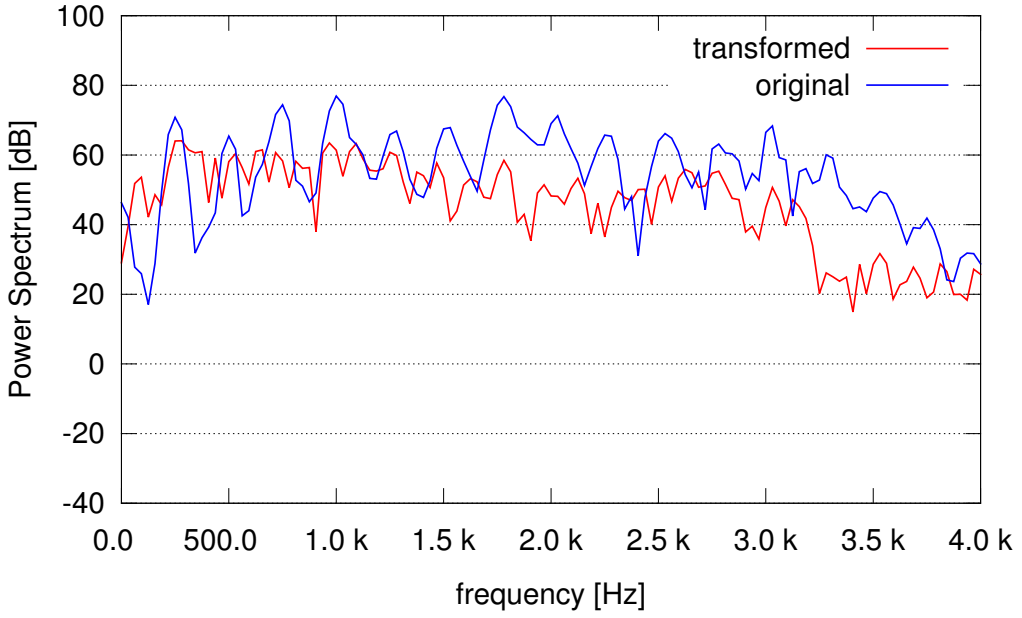


Figura 3.30: Effetto della trasformazione sul power spectrum di una voce femminile per un generico frame voiced di una utterance.

I test effettuati su parlatore A e B di genere diverso non hanno prodotto risultati migliori.

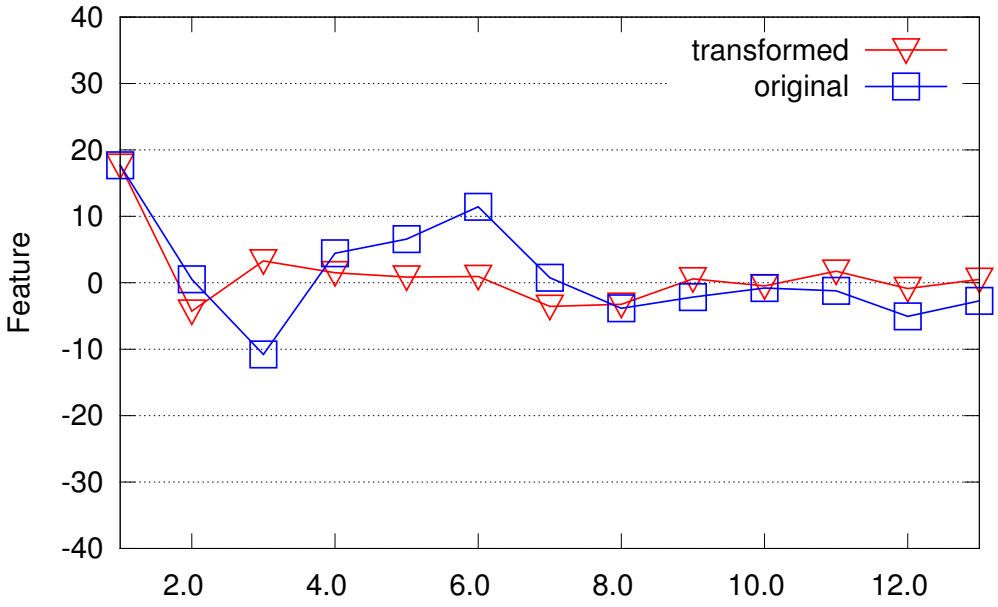


Figura 3.31: Effetto della trasformazione sulle features per un generico frame voiced di una utterance.

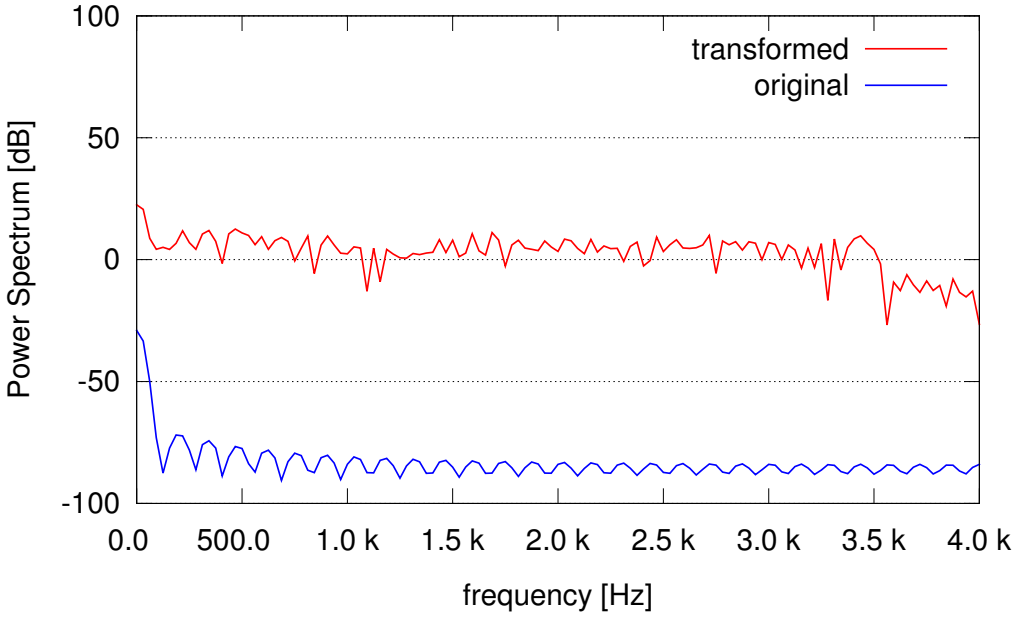


Figura 3.32: Problema trasformazione SIL.

Tabella 3.10: Accuracy valutata su parlatori dello stesso genere.

Mat.riconoscimento	Modello	Trasformazione	Accuracy
Calamita	Calamita etsi8k	-	90 %
Calamita	Cecchini etsi8k	-	52 %
Calamita	Cecchini etsi8k	Applicata	25 %
Calamita	Cecchini etsi8k	Applicata (A diagonale)	52 %

Tabella 3.11: Accuracy valutata su parlatori di genere diverso.

Mat.riconoscimento	Modello	Trasformazione	Accuracy
Calamita	Calamita etsi8k	-	90 %
Calamita	Marangoni etsi8k	-	43 %
Calamita	Marangoni etsi8k	Applicata	32 %
Calamita	Marangoni etsi8k	Applicata (A diagonale)	43 %

Conclusioni Dall'analisi della media delle MFCC in Fig. 3.33 sembra che un problema posso essere il fatto che la matrice A viene calcolata mediando su tutto il segnale senza distinguere gli stati dei fonemi. Infatti le medie delle features dei singoli parlatori non presentano grandi differenze se non nell'energia e le features trasformate si discostano dal valore atteso. Anche analizzando la distribuzione delle features in Fig. 3.34 (si riporta a titolo di esempio la distribuzione della coppia di features MFCCs k_2, k_3) non si riscontrano cluster separati per il parlatore A e B.

Passi successivi per la possibilità di applicare l'adaptation lato FE sono orientati al calcolo delle varianze associate agli stati di ciascun fonema.

3.7 Large Vocabulary Continuous Speech Recognition (LVCSR)

Un altro problema affrontato nell'ambito dello speech recognition è legato all'utilizzo del modello linguistico per l'estrazione di parole dal flusso di fonemi.

I sistemi *Large vocabulary continuous speech recognition* (LVCSR) si basano sulla regola di Bayes che decompone la likelihood per essere massimizzata nel prodotto di due termini: modello acustico e modello linguistico. Dato che en-

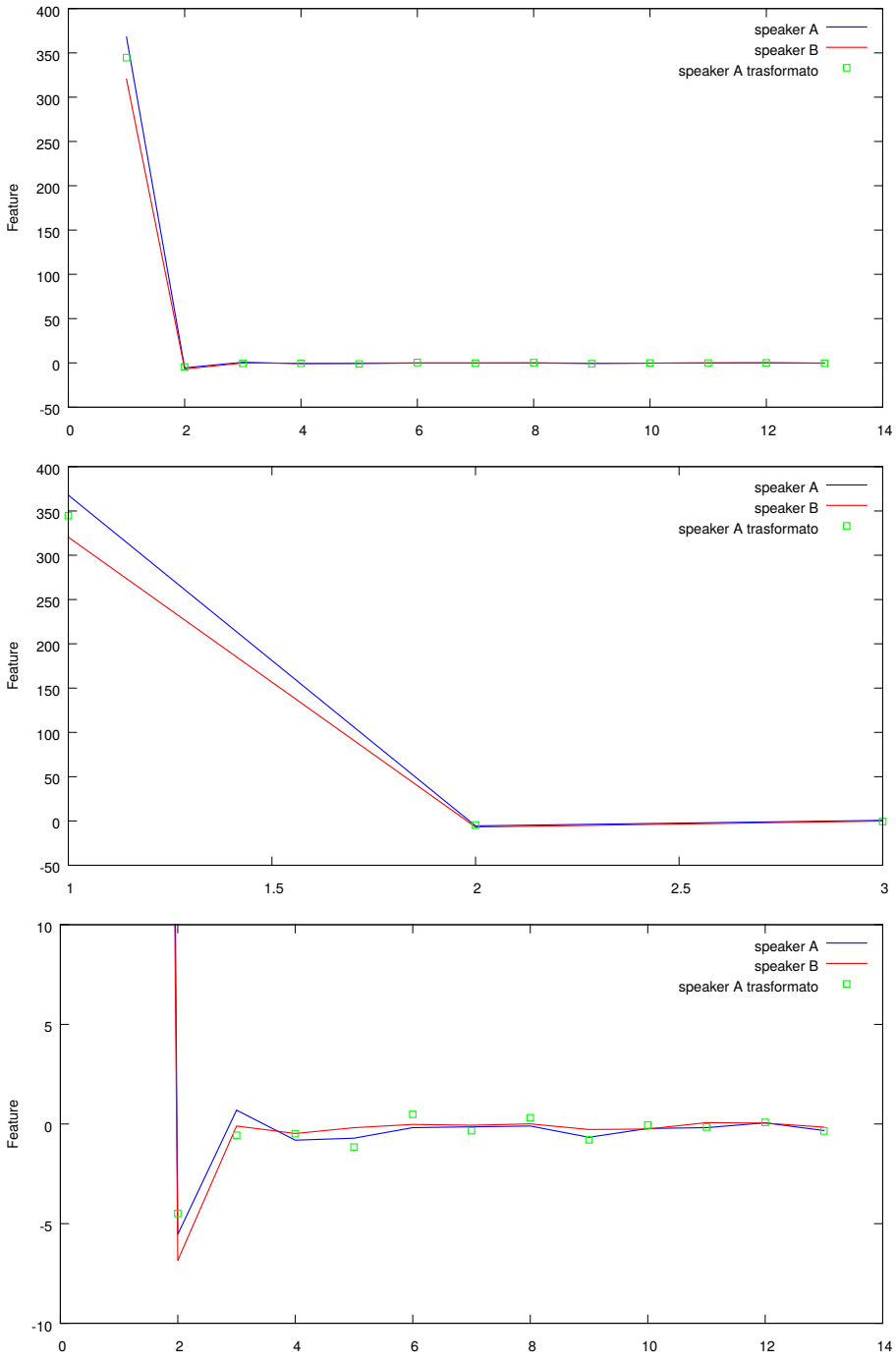


Figura 3.33: Media delle features.

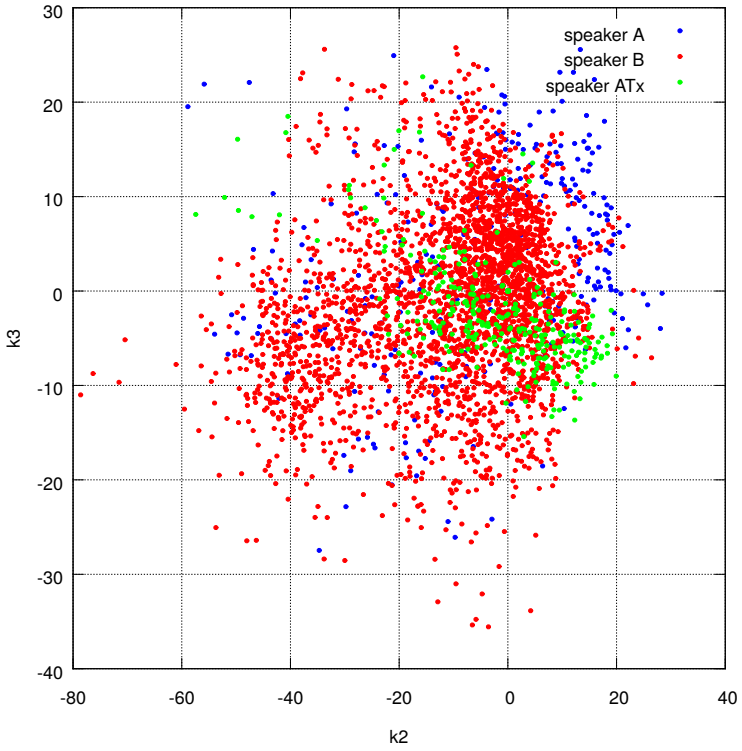


Figura 3.34: Distribuzione della coppia di componenti MFCCs k_2, k_3.

trambi questi termini rappresentano funzioni di densità di probabilità, il problema principale della fase di training del modello si traduce nella stima di queste probabilità, la cui accuratezza influenza fortemente le performance del sistema ASR.

Negli ultimi anni la ricerca ha fatto notevoli progressi nell'ottimizzazione dei modelli acustici, grazie all'impiego delle deep neural networks (DNNs) [68]. Molti articoli hanno dimostrato come il riconoscimento dei fonemi migliori notevolmente sostituendo il Gaussian mixture model con le DNNs [69, 70, 71, 72].

Attualmente, mentre il modello acustico può essere stimato in maniera molto accurata, grazie alla successiva applicazione delle DNNs, il modello linguistico rimane il "collo di bottiglia" dato che richiede un'enorme quantità di materiale di training [12]. A titolo di esempio: per un vocabolario di 20,000 parole, il modello linguistico consiste di 400 milioni di bigrammi, e 8 trilioni di trigrammi. Per questo motivo, un modello linguistico a N-grammi può essere usato con successo se si ha a disposizione un grande data-set di training, mentre fallisce

in modo drammatico con un set limitato di training data. Numerose tecniche sono state proposte in letteratura per risolvere questo problema noto come "the curse of dimensionality" [73]: smoothing techniques [74], class-based N-gram models [75] [76], history merging [77], factored language models [78], low-rank exponential language [79]. Tuttavia nessuna di queste tecniche è stata inserita negli attuali sistemi LVCSR, in quanto devono essere integrate direttamente in fase di decodifica, al fine di sfruttare completamente le loro potenzialità.

3.8 Implementazione di un sistema Dictionary-Based LVCSR

In questo ambito è stata implementata una tecnica "dictionary-based large vocabulary speech recognition" (DB-LVCSR), nella quale il task del decoder è quello di determinare la sequenza più probabile di fonemi, anziché ricercare la sequenza di parole che massimizza il prodotto di modelli acustici e linguistici, come di solito avviene negli attuali sistemi ASR. In questo schema la sequenza di parole è ottenuta al termine di una fase chiamata *words extractor* di estrazione automatica delle parole dallo stream di fonemi (affetti da errore) in ingresso, il cui compito principale è quello di trasformare una sequenza di fonemi in una sequenza di parole appartenenti ad un dato dizionario.

Alla luce quindi di quanto esposto nel cap. 3.7, per proporre una soluzione al problema del "curse of dimensionality" legato alla determinazione del modello linguistico e superare i limiti imposti dai correnti sistemi LVCSR, si propone di seguito un algoritmo che non si basa sulla stima del modello linguistico ma permette di realizzare l'estrazione automatica di parole da un flusso di fonemi in un sistema dictionary-based LVCSR. La necessità di ottenere questa soluzione, nasce anche dall'esigenza di ottenere un algoritmo che possa essere integrato nel sistema ASR implementato, che non fa uso del modello linguistico nella modalità LVCSR. I risultati sperimentali mostrano la validità dell'approccio proposto.

L'algoritmo proposto è stato sviluppato in [80] e di seguito se ne riportano caratteristiche e risultati ottenuti.

3.8.1 DB-LVCSR system

I correnti sistemi LVCSR si basano sul principio del riconoscimento statistico di pattern. La Fig. 3.35 mostra la struttura di base di un sistema LVCSR. Un segnale vocale è convertito dal processamento del front-end in una sequenza di vettori $O = [o_1, \ldots, o_T]$, ognuno dei quali è la rappresentazione compatta dello short-time speech frame estratto dalla forma d'onda nel tempo dell'utterance presa in considerazione. Questa utterance consiste in una sequenza di parole $W = [w_1, \ldots, w_n]$ ed il task del sistema LVCSR è quello di determinare la sequenza più probabile di parole $\hat{W}$ data una sequenza di osservazione O. La regola di Bayes è usata per decomporre la probabilità richiesta $P(W|O)$ in due componenti:

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|O) = \underset{W}{\operatorname{argmax}} \frac{P(W)P(O|W)}{P(O)} = \underset{W}{\operatorname{argmax}} P(W)P(O|W) \; . \qquad (3.18)$$

Questa equazione indica che la sequenza più probabile di parole $\hat{W}$ massimizza sia il termine $P(W)$ che $P(O|W)$. $P(W)$ rappresenta la probabilità a priori di osservare W indipendentemente dal segnale osservato, e questa probabilità è determinata dal modello linguistico. Il termine $P(O|W)$ rappresenta la probabilità di osservare la sequenza di vettori O data la sequenza di parole W, e questa probabilità è determinata dal modello acustico.

Come già detto, l'approccio tipicamente usato per il modello linguistico è quello a N-grammi, il che porta, dato un vocabolario di V words, a V^N potenziali N-grammi; quindi un numero veramente elevato. L'idea è quella di risolvere il problema con un approccio diverso che non richieda la stima del modello linguistico.

La Fig. 3.36 mostra lo schema a blocchi di questo approccio, chiamato dictionary-based LVCSR system. In questo schema l'utterance è considerata come una sequenza di fonemi $S = [s_1, \ldots, s_m]$ ed è compito del decoder quello di determinare la sequenza più probabile di fonemi $\hat{S}$, data l'osservazione O. Usando la regola di Bayes per decomporre $P(S|O)$, si ottiene:

$$\hat{S} = \underset{S}{\operatorname{argmax}} P(S|O) = \underset{S}{\operatorname{argmax}} \frac{P(S)P(O|S)}{P(O)} = \underset{S}{\operatorname{argmax}} P(O|S) \quad (3.19)$$

In questo schema, è compito del word extractor determinare la sequenza di parole $\hat{W}$ data la sequenza di fonemi $\hat{S}$. Formalmente, questo stage può essere rappresentato come una trasformazione H tale che:

$$\hat{W} = H(\hat{S}, D) \tag{3.20}$$

dove D è il dizionario dato.

La principale differenza tra i sistemi LVCSR e DB-LVCSR è che nel primo è necessaria la stima della pdf in uno spazio high-dimensional, (e questo porta al problema del "curse of dimensionality"), mentre nel secondo questo problema non esiste perché H è una trasformazione deterministica.

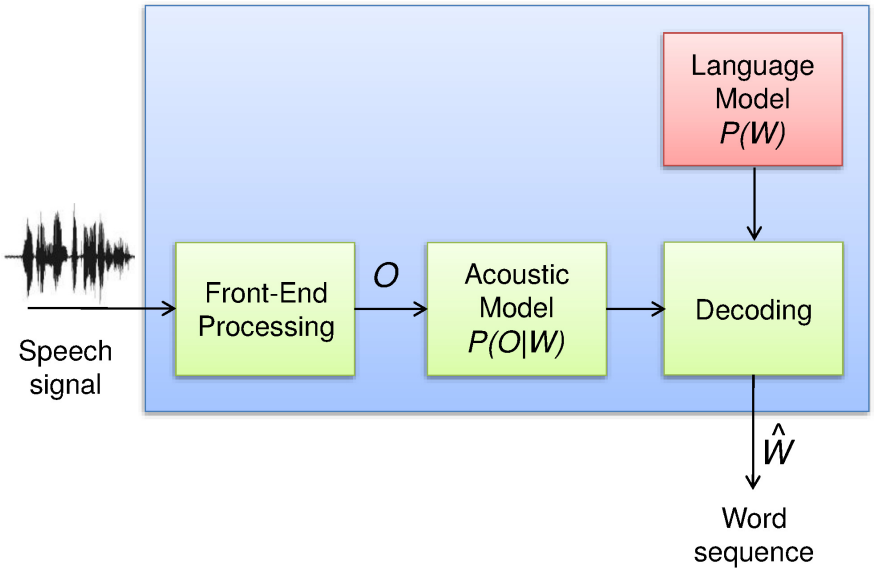


Figura 3.35: Language-based LVCSR system.

Di seguito si farà quindi riferimento ad un sistema dictionary-based LVCSR ed alla determinazione della trasformazione H.

3.8.2 Formulazione matematica del problema

Il problema in esame è quindi quello di decodificare una sequenza di fonemi in una sequenza di parole, e può essere formulato come segue. Data la sequenza di fonemi $S = [s_1, \ldots, s_m]$, trovare la sequenza di parole $\hat{W} = [\hat{w}_1, \ldots, \hat{w}_n]$ appartenenti al dizionario D con trascrizione fonetica $\hat{S} = [\hat{s}_1, \ldots, \hat{s}_m]$, in modo tale che

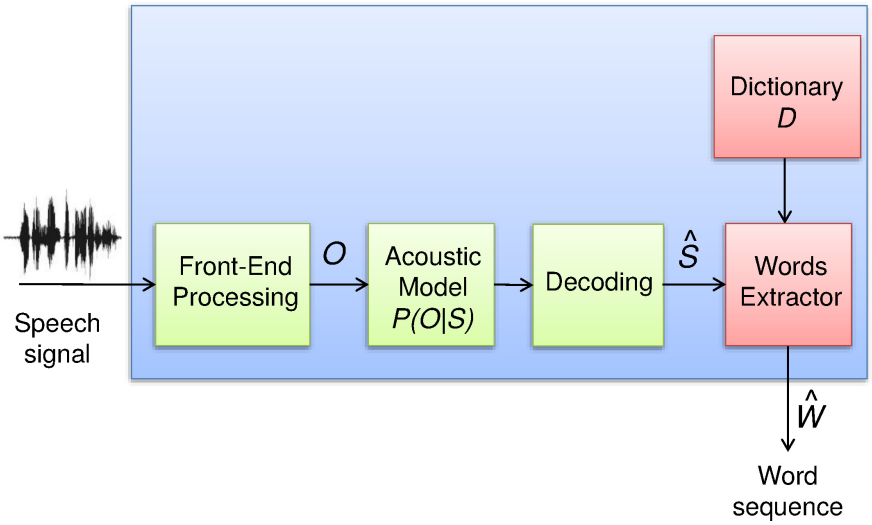


Figura 3.36: Dictionary-based LVCSR system.

la distanza $L(S, \hat{S})$ sia minima.

$$\hat{S} : \min L(S, \hat{S}) = \min_{W} L\left(S, \hat{S} = T(W) | w_i \in D\right) \tag{3.21}$$

dove $T(W)$ denota la trascrizione fonetica, cioè la trasformazione della sequenza di parole W in una sequenza di fonemi S. Per risolvere questo problema, può essere applicato un algoritmo di *inexact matching* in modo tale da determinare il valore della similarity delle due sequenze S, $\hat{S}$. A questo fine, può essere usata la ben nota distanza di Levensthein [81]. Nella teoria dell'informazione e nella teoria dei linguaggi, la distanza di Levenshtein, o distanza di edit, è una misura per la differenza fra due stringhe; serve a determinare quanto due stringhe siano simili. La distanza di Levenshtein tra due stringhe x e y è il numero minimo di modifiche elementari che consentono di trasformare la stringa x nella y. Per modifica elementare si intende: la cancellazione di un carattere, la sostituzione di un carattere con un altro, o l'inserimento di un carattere. Quindi la distanza di Levensthein tra due stringhe x e y di lunghezza p e q, può essere definita come:

$$L\left([x_1, .., x_p], [y_1, .., y_q]\right) = L_{p,q} \tag{3.22}$$

dove

$$L_{p,q} = \begin{cases} p\ , & \text{if } \ q = 0 \\ q\ , & \text{if } \ p = 0 \\ \min\{L_{p-1,q+1}, L_{p,q-1} + 1, & \\ \qquad L_{p-1,q-1} + \Delta(x_p, y_q)\}\ , & \text{otherwise} \end{cases} \tag{3.23}$$

e $\Delta(x, y)$ è la distanza tra i due simboli x e y definita come:

$$\Delta(x, y) = \begin{cases} 0, & \text{if } x = \text{y} \\ 1, & \text{otherwise} \end{cases} \ . \tag{3.24}$$

Sebbene (3.21) e (3.22) definiscano formalmente la trasformazione non nota H in (3.20), l'equazione risultante è troppo complessa da risolvere, di conseguenza verrà adottato un approccio subottimo.

3.8.3 Algoritmo per l'estrazione automatica delle parole da un flusso di fonemi in un sistema DB-LVCSR

L'algoritmo proposto, riceve in ingresso la sequenza di fonemi $S = [s_1, \ldots, s_m]$ e determina in uscita la sequenza di parole $\hat{W} = [\hat{w}_1, \ldots, \hat{w}_n]$, attraverso una serie di passaggi, come mostrato nel flow-chart di Fig. 3.37.

L'idea è quella di sfruttare l'informazione sull'accento (che viene quindi individuato in fase di modeling del modello acustico) come flag per l'esistenza di una parola. La funzione dell'accento infatti è quella di identificare univocamente una parola. La compensazione degli errori viene fatta utilizzando la distanza di Levensthein. Si individuano quindi nello stream di ingresso delle finestre di lunghezza variabile che hanno come estremo superiore l'accento successivo e che quindi sicuramente conterranno una parola. Nella finestra di analisi si individua la parola candidata come la parola che minimizza la distanza di Levensthein calcolata rispetto alle parole del dizionario di riferimento. Con lo stesso ragionamento si individua la parola successiva. La zona tra i due accenti rappresenta una zona di incertezza per la scelta del taglio. Si considerano allora tutte le coppie di finestre ottenute tramite shift nella zona di incertezza, e si determina la parola ottima come appartenente alla coppia che minimizza la distanza di Leventhein rispetto alla distanza di riferimento. Sostanzialmente combinando le

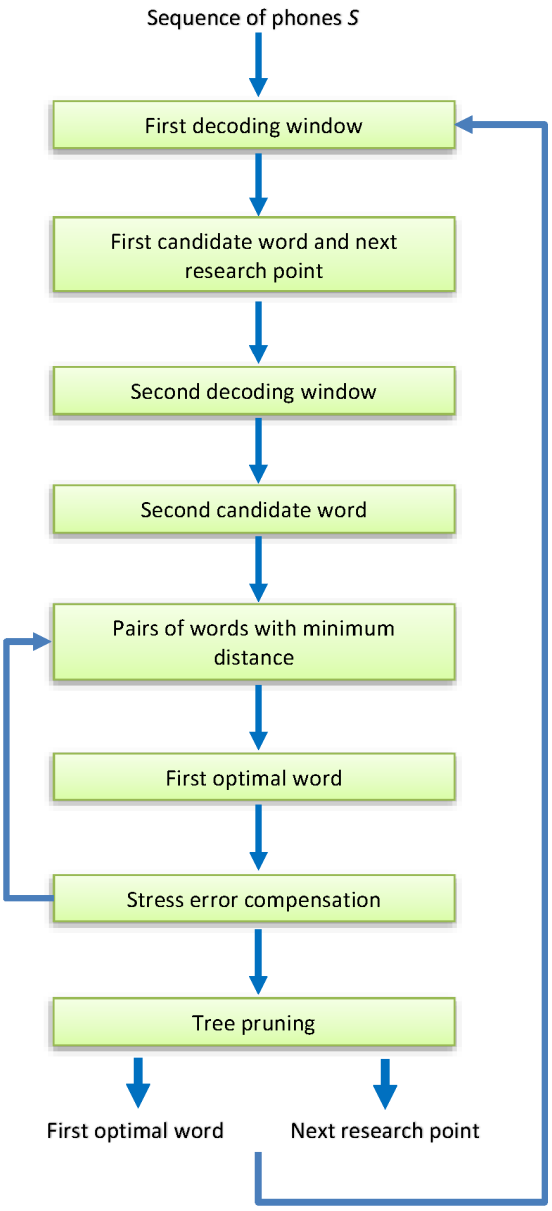


Figura 3.37: Schema a blocchi dell'algoritmo.

due informazioni sull'accento e sulla distanza di Levensthein ed utilizzando delle finestre di lunghezza variabile che si muovono nella zona di incertezza definita tra un accento ed il successivo, vengono determinati i rami di un albero contenente le parole candidate, che vengono via via eliminati sulla base di regole euristiche, portando alla scelta delle parole ottime.

Viene fatta l'assunzione che l'accento di una parola la caratterizzi completamente e viene derivato il vettore $a = (a_0, \ldots, a_M)$ delle posizioni dell'accento durante la fase di training. La trascrizione fonetica del materiale audio usato per lo stage di training, viene realizzata tramite il software "eSpeak" [82]. Questo tool è in grado anche di estrarre l'informazione sull'accento, in questo modo il modello acustico contiene l'informazione sulla posizione dell'accento e lo stream S di fonemi contiene anche il simbolo dell'accento. A causa delle limitazioni nella precisione del modello acustico, lo stream S è affetto da errori sia sui simboli fonetici che sulla posizione dell'accento; sarà quindi compito dell'algoritmo compensare questi errori.

Step 1) Scelta della prima decoding window La decodifica dello stream di fonemi in uno stream di parole è fatta attraverso l'uso di decoding windows DW. Una volta che la posizione dell'accento è determinata, viene scelta la prima decoding window compresa tra lo start index b_{j-1} e la posizione dell'accento a_j. In questo modo, l'accento a_{j-1} appartiene sicuramente alla finestra DW_{j-1}. Dato che l'accento identifica univocamente una parola, questa finestra sicuramente conterrà una parola.

Step 2) Scelta della prima parola candidata e del punto di ricerca successivo La regione compresa tra gli indici a_{j-1} e a_j rappresenta una zona di incertezza δ: i due accenti alle posizioni a_{j-1} e a_j corrispondono univocamente a due parole, ma la posizione esatta dell'inizio e fine di entrambe le parole non è nota. Quindi, come mostrato in Fig. 3.38, vengono selezionate tutte le sequenze $p_j^1, p_j^2, \ldots$ all'interno della finestra DW_{j-1} che inizia all'indice b_{j-1} e termina all'indice $a_{j-1} + i$, con $i = 1, \ldots, \delta$. Ognuna di queste sequenze è confrontata con la trascrizione delle parole appartenenti al dizionario D e per ogni coppia di sequenze viene calcolata la distanza di Levensthein, come definita in (3.22)–(3.23), per compensare gli errori sui simboli fonetici (e non quelli sull'accento). La sequenza che minimizza la distanza di Levensthein viene scelta come "parola candidata". Ovviamente nel primo searching stage, si può verificare di ottenere più parole con la stessa distanza. Si sceglie, come semplice regola, di scegliere la prima parola con la distanza inferiore ad una data soglia, dato che questa parola sarà utilizzata semplicemente per la definizione del successivo punto di

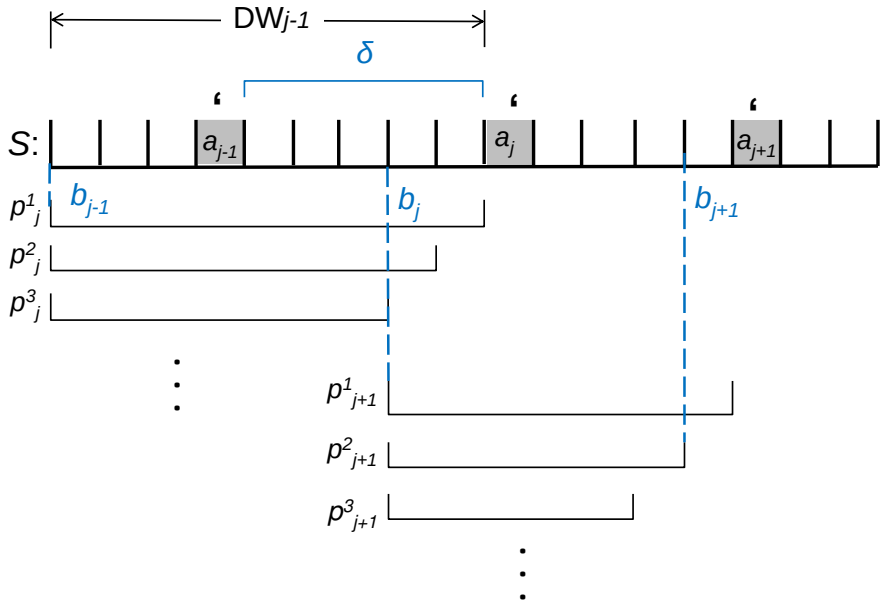


Figura 3.38: Windowing e selezione della coppia di parole.

ricerca b_j e qualsiasi errore sulla larghezza della finestra sarà compensato dallo step successivo dell'algoritmo.

Step 3) Scelta della seconda decoding window Come nello step 1 (3.8.3), la seconda decoding window è scelta tra l'indice b_j derivato dallo step precedente, e l'indice a_{j+1}, in modo tale da contenere l'accento alla posizione a_j.

Step 4) Scelta della seconda parola candidata Come nello step 2 (3.8.3), la seconda "parola candidata" è, tra le parole appartenenti al dizionario D, quella che minimizza la distanza di Levensthein rispetto alla sequenza nella finestra. In questo modo, viene identificato nello stream di fonemi, l'indice b_{j+1} corrispondente alla fine della seconda parola.

Step 5) Scelta della coppia di parole alla minima distanza La coppia di parole candidate viene presa come baseline per la determinazione della prima parola estratta. Per rimuovere l'incertezza della regione δ viene selezionato un set di coppie (w_{j-1}, w_j) muovendo l'indice corrispondente al taglio tra le due parole nella regione di incertezza.

Step 6) Scelta della prima parola ottima Per ogni parola appartenente ad una data coppia, viene calcolata la distanza di Levensthein rispetto alle parole del dizionario. La "parola ottima" è $\hat{w}_{j-1} = min\,(L(w_{j-1}, w_q) + L(w_j, w_q))$, con $w_q \in D$. Alla fine, si ottengono la prima "parola ottima" $\hat{w}_{j-1}$ e l'indice $\hat{b}_{j-1}$, e questi valori rappresentano il punto iniziale per la ricerca della parola successiva ripartendo dallo step 1 3.8.3. Un esempio della procedura è illustrato in Fig. 3.39.

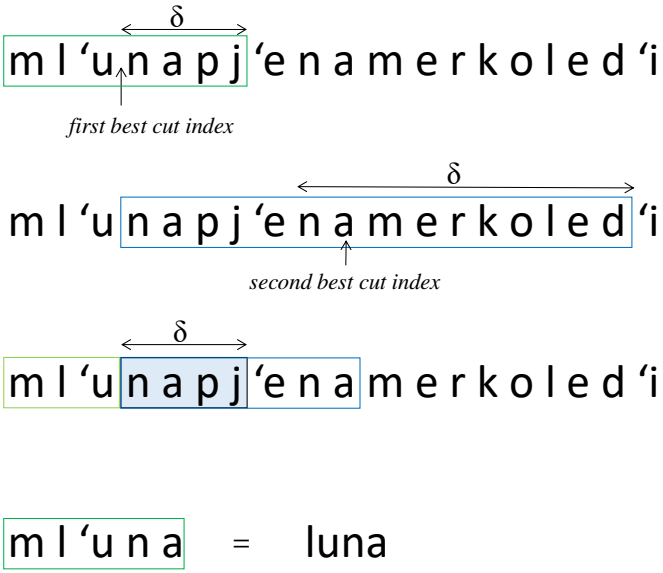


Figura 3.39: Scelta della prima "parola ottima".

Step 7) Compensazione dell'errore sull'accento La sequenza di fonemi S è affetta da errori random sull'accento che possono essere di tre tipi: errore sulla posizione, mancato accento, multipli accenti. Il primo errore può essere compensato facilmente dato che l'accento è semplicemente un flag per l'individuazione dell'occorrenza di una parola. La presenta di accenti multipli o l'assenza di accento, invece, comportano variazioni significative sulla larghezza della finestra che si traducono nell'estrazione di una singola parola lunga (assenza dell'accento) o di più parole brevi (accento multiplo). L'assenza di accento è compensata verificando che la parola selezionata come parola ottima sia quella con la distanza minima inferiore ad una data soglia ($th = 2$ per permettere la correzione di un errore alla volta, evitando un rapido incremento dei risultato). Se questa condizione non è soddisfatta, la finestra contenente la sequenza di fonemi, corrispondenti alla

parola ottima, è ulteriormente segmentata posizionando l'accento all'inizio della finestra, prima di ritornare allo step 5 (3.8.3).

Per comprendere meglio l'algoritmo, si consideri la finestra $|l\ u\ n\ a\ p\ j\ 'e\ n\ a|$ nella quale l'accento sul fonema $|'u|$ è assente. In questo caso la parola ottima ottenuta ha una distanza di Levensthein pari a 4, e non soddisfa il vincolo sulla soglia. In conseguenza di ciò, assumendo questo valore come una distanza troppo grande, si deduce che l'accento è assente e quindi l'algoritmo procede inserendo l'accento alla posizione $|l\ 'u\ n\ a\ p\ j\ 'e\ n\ a|$, tornando quindi allo step 5 (3.8.3) e cercando la coppia di parole con la minima distanza: $|l\ u| + |n\ a\ p\ j\ e\ n\ a|$
$|l\ u\ n| + |a\ p\ j\ e\ n\ a|$
$|l\ u\ n\ a| + |p\ j\ e\ n\ a|$
...
$|l\ u\ n\ a\ p\ j| + |e\ n\ a|$
La coppia $|l\ u\ n\ a| + |p\ j\ e\ n\ a|$ ha la minima distanza e quindi la parola corretta è *luna*. L'accento multiplo può essere compensato migliorando il modello acustico, interpolando gli accenti nella sequenza da correggere. La correzione dell'accento multipla non è stata ancora implementata.

Step 8) Pruning L'algoritmo di Levensthein è capace di correggere gli errori ma produce risultati multipli. Al fine di rimuovere la ridondanza nelle soluzioni, viene incluso nell'algoritmo un meccanismo di pruning che permette di rimuovere dall'albero i rami corrispondenti a path non a minima distanza; meccanismo che agisce man mano che vengono estratte le parole, come illustrato in Fig. 3.40.

3.8.4 Risultati sperimentali

I test sperimentali sono stati condotti su un ampio speech corpus (8 kilosamples per second, 16 bit) ricavato da audiolibri in lingua italiana liberamente scaricabili dal sito *liber liber* [67] e relativo a cinque diversi parlatori, due donne (A, B) e tre uomini (C, D, E) come riportato in Tab. 3.12.

Dato che le performance del sistema dictionary-based LVCSR dipendono dall'accuratezza sia del modello acustico che dell'algoritmo stesso, i test sono condotti in modo da evidenziare le proprietà esclusivamente di quest'ultimo. Come detto, l'algoritmo ha due caratteristiche principali: *i)* la capacità di estrarre da una sequenza di fonemi, una sequenza di parole; *ii)* la capacità di correggere gli errori sia sui fonemi, in termini di inserimento e cancellazione, sia sull'accento. I

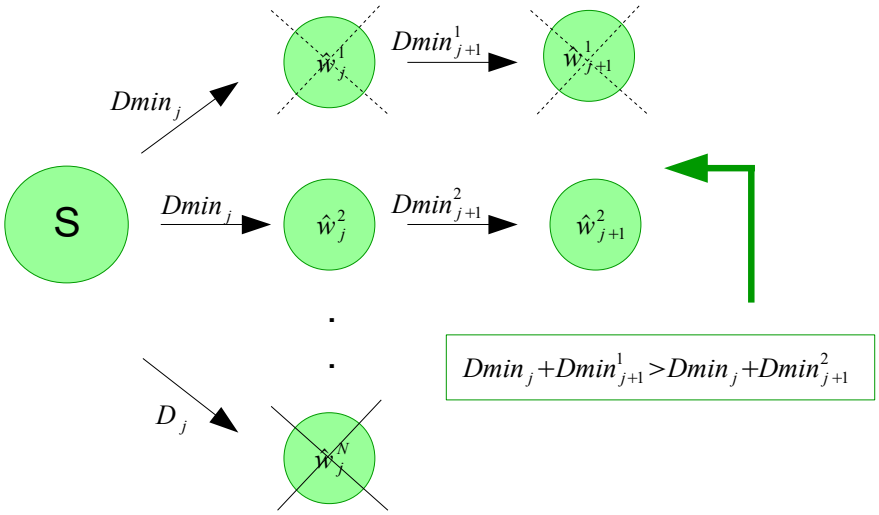


Figura 3.40: Tree pruning.

Tabella 3.12: Materiale audio utilizzato per la creazione del database. Sorgente: *liber liber* (`http://www.liberliber.it/`). Il materiale è stato utilizzato sia a scopo di training che di testing.

Speaker	Gender	Audiobook	Chapter	Duration [s]
A	F	"Il giornalino di Gianburrasca" by L. Bertelli	I	28639
B	F	"I promessi Sposi" by A. Manzoni	I	87569
C	M	"Fu Mattia Pascal" by L. Pirandello	I	30004
D	M	"Le tigri di Mompracem" by E. Salgari	I	34491
E	M	"I Malavoglia" by G. Verga	I	33031

tesi sperimentali sono quindi orientati alla misura della *word recognition accuracy* W_{Acc} all'uscita del words extractor, in funzione di errori dati nel modello acustico (utilizzando quindi una popolazione di stream di fonemi affetti da errori sui fonemi e sull'accento). Nel primo grafico Fig. 3.41 è mostrata la W_{Acc} in funzione dell'errore sull'accento mantenendo inalterati i simboli fonetici $E_{phone} = 0$, nel secondo Fig. 3.42 si considera la W_{Acc} al variare dell'errore sui simboli fonetici con $E_{stress} = 0$.

Conclusioni Come mostrato dai risultati, la W_{Acc} dell'estrattore è linearmente dipendente all'errore sull'accento ed all'errore sui fonemi. Tuttavia, a meno di

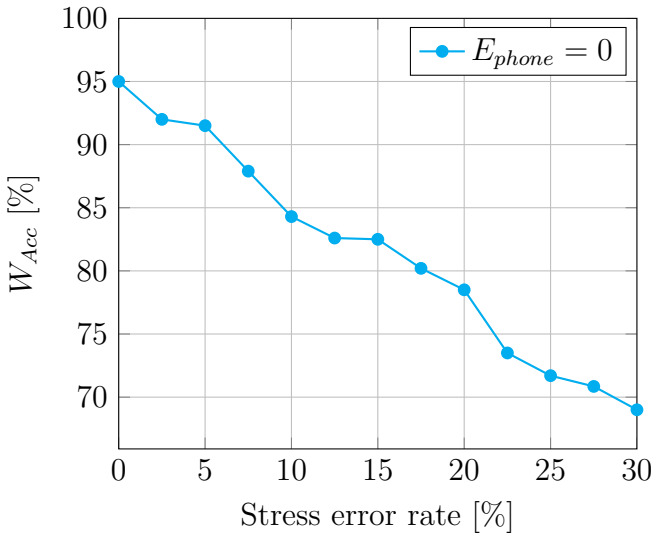


Figura 3.41: Word recognition accuracy in funzione dello stress error rate.

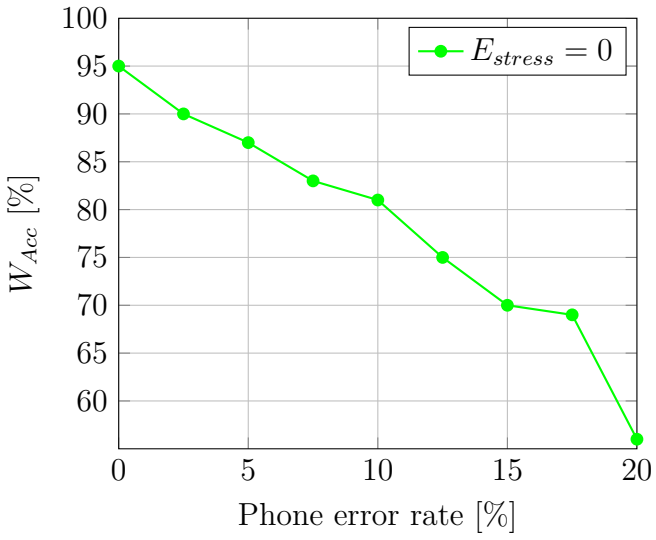


Figura 3.42: Word recognition accuracy in funzione del phone error rate.

un offset del 5% dipendente dall'euristica dell'algoritmo, l'algoritmo è in grado di evitare la degenerazione dell'errore. Ad ogni modo i risultati sono utili per determinazione l'accuratezza del modello acustico richiesta per avere buone performance in un sistema LVCSR. Sviluppi futuri sono quindi orientati all'ottimizzazione del modeling acustico, sia in termini di accuracy sui simboli che

in termini di stress detection ed ovviamente ad una ottimizzazione dell'algoritmo per cercare di compensare il 5% di offset sui risultati finali. Il primo passo consiste nel concatenare più di due decoding windows, per incrementare la word recognition performance. Il secondo passo potrebbe essere quello di usare un criterio basato su soglia adattativa al fine di adattare la capacità di correzione dell'algoritmo all'errore stimato sull'input stream.

Capitolo 4

Speaker Identification

Un sistema biometrico è un particolare sistema che ha la funzionalità e lo scopo di identificare una persona sulla base di una o più caratteristiche biologiche e comportamentali (biometria), confrontandole con dati precedentemente acquisisti e presenti nel database del sistema, tramite algoritmi e sensori di acquisizione di dati in input. In un sistema di identificazione biometrica, le caratteristiche prese in considerazione sono: fisionomia del volto, impronta digitale, palmo della mano, retina, . . . e impronta vocale. Nell'ambito dei sistemi di identificazione biometrica, l'identificazione del parlatore (*speaker identification*) tramite impronta vocale è considerato uno dei metodi meno invasivi, essendo la voce il segnale più naturale da produrre per il soggetto ed il più semplice da acquisire. Infatti sia il sistema di telefonia che Internet, fornisco reti di sensori per il trasporto del segnale vocale [83, 84]. Applicazioni tipiche di speaker identification sono: accesso controllato ad applicazioni, servizi di telefonia per l'autorizzazione di transazioni, sistemi di diarization, ricerca di uno speaker in un ampio audio corpora [85, 86].

In generale, i sistemi di *speaker recognition* utilizzano la voce per riconoscere, verificare o identificare individui [87, 88, 89] e possono operare in due diverse modalità: *verifica* e *identificazione*. La verifica del parlatore (*speaker verification*) è un confronto $1 : 1$, dove la voce del parlatore è confrontata con un'unica impronta vocale (o "modello del parlatore"), mentre l'identificazione (*speaker identification*) è un confronto $1 : N$ dove la voce è confrontata con N modelli distinti [90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102]. In sostanza, nella Speaker Verification, la persona "si presenta" tramite un suo identificativo di qualche tipo (nome utente, smart card, ecc.) e l'applicazione verifica se la voce di chi sta parlando corrisponde a quella della sua impronta vocale memorizzata nel sistema. Se la voce di chi parla e l'impronta sono simili, cioè se il valore

di somiglianza supera una certa soglia, la verifica ha successo e la persona è autenticata. In una applicazione di Speaker Identification, invece, la persona che parla "non si presenta", e si deve misurare il grado di somiglianza della sua voce rispetto ad un insieme di impronte vocali disponibili. Se tutti i possibili utenti del sistema sono già stati catalogati ("closed-set"), si assume che la voce della persona debba comunque corrispondere ad una di quelle disponibili. Diversamente, se esiste la possibilità che un utente non sia stato catalogato in precedenza ("open-set"), può verificarsi che la voce non corrisponda ad alcuna voce presente nel data-set. Entrambe le metodologie possono essere considerate come un un caso particolare del problema più generale di statistical pattern recognition [93]. Da questo punto di vista, la differenza principale tra l'identificazione e la verifica del parlatore, è che nel primo la classificazione è basata su un set di N modelli (uno per ogni speakers), mentre, nel secondo caso, il numero totale di modelli derivati dal training equivale a due (hypothesized speaker e background model).

In questo capitolo, si farà riferimento esclusivamente alla tecnica di speaker identification, e più specificatamente all'identificazione closed-set, nella quale il parlatore da identificare appartiene ad un gruppo preesistente di parlatori [87].

Dato il crescente interesse verso questi sistemi, si è pensato di unire le conoscenze nell'ambito dei sistemi ASR, alle tecniche di identificazione biometrica per poter realizzare sistemi di interazione vocale "personalizzati", che siano in grado cioè di capire "cosa è stato detto" e "da chi è stato detto". In questo capitolo, verranno quindi dapprima illustrate le attuali tecniche di speaker identification, per poi proporne una variante innovativa, validata da risultati sperimentali ed infine integrarla nel sistema DSR illustrato nel capitolo 3 al fine di ottenere un sistema combinato di speech recognition/speaker identification. Il sistema proposto verrà testato su un numero variabile di parlatori ed anche in condizioni di overlap tra gli speakers.

4.1 Cenni storici

Negli ultimi 50 anni, la ricerca nell'ambito dello speech e speaker recognition è stata portata avanti intensamente in tutto il mondo, spinti dai progressi nel campo di signal processing, algoritmi, architetture, e hardware. I progressi tecnologici compiuti negli ultimi 50 anni possono essere riassunti come segue [103]:

- Dal template matching al modeling statistico basato su corpus: HMM e n-grams;
- Dal banco di filtri alle features cepstrali (cepstrum + Δcepstrum + $\Delta\Delta$cepstrum),
- Dai metodi "distance"-based a quelli basati su verosimiglianza;
- Dalla massima verosimiglianza ai metodi discriminativi,
- Dalle singole parole al riconoscimento continuous speech,
- Dal riconoscimento small vocabulary a quello large vocabulary recognition,
- Dalle unità context-independent fino a quelle context-dependent utilizzate nel riconoscimento,
- Dal clean speech al noisy/telephone speech,
- Da un singolo parlatore fino al riconoscimento speaker independent,
- Dal riconoscimento di un monologo fino ad una intera conversazione,
- Dalla modalità singola (solo segnale audio) al multimodale (audio/visual speech recognition),
- Da applicazioni non commerciali a molte applicazioni commerciali pratiche.

La maggior parte di questi progressi hanno avuto luogo sia nel campi del riconoscimento vocale che del riconoscimento del parlatore.

4.2 Tecniche di speaker identification

Le operazioni principali svolte da un sistema di speaker identification sono: *feature extraction*, *speaker modeling*, e *speaker classification*.

La fase di estrazione delle features è molto importante nel processo di identificazione perché permette di catturare le caratteristiche specifiche del segnale vocale del parlatore, riducendo la complessità del modello. Come nei sistemi di speech recognition, questa fase equivale ad una parametrizzazione dello speech e viene eseguita dal front-end. L'approccio tradizionale per la parametrizzazione

dello speech consiste nel source-filter model, che porta all'estrazione di parametri quali linear predictive coding, Mel frequency cepstral coefficients (MFCCs), perceptual linear prediction coefficients ecc. [94, 104, 105]. Tra queste, negli anni le MFCCs si sono attestate come le features di maggior successo ed utilizzo, sia nei sistemi di speech che di speaker recognition, grazie alla loro robustezza all'ambiente, flessibilità ed alla capacità di catturare le caratteristiche specifiche del parlato con una dimensionalità ridotta[105].

Per quanto riguarda la fase di modeling del parlatore, dato che una utterance pronunciata da un parlatore consiste in una sequenza random di frames, il modello più utilizzato per l'identificazione è il modello statistico, come il Gaussian mixture model (GMM) [106, 107].

Per quanto riguarda la classificazione di campioni vocali, il classificatore Bayesiano ottimo garantisce un errore minimo di classificazione identificando lo speaker model che presenta la massima probabilità GMM a posteriori [108].

Molto popolare nella categoria dei classificatori discriminanti, che richiedono dati di training sia per il parlatore target che per gli impostori, è stata l'adozione di tecniche di support vector machines (SVM) [98, 109].

In letteratura, lavori precedenti adottano una combinazione di tali tecniche per effettuare i compiti fondamentali che caratterizzano l'identificazione dello speaker [110, 85, 111, 112, 90, 113, 114, 115].

4.3 Speaker identification tramite rappresentazione Karhunen-Loève transform (KLT) troncata

Come descritto nella sezione precedente, lo step fondamentale in un sistema di speaker identification consiste nell'estrazione delle features caratteristiche del segnale vocale e l'approccio tradizionale si basa su rappresentazione MFCC. Questo tipo di features sono ampiamente usate nell'ambito del riconoscimento vocale perché si sono dimostrate essere particolarmente robuste ed inoltre permettono di catturare le caratteristiche specifiche del parlato con una dimensionalità ridotta. Presentano però inconvenienti se applicate al riconoscimento del parlatore (sia identificazione che verifica) [116, 117]: le differenze tra parlatori date dal pitch [1]

[1] In acustica, la sensazione uditiva che consente di ordinare un suono su una scala che si estenda dal basso verso l'alto (propriamente gradazione di tonalità). Il pitch dipende principalmente dal contenuto di frequenze dello stimolo sonoro, ma anche dalla pressione e dalla forma dello

mismatch è ampiamente mitigata dalle proprietà di smoothing del banco di filtri MFCC [118].

Si è scelto quindi di sperimentare un approccio diverso, basato su rappresentazione Karhunen-Loève transform (KLT) discreta (DKLT) (quindi considerando un sottoinsieme di componenti della trasformata) applicata al log-spectrum del segnale vocale. Questa rappresentazione presenta proprietà di convergenza che garantiscono buone prestazioni in termini di precisione di classificazione, senza influenzare la speaker variability, come invece accade nell'approccio MFCCs. È ben noto che tra le trasformazioni lineari che possono essere usate per l'estrazione di feature e riduzione della dimensionalità, il metodo più conosciuto è quello della espansione DKLT. Sebbene, questa tecnica è stata utilizzata con successo in molti campi di applicazione, in particolare per la riduzione della dimensionalità, questa rappresentazione non è stata ampiamente applicata nell'ambito dello speech recognition. La motivazione principale è legata al fatto che mentre la trasformazione MFCC è funzionalmente indipendente dai dati, la DKLT dipende dalla matrice di covarianza dei dati. Questo implica che l'accuratezza raggiunta nel riconoscimento vocale con training data appartenenti ad un set dato di parlatori, decrementa sensibilmente quando vengono utilizzati testing data derivati da altri parlatori. Questo inconveniente è mitigato usando le MFCCs, dato che questa trasformazione è indipendente dai dati. La conseguenza di ciò, è che date le ottime performance ottenute tramite MFCCs nel campo dello speech recognition [119], le stesse features sono state ampiamente impiegate nell'ambito dello speaker identification [104, 117, 120, 88]. Si vuole quindi dimostrare come l'applicazione delle features ottenute tramite DKLT, presenti un comportamento diverso se applicata nell'ambito della speaker identification rispetto a quanto ottenuto nello speech recognition. In particolare, viene affrontato il problema della speaker identification con brevi sequenze di speech frames, condizione di particolare interesse nei sistemi di audio conferenza, servizi di telefonia e controllo dell'accesso. In queste applicazioni, una latenza di 5 s è considerata come una latenza impossibile da tollerare [121, 122]. I test condotti dimostrano che su utterance di durata inferiore a 2 s, le performance della rappresentazione truncated KLT, applicata all'identificazione di cinque parlatori (lingua italiano), sono sempre migliori di MFCC in termini di classification accuracy. Il framework è

stimolo stesso. Il valore del pitch è dato dalla frequenza del suono sinusoidale e si misura in hertz.

stato inoltre testato su un database composto da un più ampio numero di parlatori (100 speakers - lingua inglese) ma ridotto training set, con sequenze di circa 3.5 s, dimostrando buone capacità di riconoscimento.

L'algoritmo proposto è basato sull'approccio presentato preliminarmente in [123] e poi ulteriormente sviluppato in [124], di cui di seguito si riporta sinteticamente la matematica.

4.3.1 Single Frame Classification

Si indichi con $y[n]$, $n = 0, \ldots, N-1$, un frame che rappresenta lo spettro di potenza del segnale vocale, estratto dalla forma d'onda corrispondente all'uttence presa in considerazione, attraverso un algoritmo di pre-processing che include tre fasi: *pre-emphasis*, *framing*, *log-spectrum* La durata tipica dei frames varia da 20 a 30 ms (solitamente 25 ms) ed ogni frame è generato ogni 10 ms (cioè frames consecutivi di 25 ms sono generati ogni 10 ms con 15 ms di overlap). Il problema di classificazione si riconduce a: dato un set $\mathcal{W}$ di tagged data (training set), tale che ognuno di essi appartiene ad una delle S classi, ed un set $\mathcal{Z}$ di dati (testing set) che devono essere classificati, determinare la regola decisionale che stabilisce a quale classe appartiene l'elemento $\mathrm{y} \in \mathcal{Z}$. Quindi si assume che lo speech acquisito sia diviso in due sets: $\mathcal{W}$ per il training e $\mathcal{Z}$ per il testing.

Un gruppo di S parlatori è descritto dalle probability density functions (pdfs)

$$p_s(\mathrm{y}) = p(\mathrm{y} \mid \theta_s) \quad , \qquad s = 1, 2, \cdots, S \tag{4.1}$$

dove θ_s rappresentano i parametri che devono essere stimati durante il training, $\mathrm{y} \in \mathcal{W}$.

L'obiettivo della classificazione è trovare il modello θ_s che ha la massima probabilità a posteriori per un dato frame y appartenente al testing set $\mathcal{Z}$. Usando il teorema di Bayes e assumendo che $p(\theta_s)$ e $p(\mathrm{y})$ sono indipendenti da S, risulta:

$$\hat{s}(\mathrm{y}) = \underset{1 \le s \le S}{\operatorname{argmax}} \{p(\theta_s \mid \mathrm{y})\} = \underset{1 \le s \le S}{\operatorname{argmax}} \{p_s(\mathrm{y})\} \ . \tag{4.2}$$

Il problema principale nella classificazione Bayesiana è stimare accuratamente la pdf $p_s(\mathrm{y})$. A tal fine, il più generico modello statistico che si può adottare per

il singolo parlatore è il modello GMM [106], dato da questa equazione:

$$p(\mathrm{y} \mid \theta_s) = \sum_{i=1}^{F} \alpha_i \mathcal{N}(\mathrm{y} \mid \mu_i, \mathrm{C}_i) \tag{4.3}$$

dove α_i, $i = 1, \ldots, F$ rappresenta i mixing weights, e $\mathcal{N}(\mathrm{y}|\mu_i, \mathrm{C}_i)$ rappresenta una distribuzione di densità Gaussiana con media μ_i e matrice di covarianza C_i.

$\theta = \{\alpha_1, \mu_1, \mathrm{C}_1, \ldots, \alpha_F, \mu_F, \mathrm{C}_F\}$, (l'indice s viene omesso per ragioni di semplicità di notazione) costituisce il set di parametri non noti da stimare che specificano la mistura Gaussiana.

Una stima di θ, con training data $\mathcal{W}$, può essere ottenuta assumendo dalla *maximum likelihood* (ML)

$$\hat{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmax}} \{\log p(\mathcal{W} \mid \theta)\} \tag{4.4}$$

tuttavia, dato che la (4.4) è difficile da risolvere analiticamente dal momento che (4.4) contiene il log di una somma, la scelta tipica per risolvere il problema della stima ML dei mixture parameters è quella di utilizzare l'algoritmo di expectation maximization (EM).

L'algoritmo di EM si basa sull'interpretazione di $\mathcal{W}$ come un set incompleto di dati e $\mathcal{H}$ come la parte mancante del set completo di dati: $\mathcal{X} = \{\mathcal{W}, \mathcal{H}\}$. La log-likelihood dei dati completi è definita come:

$$\log\left[p\left(\mathcal{W}, \mathcal{H}|\theta\right)\right] = \sum_{\ell=1}^{L} \sum_{i=1}^{F} \mathrm{h}_i^{(\ell)} \log\left[\alpha_i \mathcal{N}(\mathrm{y}^{(\ell)}|\mu_i, \mathrm{C}_i)\right] \ . \tag{4.5}$$

In generale, l'algoritmo di EM calcola una sequenza di parametri stimati $\left\{\hat{\theta}(p)\ , p = 0, 1, \ldots\right\}$, realizzando iterativamente i due steps:

- *Expectation step:* calcola il valore atteso della log-likelihood completa, dati il training set $\mathcal{W}$ e la stima corrente del parametro $\hat{\theta}(p)$. Il risultato è la così chiamata *auxiliary function*

$$Q\left(\theta|\hat{\theta}(p)\right) = E\left\{\log\left[p\left(\mathcal{W}, \mathcal{H}|\theta\right)\right]|\mathcal{W}, \hat{\theta}(p)\right\} \ . \tag{4.6}$$

- *Maximization step:* aggiorna la stima dei parametri

$$\hat{\theta}(p+1) = \underset{\theta}{\operatorname{argmax}} \left\{ Q\left(\theta|\hat{\theta}(p)\right)\right\} \tag{4.7}$$

  massimizzando la Q-function.

Figuereido *et al.* [125] propose un algoritmo non-supervisionato per il learning di un finite mixture da multivariate data, che supera i principali limiti dell'approccio standard EM, i.e., sensibilità all'inizializzazione e scelta del numero F di componenti. Questo algoritmo integra nello stesso framework sia la stima del modello che la selezione delle componenti, con l'abilità di scegliere il numero ottimo di mixture components F secondo un predefinito criterio di minimizzazione.

Tipicamente per un segnale a 8 kHz (16 kHz), il vettore y ha una dimensione di $N = 128$ (256); quindi è necessario un largo ammontare di training data per la stima della pdf $p(\mathrm{y}|\theta)$ e, in ogni caso, con una tale dimensione il problema della stima è impraticabile. Per risolvere questo problema la scelta tipica è quella di ridurre il vettore y in un vettore k_M di dimensione più piccola tramite una trasformazione lineare tale che

$$\mathrm{k}_M = \mathrm{H}\ \mathrm{y}\ , \tag{4.8}$$

dove y è un vettore $N \times 1$, k_M un vettore $M \times 1$, H una matrice $M \times N$, e $M \ll N$. Il vettore k_M rappresenta il feature-vector appartenente ad un appropriato sottospazio a M-dimensioni [126, 127].

Principal component analysis (PCA) [128, 129] è stata provata essere una tecnica eccellente per la riduzione della dimensionalità in molte aree applicative (data compression, image analysis, visualization ecc.).

La PCA di y è derivata dalla KLT, definita dalla coppia di equazioni

$$\mathrm{y} = \Phi\ \mathrm{k}\ , \tag{4.9}$$

$$\mathrm{k} = \Phi^T \mathrm{y}\ , \tag{4.10}$$

dove $\Phi = [\phi_1, \ldots, \phi_N]$ è una matrice $N \times M$ e $\mathrm{k} = [k_1, \ldots, k_N]^T$ è il random vector trasformato. Quindi la PCA per processi stocastici è anche indicata come KLT; più specificatamente, per discrete signal random vectors, è indicata come

discrete KLT (DKLT) [130, 131].

Gli assi principali M sono identificati come quelli corrispondenti agli M maximal eigenvalues $\lambda_j, j = 1, \dots M$ di $\mathrm{R_{yy}}\phi_j = \lambda_j\phi_j$, $j = 1, \dots, N$, dove $\mathrm{R_{yy}}$ è la funzione di autocorrelazione. Così Φ si decompone come $\Phi = [\Phi_M\ ,\ \Phi_\eta]$, e (4.9) può essere riscritta come:

$$\mathrm{y} = \Phi\ \mathrm{k} = \Phi_M\ \mathrm{k}_M + \Phi_\eta\ \mathrm{k}_\eta = \mathrm{x}_M + \eta_\mathrm{y}\ , \tag{4.11}$$

essendo $\Phi_M = [\phi_1, \dots, \phi_M]$ una matrice $N \times M$, k_M un vettore $M \times 1$. In modo simile (4.10) diventa:

$$\begin{bmatrix} \mathrm{k}_M \\ \mathrm{k}_\eta \end{bmatrix} = \begin{bmatrix} \Phi_M^T \\ \Phi_\eta^T \end{bmatrix}\ \mathrm{y}\ . \tag{4.12}$$

In (4.11) il termine

$$\mathrm{x}_M = \Phi_M\ \mathrm{k}_M\ , \tag{4.13}$$

rappresenta l'espansione troncata, ed è equivalente all'approssimazione

$$\mathrm{y} \approx \mathrm{x}_M \quad , \qquad \mathrm{k} \approx \mathrm{k}_T = \begin{pmatrix} \mathrm{k}_M \\ 0 \end{pmatrix}\ , \tag{4.14}$$

In questo modo, dato che k_M è dato da

$$\mathrm{k}_M = \Phi_M^T\ \mathrm{y}\ , \tag{4.15}$$

confrontandolo con (4.8), si ottiene $\mathrm{H} = \Phi_M^T$.

Sulla base dei risultati precedenti, può essere derivato uno schema di classificazione Bayesiana che sia consistente con l'analisi PCA.

Dato un gruppo di S parlatori, si definisca pdfs

$$p_s(\mathrm{k}_T) = p(\mathrm{k}_T \mid \theta_s), s = 1, 2, \dots, S\ , \tag{4.16}$$

dove k_T è il troncamento di k. Conseguentemente la pdf $p_s(\mathrm{k}_T) = p_s(\mathrm{k}_M)\ \delta(\mathrm{k}_\eta)$, con $\delta(\,\cdot\,)$ la Dirac δ-function, rappresenta un'approssimazione della pdf in (4.1). Così (4.2) diventa:

$$\hat{s}(\mathrm{y}) = \underset{1 \le s \le S}{\operatorname{argmax}} \left\{ p_s(\mathrm{k}_M)\ \delta(\mathrm{k}_\eta) \right\} = \underset{1 \le s \le S}{\operatorname{argmax}} \left\{ p_s(\mathrm{k}_M) \right\}\ . \tag{4.17}$$

Confrontando la (4.17) con la (4.2), la dimensionalità del problema di classificazione si riduce da N a M, con $M < N$.

4.3.2 Multi Frame Classification

L'accuratezza nell'identificazione del parlatore può essere considerevolmente migliorata usando una sequenza di frames al posto di un singolo frame. A tal fine, si consideri una sequenza di frames V definita come

$$Y = \{\mathrm{y}^{(1)}, \ldots, \mathrm{y}^{(V)}\} \,, \tag{4.18}$$

dove $\mathrm{y}^{(v)}$ rappresenta il v-th frame. Applicando la (4.17) è possibile determinare la classe a cui ogni frame $\mathrm{y}^{(v)}$ appartiene. Così, gli S sets

$$\mathcal{Z}_s = \left\{\mathrm{y}^{(v)} \mid \mathrm{y}^{(v)} \text{ belongs to class } s\right\}, s = 1, \ldots, S \,, \tag{4.19}$$

sono univocamente determinati.

Dato Y, si definisce lo score di ogni classe s come

$$r_s(Y) \;=\; \sum_{\mathrm{y}^{(v)} \in \mathcal{Z}_s} p(\mathrm{y}^{(v)}) \tag{4.20}$$

dove $p(\mathrm{y}^{(v)})$ rappresenta la probabilità ottenuta dal frame $\mathrm{y}^{(v)}$.

Infine, l'identificazione multi-frame del parlatore è basata su:

$$\hat{s}(Y) = \underset{1 \le s \le S}{\operatorname{argmax}} \{r_s(Y)\} \;. \tag{4.21}$$

4.3.3 DKLT Feature extractor

In un sistema di speaker identification, come già detto, il primo step consiste nell'estrazione delle features caratteristiche del parlatore. La Fig. 4.1 mostra lo schema a blocchi dell'estrattore di features che caratterizza il front-end del sistema proposto.

Il segnale audio in ingresso viene elaborato attraverso il VAD, per poi essere diviso in frame di 25 ms con overlap (200 campioni), con un frame shift di 10 ms (80 campioni); è necessario quindi un blocco di buffering per memorizzare le regioni di overlap tra i segnali. Inoltre, prima che vengano calcolate le features

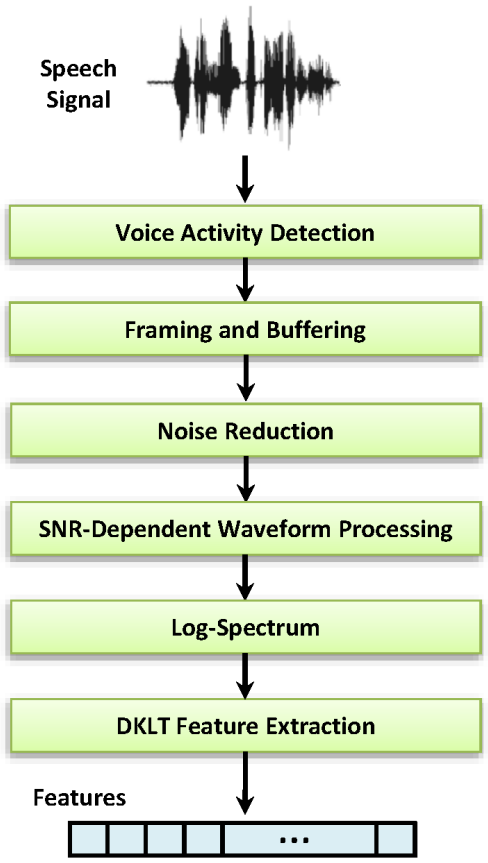


Figura 4.1: Front-end per l'estrazione delle features tramite rappresentazione DKLT.

DKLT, ogni frame viene filtrato tramite un blocco di riduzione del rumore basato su filtro di Wiener. Ulteriori miglioramenti del segnale vengono realizzati dalla fase di SNR-dependent waveform processing, che pesa il frame in ingresso (privo di rumore) secondo la posizione dei suoi smoothed instant energy contour maxima. Gli ultimi due step, il calcolo del logaritmo dello spettro e della trasformazione DKLT, permettono di ottenere le features finali, che verranno utilizzate dall'algoritmo di identificazione.

4.3.4 Risultati sperimentali

Di seguito i numerosi test condotti per dimostrare la validità dell'approccio proposto, suddivisi in due sezioni: *small-scale database* e *large-scale database*.

4.3.4.1 Small-scale database

I primi esperimenti sono stati condotti su un corpus di identificazione basato sulle registrazioni audio acquisite da cinque parlatori, due donne (A,B) e tre uomini (C,D,E). Il materiale usato a questo scopo è riportato in Tab. 4.1 ed è stato estratto da cinque audiolibri in lingua italiana disponibili gratuitamente [67]; tutte le registrazioni sono mono 8 kHz, 16 bit. Questo database sarà indicato come small-scale database (SSD) e chiamato DBT.

Tabella 4.1: Materiale audio utilizzato per la creazione del corpus di identificazione. Sorgente: *Liber Liber* (`http://www.liberliber.it/`). Il materiale è stato utilizzato sia a scopo di traninig che di testing.

Speaker	Gender	Audiobook	Chapter	Duration [s]
A	F	L. Bertelli, "Il giornalino di Gianburrasca"	I	761
B	F	A. Manzoni, "I promessi Sposi"	I	2593
C	M	L. Pirandello, "Fu Mattia Pascal"	I	251
D	M	E. Salgari, "Le tigri di Mompracem"	I	838
E	M	G. Verga, "I Malavoglia"	I	1162

La Tab. 4.2 mostra la consistenza del database DBT in termini di numero di frames usati per ogni speaker, Da questo database sono state ricavate tre partizioni DB1, DB2, e DB3, con una diversa percentuale di materiale assegnato al training ed al testing, come specificato nella Tab. 4.2.

Sono stati realizzati diversi esperimenti, variando il numero di componenti DKLT del modello GMM, e considerando le tre differenti partizioni in modo da valutare gli effetti della quantità di materiale usato per il training sul risultato della classificazione.

Considerando frames appartenenti ai testing sets, si è valutato il numero di occorrenze per ogni riconoscimento all'uscita del classificatore, in modo da ottenere una matrice di confusione per ogni esperimento di speaker identification. La matrice di confusione risultante è riportata nelle Tabs. 4.3, 4.4, e 4.5 per il caso single-frame, e nelle Tabs. 4.6, 4.7, e 4.8 per il caso multi-frame; si osserva così come le performance migliorino quando vengono presi in considerazione $V = 100$ frame consecutivi (corrispondenti ad un sequenza vocale di 1s).

Tabella 4.2: Consistenza (in termini di numero di frames) del database DBT e delle sue partizioni (DB1, DB2, and DB3) usate per le vatutazioni sperimentali.

Data	DBT	DB1 (80/20)		DB2 (50/50)		DB3 (20/80)	
Speaker		train	test	train	test	train	test
A	58903	47122	11781	29451	29452	11780	47123
B	195591	156472	39119	97795	97796	39118	156473
C	18867	15093	3774	9431	9434	3773	15094
D	63713	50970	12743	31856	31857	12742	50971
E	91253	73002	18251	45626	45627	18250	73003
Total	428327	342659	85668	214161	214166	85663	342664

Per avere informazioni aggiuntive sulle prestazioni del metodo, dalla matrice di confusione sono stati estratti i seguenti parametri:

$$\text{sensitivity} = \text{TP}/(\text{TP} + \text{FN}) \tag{4.22}$$

$$\text{specificity} = \text{TN}/(\text{TN} + \text{FP}) \tag{4.23}$$

$$\text{precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{4.24}$$

$$\text{accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{4.25}$$

dove TP sono i veri positivi (gli elementi della diagonale della matrice di confusione), FN sono i falsi negativi (la somma degli altri elementi sulla stessa riga della matrice di confusione), FP sono i falsi positivi (la somma degli altri elementi sulla stessa colonna della matrice di confusione), e TN sono i veri negativi (la somma degli elementi sulle altre righe e colonne della matrice di confusione). Si è considerata, inoltre la *overall sensitivity* definita come il rapporto della somma degli elementi sulla diagonale (TP) rispetto alla somma di tutti gli altri elementi della matrice di confusione. Al fine di investigare gli effetti della complessità del modello in termini di componenti DKLT, la overall classifier sensitivity è stata calcolata in funzione della lunghezza della sequenza per valori di M variabili nel range 5-40. I risultati sono riportati in Fig. 4.2, prendendo in considerazione la partizione DB1.

Tabella 4.3: Matrice di confusione: single-frame - 12 componenti DKLT, partizione DB1.

Input	Recognized				
	A	B	C	D	E
A	9885	688	349	312	547
B	1927	35226	570	720	676
C	200	100	2313	573	588
D	812	384	2113	6831	2603
E	1165	429	2443	2648	11566

Tabella 4.4: Matrice di confusione: single-frame - 12 componenti DKLT, partizione DB2.

Input	Recognized				
	A	B	C	D	E
A	23729	2166	1138	823	1596
B	4027	89214	1401	1810	1344
C	427	270	5444	2185	1108
D	1925	1151	5590	17199	5992
E	2919	1232	6265	8400	26811

Tabella 4.5: Matrice di confusione: single-frame - 12 componenti DKLT, partizione DB3.

Input	Recognized				
	A	B	C	D	E
A	36552	5135	1682	1395	2359
B	6104	143531	2018	2973	1847
C	708	557	9132	2671	2026
D	3364	2183	10957	24372	10095
E	6186	2587	11127	12613	40490

Tabella 4.6: Matrice di confusione: multi-frame ($V = 100$) - 12 componenti DKLT, partizione DB1.

Input	Recognized				
	A	B	C	D	E
A	117	0	0	0	0
B	0	391	0	0	0
C	0	0	36	0	1
D	0	0	1	125	1
E	0	0	0	0	182

Tabella 4.7: Matrice di confusione: multi-frame ($V = 100$) - 12 componenti DKLT, partizione DB2.

Input	Recognized				
	A	B	C	D	E
A	294	0	0	0	0
B	0	977	0	0	0
C	1	0	89	4	0
D	3	0	12	301	2
E	4	0	1	4	447

Tabella 4.8: Matrice di confusione: multi-frame ($V = 100$) - 12 componenti DKLT, partizione DB3.

Input	Recognized				
	A	B	C	D	E
A	471	0	0	0	0
B	0	1564	0	0	0
C	0	0	148	1	1
D	11	0	31	436	31
E	42	0	10	20	658

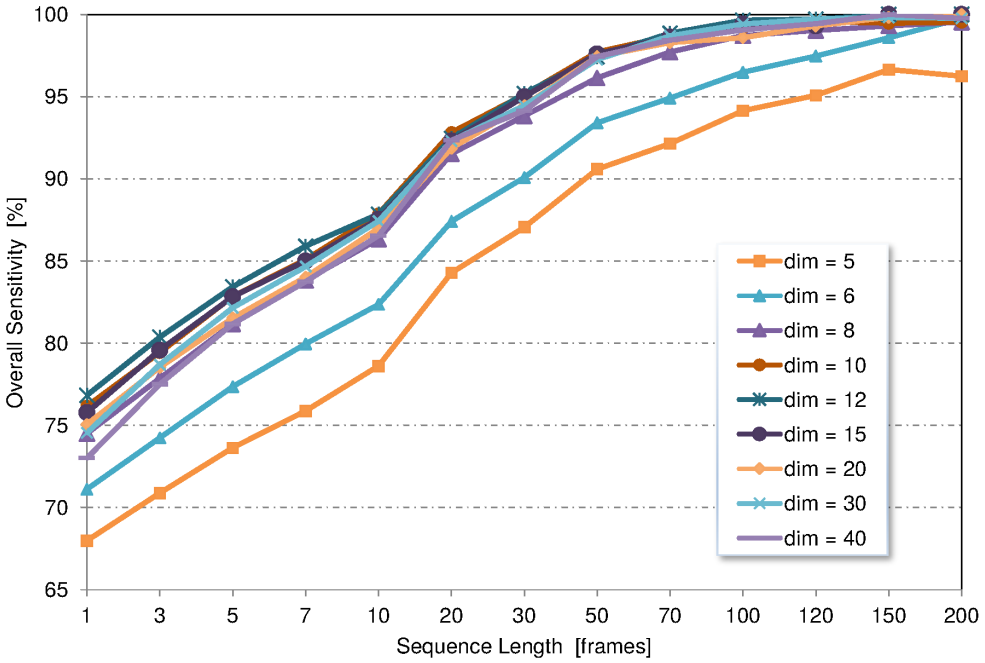


Figura 4.2: Overall classifier performance in funzione della lunghezza della sequenza, per diversi valori del numero di componenti DKLT e usando la partizione DB1.

Calcolando la average overall sensitivity con una lunghezza della sequenza variabile da 1 a 100, si ottengono i risultati in Fig. 4.3. Quindi, si sceglie come valore ottimo per il numero di componenti DKLT per il modello GMM il valore di $M = 12$, che corrisponde al massimo valore (92.13%) della average overall sensitivity.

I risultati delle simulazioni effettuate con il valore ottimo $M = 12$ delle componenti DKLT relativi agli indici (4.22)–(4.25) sono riportati in Tab. 4.9 nel caso single-frame ed in Tab. 4.10 nel caso multi-frame (sequenza di $V = 100$ frames consecutivi), per ognuna delle tre partizioni.

Per mostrare l'effetto della lunghezza della sequenza sul processo di speaker identification, le Figs. 4.4 (a)–(c) mostrano la sensitivity in funzione del numero V di frames per diversi valori delle componenti DKLT, rispettivamente $M = 20$, 15, and 12, usando la partizione DB1.

Per confrontare le performance del sistema proposto con lo stato dell'arte, sono stati condotti alcuni test utilizzando le features MFCC e per rendere questi esperimenti comparabili con quelli realizzati utilizzando le componenti DKLT,

Tabella 4.9: Analisi delle performance della tecnica proposta per le tre diverse partizioni, considerando $M = 12$ componenti DKLT ed usando un singolo frame.

Speaker	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
	DB1 (80/20)			
A	83.91	94.45	70.66	93.00
B	90.05	96.56	95.65	93.59
C	61.29	93.31	29.70	91.90
D	53.61	94.17	61.63	88.13
E	63.37	93.45	72.38	87.04
	DB2 (50/50)			
A	80.57	94.97	71.85	92.99
B	91.22	95.86	94.88	93.74
C	57.71	92.97	27.44	91.42
D	53.99	92.75	56.54	86.98
E	58.76	94.04	72.76	86.53
	DB3 (20/80)			
A	77.57	94.46	69.08	92.14
B	91.73	94.38	93.21	93.17
C	60.50	92.13	26.15	90.74
D	47.82	93.26	55.36	86.50
E	55.46	93.95	71.26	85.75

Tabella 4.10: Analisi delle performance della tecnica proposta per le tre diverse partizioni, considerando $M = 12$ componenti DKLT ed usando 100 frames.

Speaker	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
	DB1 (80/20)			
A	100.00	100.00	100.00	100.00
B	100.00	100.00	100.00	100.00
C	97.30	99.88	97.30	99.77
D	98.43	100.00	100.00	99.77
E	100.00	99.70	98.91	99.77
	DB2 (50/50)			
A	100.00	99.57	97.35	99.63
B	100.00	100.00	100.00	100.00
C	94.68	99.36	87.25	99.16
D	94.65	99.56	97.41	98.83
E	98.03	99.88	99.55	99.49
	DB3 (20/80)			
A	100.00	98.21	89.89	98.45
B	100.00	100.00	100.00	100.00
C	98.67	98.75	78.31	98.74
D	85.66	99.28	95.40	97.25
E	90.14	98.81	95.36	96.96

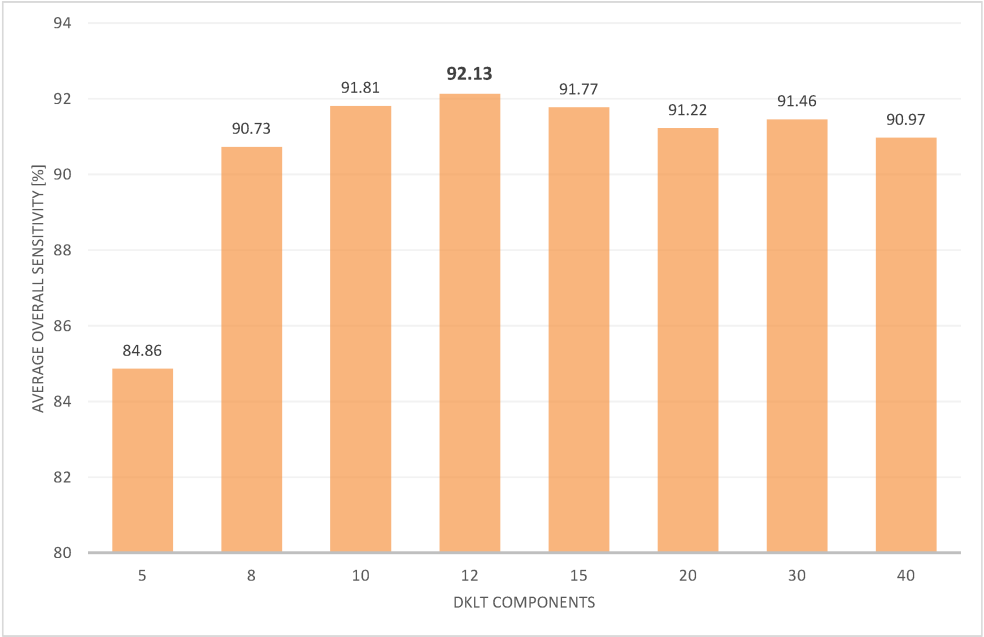


Figura 4.3: Overall classifier performance (mediato sulla lunghezza delle sequenze), per diversi valori del numero di componenti DKLT e usando la partizione DB1.

si è utilizzato lo stesso front-end in entrambi i casi. Le performance ottenute considerando 13 features MFCC, per ognuna delle tre partizioni, sono riportati in Tab. 4.11 per un singolo frame ed in Tab. 4.12 per sequenze di 100 frames. Confrontando questi risultati con quelli in Tabs. 4.9 e 4.10, relativi a $M = 12$ componenti DKLT, è evidente che il metodo proposto si comporta meglio del metodo basato su features MFCC, sia nel caso single-frame che nel caso multi-frame speaker identification.

Tipicamente al vettore di features MFCC composto da 13 componenti (12 MFCCs, 1 energia), vengono aggiunte le features delta e double delta [38], per un vettore risultante di 26 e 39 componenti per ogni frames. Al fine di confrontare la meglio i due approcci, sono stati quindi condotti ulteriori esperimenti utilizzando $M = 13$, 26, e 39 componenti DKLT.

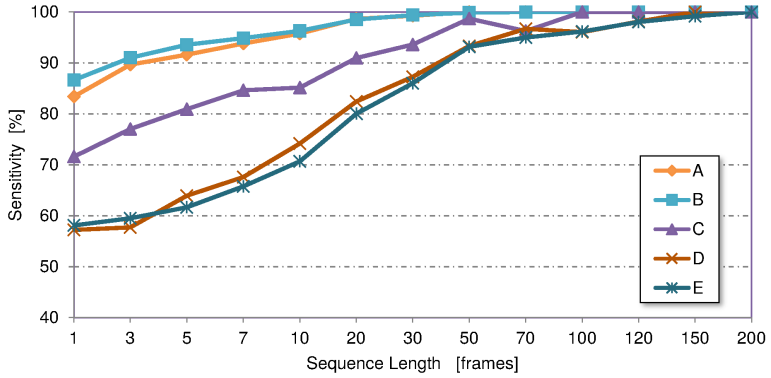
Le Figs. 4.5 (a)–(c) riportano le overall sensitivities per entrambi i metodi, in funzione della lunghezza della sequenza che viene fatta variare da 1 a 200 frames (fino a 2 s) e per le tre differenti partizioni, Anche in questo caso, l'approccio DKLT si dimostra essere migliore di quello basato su MFCCs.

Tabella 4.11: Analisi delle performance delle features MFCC per le tre differenti partizioni ed usando un singolo frame.

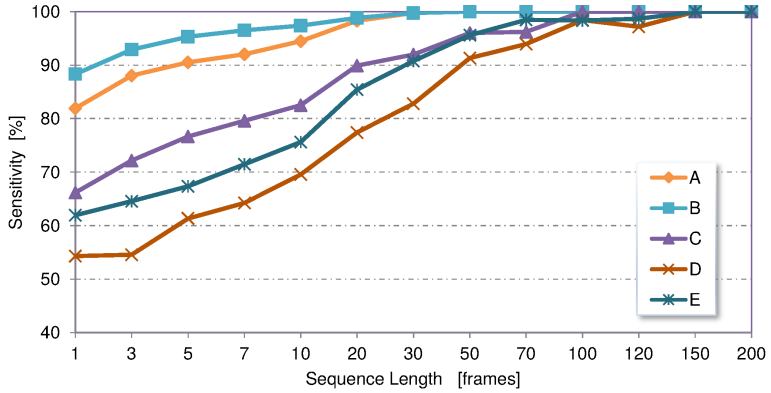
Speaker	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
	DB1 (80/20)			
A	71.34	89.99	53.18	87.42
B	72.46	93.03	89.73	83.64
C	49.95	89.67	18.23	87.92
D	51.08	90.28	47.87	84.45
E	44.75	90.85	56.98	81.03
	DB2 (50/50)			
A	71.59	90.45	54.44	87.85
B	75.85	93.03	90.14	85.19
C	45.56	90.71	18.44	88.72
D	52.13	90.22	48.24	84.56
E	45.95	91.44	59.23	81.75
	DB3 (20/80)			
A	64.02	91.62	54.93	87.83
B	74.71	93.50	90.62	84.92
C	44.76	88.78	15.53	86.84
D	49.53	88.62	43.19	82.80
E	43.31	90.67	55.68	80.58

Tabella 4.12: Analisi delle performance delle features MFCC per le tre differenti partizioni ed usando 100 frames.

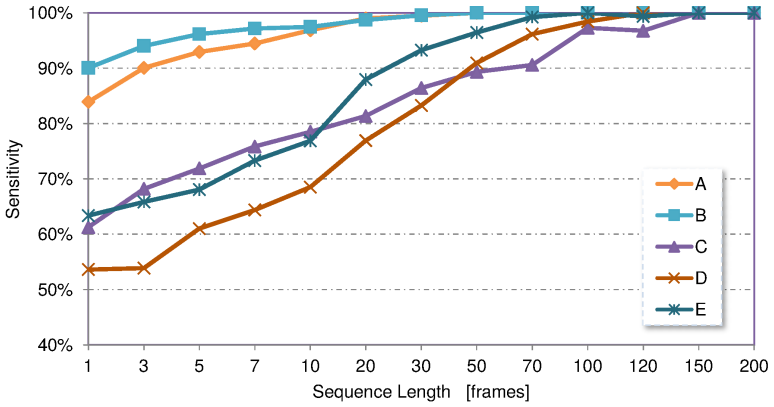
Speaker	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
	DB1 (80/20)			
A	100.00	96.34	81.25	96.84
B	93.35	99.78	99.73	96.84
C	94.59	98.65	76.09	98.48
D	94.49	98.35	90.91	97.78
E	87.91	99.11	96.39	96.72
	DB2 (50/50)			
A	100.00	97.62	86.98	97.94
B	96.72	99.83	99.79	98.41
C	85.11	99.56	89.89	98.92
D	96.23	97.69	87.93	97.48
E	88.38	99.17	96.64	96.87
	DB3 (20/80)			
A	99.36	98.68	92.31	98.77
B	98.02	99.57	99.48	98.86
C	84.67	98.78	76.05	98.16
D	92.14	96.54	82.28	95.88
E	82.88	98.74	94.68	95.36



(a)

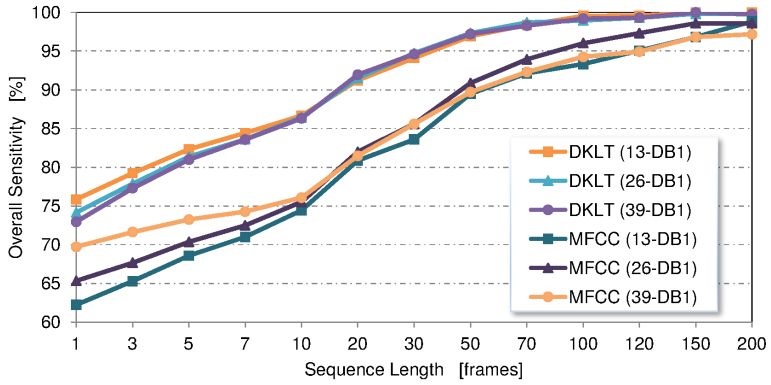


(b)

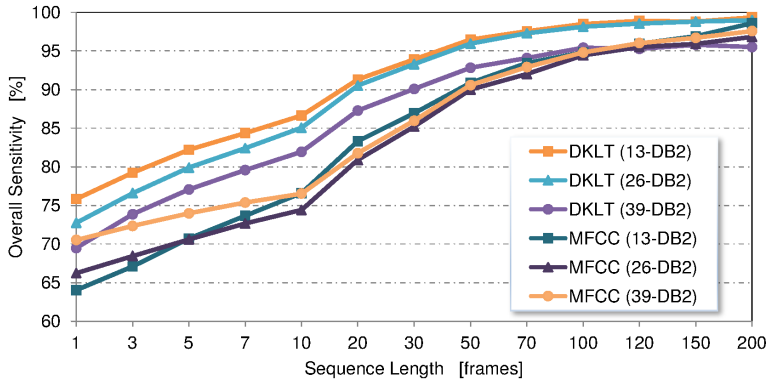


(c)

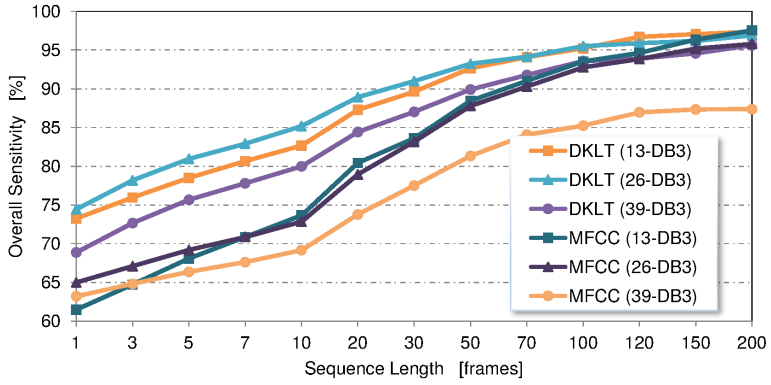
Figura 4.4: Performance del classificatore in funzione della lunghezza della sequenza, con *(a)* $M = 20$, *(b)* $M = 15$, e *(c)* $M = 12$ componenti DKLT ed usando la partizione DB1.



(a)



(b)



(c)

Figura 4.5: Overall sensitivity usando le features MFCC e DKLT ($M = 13, 26, 39$), in funzione della lunghezza della sequenza, per le tre diverse partizioni *(a)* DB1, *(b)* DB2, e *(c)* DB3.

4.3.4.2 Large-scale database

Per indagare ulteriormente le performance del sistema, si sceglie di sperimentare il framework proposto anche su una lingua diversa dalla lingua madre e su una popolazione di parlatori più ampia. A questo scopo, si sceglie di utilizzate il noto database TIMIT [132]. Il TIMIT speech corpus è una collezione di sentenze foneticamente bilanciate, contenente 10 utterances per 630 speakers provenienti da 8 regioni negli USA dialetticamente diverse. Si è scelto di utilizzare i primi 100 speakers (67 uomini e 33 donne) corrispondenti al dialetto del New England (Dialect 1) e del Northern (Dialect 2), indicando questo database come large-scale database (LSD). La Tab. 4.13 riporta la consistenza del database usato per gli esperimenti TIMIT-based. Per ogni speaker, sono state utilizzate 10 frasi con una

Tabella 4.13: Consistenza del database usato per gli esperimenti TIMIT-based.

Speakers	Total		Dialect 1		Dialect 2	
	M	F	M	F	M	F
100	67	33	22	15	45	18
50	24	26	22	15	2	11
30	15	15	15	15	–	–
20	5	15	5	15	–	–
10	–	10	–	10	–	–
5	–	5	–	5	–	–
3	–	3	–	3	–	–
1	–	1	–	1	–	–

lunghezza media di 3.10 s (lunghezza minima: 1.09 s, lunghezza massima: 7.44 s) ad una frequenza di campionamento di 16 kHz (16-bit samples). Queste sequenze sono state suddivise in overlapping frames di 25 ms (400 campioni), con un frame shift di 10 ms (160 campioni), ottenendo così per ogni sentenza un numero medio di 460 campioni. Anche in questo caso il database è stato partizionato per ottenere una database di training (80%) ed uno di testing (20%).

La Fig. 4.6 riporta le performance del framework in termini di overall sensitivity per differenti lunghezze delle sequenze che variano da 1 (25 ms) a 350 (3.5 s) frame consecutivi con overlap. I risultati sono stati ottenuti per differenti speaker

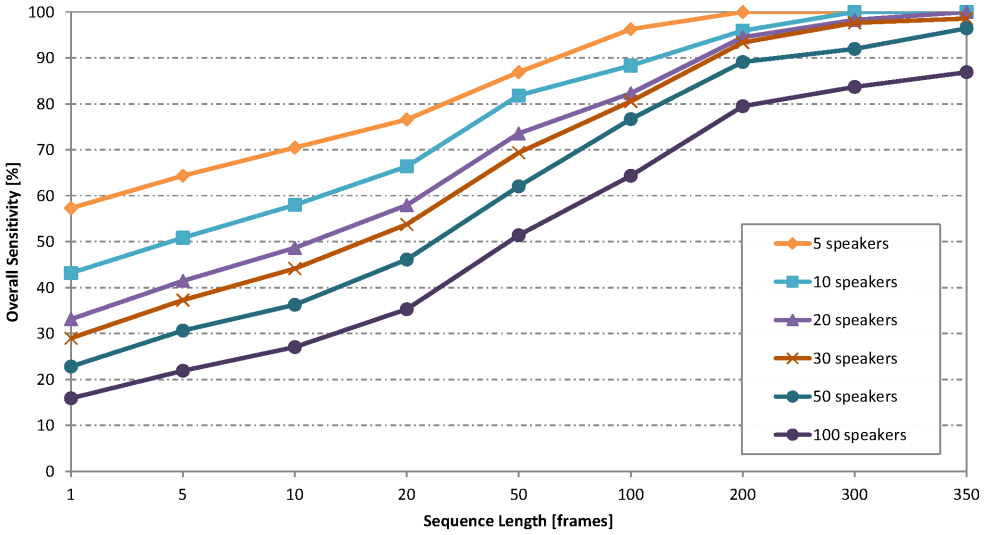


Figura 4.6: Performance del classificatore in funzione della lunghezza della sequenza, con $M = 20$ componenti DKLT, utilizzando il TIMIT corpus.

pools, cioè 5, 10, 20, 30, 50, e 100 persone con $M = 20$ componenti DKLT. Per sequenze di 350 frames, si ottiene una overall sensitivities di 100%, 94.64%, e 84.16%, rispettivamente per 30, 50, e 100 speakers.

Al fine di indagare l'effetto della complessità del modello, in termini di componenti DKLT usate, sulle performance ottenute, viene calcolata la overall classifier sensitivity in funzione della lunghezza della sequenza per una popolazione di riferimento di 50 parlatori e per valori di M variabili da 8 a 30; i risultati sono mostrati in Fig. 4.7.

Si ottiene che il valore ottimo di componenti DKLT da utilizzare in questo caso è $M = 22$, che corrisponde ad una average overall sensitivity di 63.43%. Con questo valore ($M = 22$ componenti DKLT), vengono nuovamente calcolate le performance del classificatore per differenti lunghezze delle sequenze che variano da 1 (25 ms) a 350 (3.5 s) frame consecutivi con overlap e per 5, 10, 20, 30, 50, e 100 speakers. Per sequenze di 350 frames, si ottiene una best overall sensitivities di 100%, 99.11%, e 91.86%, rispettivamente per 30, 50, e 100 speakers.

Conclusioni Il metodo proposto, volto a superare i limiti che l'approccio basato su MFCCs mostra quando applicato al campo della speaker identification, ha dimostrato produrre buoni risultati (sempre migliori di quelli ottenuti applicando

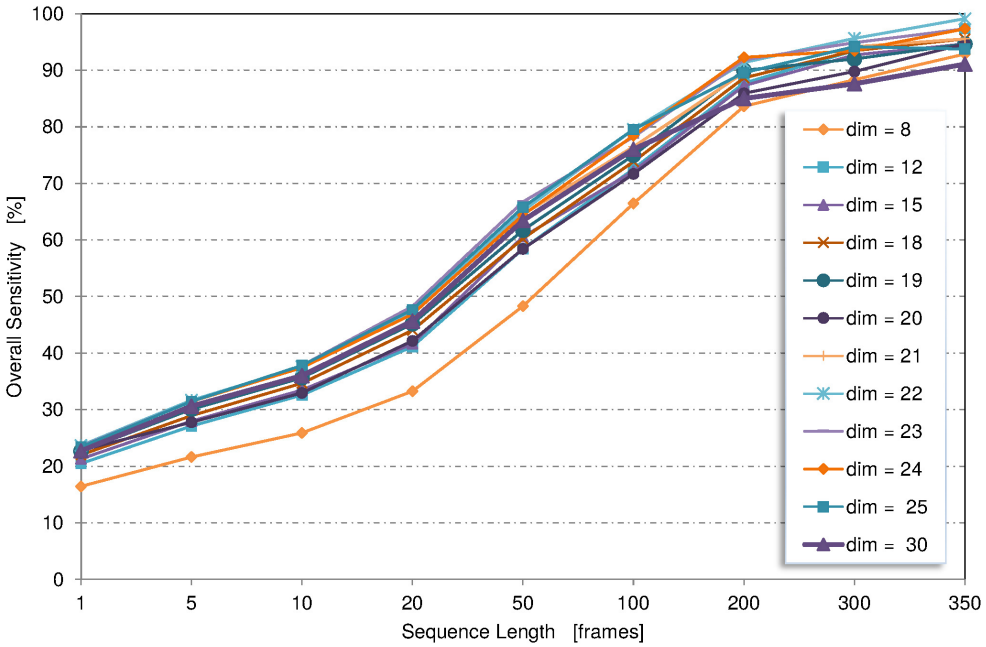


Figura 4.7: 50 speakers overall classifier performance in funzione della lunghezza della sequenza, per diversi valori del numero di componenti DKLT, utilizzando il TIMIT corpus.

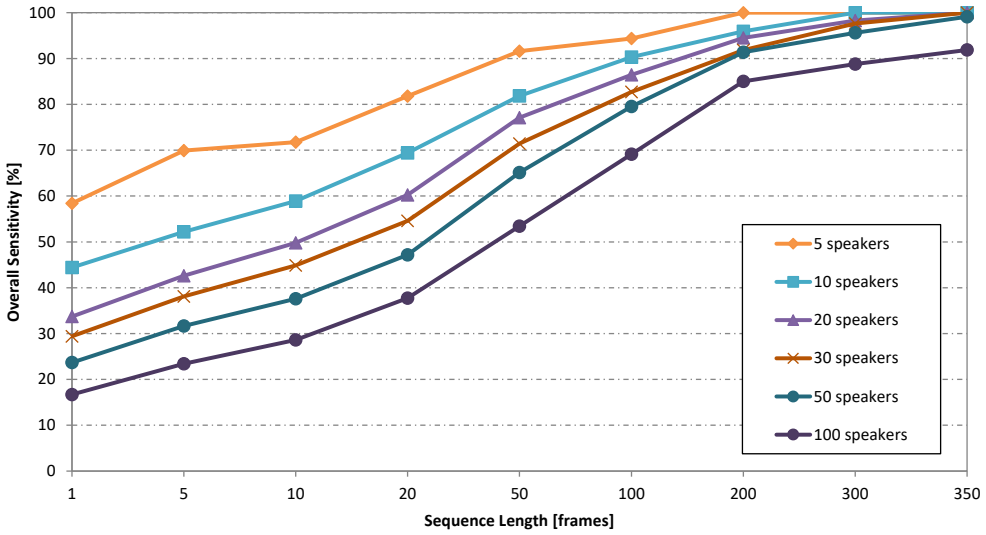


Figura 4.8: Performance del classificatore in funzione della lunghezza della sequenza, con $M = 22$ componenti DKLT, utilizzando il TIMIT corpus.

le features MFCCs) utilizzando brevi sequenze di speech frames sia applicato ad un limitato (cinque parlatori) che ad un più ampio (TIMIT corpus) database.

4.4 Robust speaker identification applicata ad uno scenario multi-speakers

Il sistema proposto vuole rispondere alla domanda "qualcuno sta parlando?", se sì, "chi?", e vuole rispondere a questo quesito nell'ambito di un meeting scenario nel quale gli interlocutori sono sicuramente più di uno e possono eventualmente sovrapporsi l'uno all'altro. Il sistema ha quindi l'obiettivo di identificare correttamente il parlatore corrente a partire da brevi sequenze di speech frames e nello scenario descritto. Questo tipo di task viene svolto sia nel campo della speaker identification che in quello della speaker diarization.

Nello scenario classico della speaker identification, il sistema deve essere in grado di identificare correttamente un utente che sta parlando da solo in un intervallo di tempo. Nel diverso scenario del meeting [122, 121, 133], il parlato di un partecipante può bruscamente variare o essere affetto da overlap da parte di un altro speaker. In particolare l'overlapping speech può degradare notevolmente le prestazioni di un sistema di speaker identification Queste problematiche sono comuni nel campo della Speaker Diarization il cui obiettivo è quello di segmentare l'audio in regioni omogenee al fine di rispondere alla domanda "chi parla quando?". Tuttavia nei sistemi di diarizzazione, l'uscita è limitata a etichettare le speaker region con numeri o lettere, senza rilevare l'identità di chi parla. Questo obiettivo viene raggiunto senza training a priori di modelli specifici, quindi molti di questi sistemi lavorano in modo non-supervisionato.

Il sistema proposto è volto a dimostrate l'efficacia dell'algoritmo illustrato nel capitolo 4.3 in uno scenario critico di più parlatori con overlap.

Il sistema proposto è stato sviluppato in [134] e di seguito se ne riportano i risultati.

4.4.1 Risultati sperimentali

I test sono stati condotti anche in questo caso su due differenti corpora: *i)* un database chiamato DBT che corrisponde al database descritto nella sezione 4.3.4.1 ma manipolato in modo da creare sinteticamente delle regioni di overlap

tra i parlatori; *ii)* il noto AMI Meeting Corpus, ampiamente impiegato nella verifica della bontà dei sistemi di diarizzazione.

4.4.1.1 Small-scale database

Il primo set di esperimento è stato realizzato utilizzato il database DBT composto da un ampio numero di registrazioni appartenenti a due donne (A, B) e tre uomini (C, D, E), come riportato nelle Tabs. 4.1, 4.2 già descritte nella sezione 4.3.4.1. Lo scenario di meeting è stato simulato inserendo 45 segmenti audio estratti dal database *liber liber* [67], per una traccia audio della durata di 20 minuti ed alternando i cinque parlatori con turni di breve durata come illustrato in Fig. 4.9. In particolare, sono state generate due tracce audio: nella prima i segmenti audio si susseguono senza overlap, mentre nella seconda viene simulato un overlap del 20% per testare la robustezza dell'algoritmo.

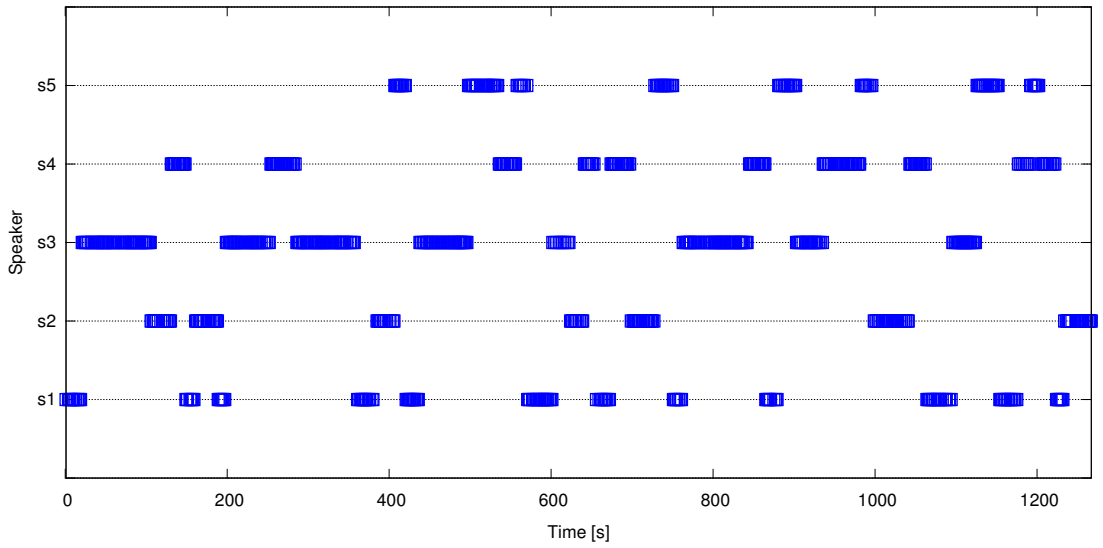


Figura 4.9: Meeting timeline.

La metrica tipicamente adottata per valutare le performance di un sistema di diarizzazione è data dal Diarization Error Rate (DER). Questo indice di valutazione è stato introdotto dal NIST nel 2000 nell'ambito della valutazione dei sistemi di speaker recognition [135] e dei task di speaker segmentation [136]. Il DER è definito come il rapporto tra il tempo in cui lo speaker preso in esame è ricosciuto erroneamente rispetto al tempo totale relativo allo stesso parlatore. Definendo i seguenti errori:

- *Speaker assignment errors* (E_{spkr}): caso in cui viene rilevato lo speech ma non viene assegnato al parlatore corretto; percentuale di tempo per cui l'ID classificato è assegnato al parlatore sbagliato.

- *Missed detections* (E_{miss}): percentuale di tempo per cui un segmento ipotizzato come non-speech è assegnato al parlatore di riferimento.

- *False alarm detections* (E_{fa}): percentuale di tempo per cui un segmento ipotizzato come appartenente al parlatore è assegnato ad un segmento di non-speech.

il DER è quindi dato da:

$$DER = E_{spkr} + E_{miss} + E_{fa} \tag{4.26}$$

Le Figs. 4.10– 4.11 mostrano il valore del DER in funzione del numero di frames considerati nell'algoritmo di speaker identification e per i quattro modelli precedentemente definiti, rispettivamente per un segnale senza o con overlap. Si evince che il DER decrementa drasticamente all'aumentare della lunghezza dei frames, con minime differenze tra i modelli. Inoltre, grazie alla robustezza

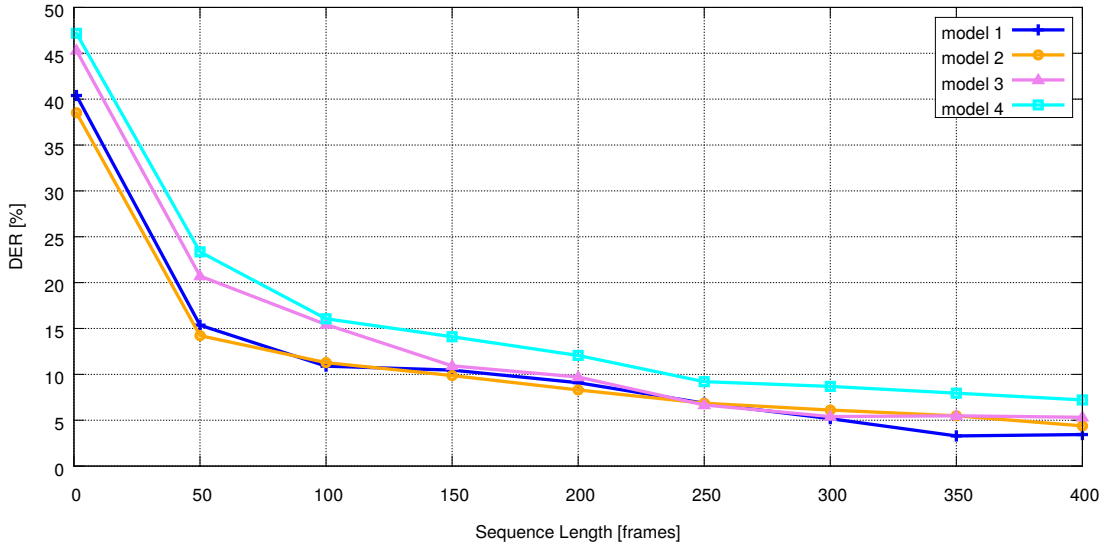


Figura 4.10: DER valutato su database DBT senza overlap in funzione della lunghezza della sequenza, per diversi modelli.

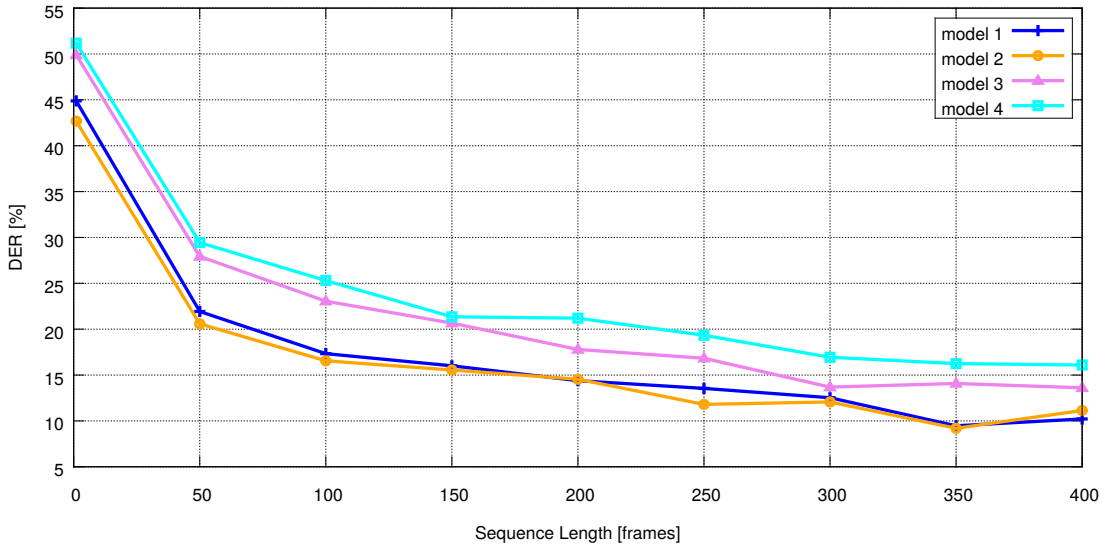


Figura 4.11: DER valutato su database DBT con overlap del 20% in funzione della lunghezza della sequenza, per diversi modelli.

dell'algoritmo, non si riscontra una degradazione significativa delle performance nel caso di overlap.

Tabs. 4.14, 4.15 riportano i valori dei tre parametri (E_{spkr}, E_{miss}, E_{fa}) che contribuiscono al valore del DER.

Tabella 4.14: Speaker identification performance per differenti modelli di training e lunghezza della sequenza.

Frames	Model 1				Model 2				Model 3				Model 4			
	E_{spkr}	E_{miss}	E_{fa}	DER	E_{spkr}	E_{miss}	E_{fa}	DER	E_{spkr}	E_{miss}	E_{fa}	DER	E_{spkr}	E_{miss}	E_{fa}	DER
	[%]				[%]				[%]				[%]			
400	2.19	1.25	0.00	3.44	2.19	1.88	0.31	4.38	3.13	1.56	0.62	5.32	1.88	4.70	0.62	7.21
350	1.64	1.09	0.54	3.29	2.19	2.74	0.54	5.48	3.01	1.92	0.54	5.48	3.29	4.38	0.27	7.95
300	2.82	2.11	0.23	5.17	2.35	2.82	0.94	6.11	3.05	1.64	0.70	5.40	3.52	3.99	1.17	8.69
250	2.15	4.11	0.58	6.85	1.95	4.31	0.58	6.85	2.54	3.52	0.58	6.66	3.91	3.91	1.37	9.20
200	3.13	5.01	0.94	9.09	2.35	4.85	1.09	8.30	4.23	3.60	1.88	9.71	5.01	4.85	2.19	12.06
150	3.52	5.05	1.88	10.46	2.82	4.58	2.46	9.87	4.34	3.40	3.17	10.93	4.58	5.64	3.87	14.10
100	4.31	3.91	2.66	10.89	4.15	3.36	3.76	11.28	6.11	3.44	5.87	15.43	6.58	4.31	5.17	16.06
50	7.48	2.93	4.93	15.36	6.26	2.82	5.13	14.22	10.30	2.50	7.9	20.72	11.48	3.60	8.26	23.35
1	25.18	4.31	10.90	40.40	23.15	4.10	11.25	38.51	27.76	3.89	13.63	45.30	28.74	4.67	13.76	47.18

Inoltre, la Fig. 4.12 mostra la validità dei risultati ottenuti, sovrapponendo la classificazione ottenuta alla timeline della traccia originale.

Tabella 4.15: Speaker identification performance per differenti modelli di training e lunghezza della sequenza - overlapping speech.

Frames	Model 1				Model 2				Model 3				Model 4			
	E_{spkr}	E_{miss}	E_{fa} [%]	DER	E_{spkr}	E_{miss}	E_{fa} [%]	DER	E_{spkr}	E_{miss}	E_{fa} [%]	DER	E_{spkr}	E_{miss}	E_{fa} [%]	DER
400	8.66	1.23	0.30	10.21	8.66	2.16	0.30	11.14	12.38	0.61	0.61	13.61	11.45	3.71	0.92	16.09
350	8.39	1.08	0.00	9.47	7.85	1.35	0.00	9.20	12.18	0.81	1.08	14.08	11.91	3.25	1.08	16.25
300	9.98	2.08	0.46	12.53	7.89	3.71	0.46	12.07	10.67	2.08	0.92	13.69	12.07	4.17	0.69	16.94
250	9.86	3.09	0.58	13.54	7.93	3.28	0.58	11.80	13.54	2.12	1.16	16.83	14.12	3.86	1.35	19.34
200	8.97	4.02	1.39	14.39	8.82	4.79	0.92	14.54	12.69	2.32	2.78	17.79	13.92	4.95	2.32	21.20
150	10.56	3.36	2.08	16.01	9.05	4.52	1.97	15.55	14.39	2.20	4.06	20.66	14.16	3.59	3.59	21.35
100	10.60	3.17	3.55	17.33	10.44	2.94	3.17	16.55	14.54	1.70	6.80	23.05	14.77	4.02	6.50	25.30
50	13.69	2.39	5.84	21.93	12.76	2.66	5.14	20.58	16.67	1.66	9.59	27.93	17.79	2.47	9.16	29.44
1	29.18	4.06	11.64	44.89	27.26	4.03	11.38	42.68	31.57	3.37	14.96	49.91	32.29	4.51	14.37	51.18

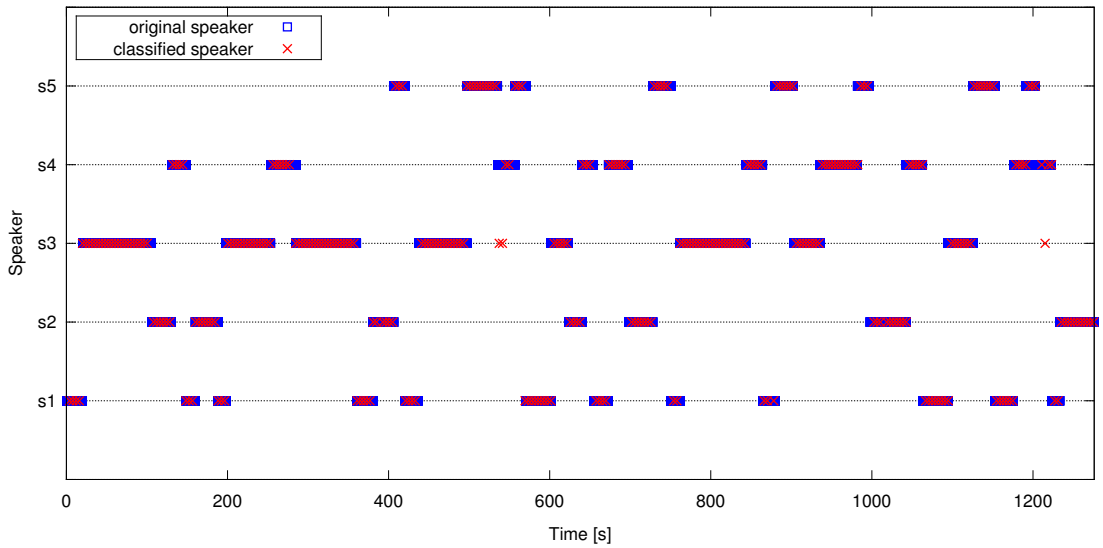


Figura 4.12: Speaker classification timeline.

4.4.1.2 Large-scale database

Il secondo set di esperimenti riguarda la valutazione del sistema di speaker identification su meeting audio data estratti dal database *AMI Meeting Corpus* [137]. AMI consiste in un progetto multidisciplinare che ha lo scopo di sviluppare tecnologie orientate al meeting browsing; a questo scopo il progetto mette a disposizione un corpus di 100 ore di parlato, relativo a meetings composti da quattro partecipanti [138]. Si è considerato un sottoinsieme dell'AMI Corpus,

composto da 20 meetings appartenenti all'IDIAP subset ('IS' meetings). Questo sottoinsieme comprende 38 meetings, ognuno dei quali coinvolge quattro partecipanti ed ha una durata variabile da 13 a 40 minuti. Viene scelto un meeting a caso per l'addestramento dei modelli dei quattro parlatori coinvolti ed una finestra di lunghezza variabile da 1 a 4 secondi. La quantità di segnale audio (per parlatore) utilizzato per addestrare i modelli è riportata nella Tab. 4.16. In aggiunta, è stato realizzato un modello (60 s) relativo al rumore di fondo della stanza. Come mostrano i risultati, utilizzando modelli addestrati con soli 60 s di parlato, il sistema è in grado di ottenere buone performance.

Tabella 4.16: DER valutato sull'IDIAP AMI Corpus utilizzando un modello addestrato su un dataset limitato e brevi sequenze di speech frame.

Frames	Model		
	30 [s]	60 [s]	90 [s]
	DER [%]		
400	12.26	11.90	12.98
300	14.87	12.71	14.33
200	18.93	15.87	16.59
100	28.76	25.96	24.61

Conclusioni Si dimostra quindi l'effettiva robustezza dell'approccio DKLT anche in condizioni critiche: più parlatori in un meeting scenario, nel quale la voce del singolo parlatore può variare bruscamente e subire la sovrapposizione da parte di un altro parlatore coinvolto nello scenario.

4.5 Implementazione di un sistema combinato di speech recognition/speaker identification per la personalizzazione dell'ambiente domestico

Una volta testata la bontà del metodo descritto nella sezione 4.3 [123, 124], si è deciso di integrare il sistema DSR descritto nel capitolo 3 [53], con l'algo-

ritmo di speaker identification nell'ottica di realizzare una possibile applicazione per la domotica. Questo binomio ha portato alla realizzazione di un sistema combinato di speech recognition/speaker identification per la personalizzazione dell'ambiente domestico, che può essere ampiamente utilizzato anche in altri contesti (car customization settings ecc.). Il sistema è così strutturato: alla prima installazione vengono addestrati i modelli acustici caratteristici di ogni parlatore appartenente all'ambiente domestico e salvati in un database; successivamente ad ogni utilizzo l'utente pronuncia un comando appartenente al set di comandi per il controllo domotico ed il sistema riconosce parallelamente "cosa è stato detto" e "da chi è stato detto". La prima informazione viene utilizzata dal sistema per distinguere l'azione richiesta ed inviare il comando all'attuatore; la seconda richiesta invece permette di individuare il soggetto che ha fatto la richiesta ed attuarla in base alle preferenze associate al soggetto. Lo stesso sistema è stato utilizzato anche nell'ambito del riconoscimento continous speech: in questo caso il sistema di speaker identification, una volta individuato il parlatore, permette di attivare un selettore che associa il modello acustico specifico allo speaker in modo da realizzare un riconoscimento vocale ad hoc sul parlatore.

Il sistema proposto è stato sviluppato in [139, 140] e di seguito se ne riportano le caratteristiche ed i risultati.

4.5.1 Architettura del sistema

A questo scopo, il sistema DSR descritto nel capitolo 3, caratterizzato da un FE basato su standard ETSI ES 201-212 e BE basato su framework CMU Sphinx, è stato modificato come illustrato in Fig. 4.13.

Il sistema svolge parallelamente due tasks: quello di speech recognition e quello di speaker identification. Entrambi prevedono due fasi: la fase di training per la realizzazione del modello acustico del parlatore e quella di testing. Per quanto riguarda il task di speech recognition, il training consiste nell'addestramento del modello acustico del parlatore, sulla base delle features MFCCs estratte dal FE del sistema e la fase di testing riguarda l'identificazione corretta del comando pronunciato come appartenente alla grammatica dei comandi, tramite l'ausilio del garbage model. Per quanto riguarda il task di speaker identification, la fase di training viene effettuata con l'ausilio delle features DKLT, e la fase di testing consiste nella classificazione dell'utente che pronuncia il comando sulla base del set di modelli precedentemente addestrati (closed-set identification). L'uscita

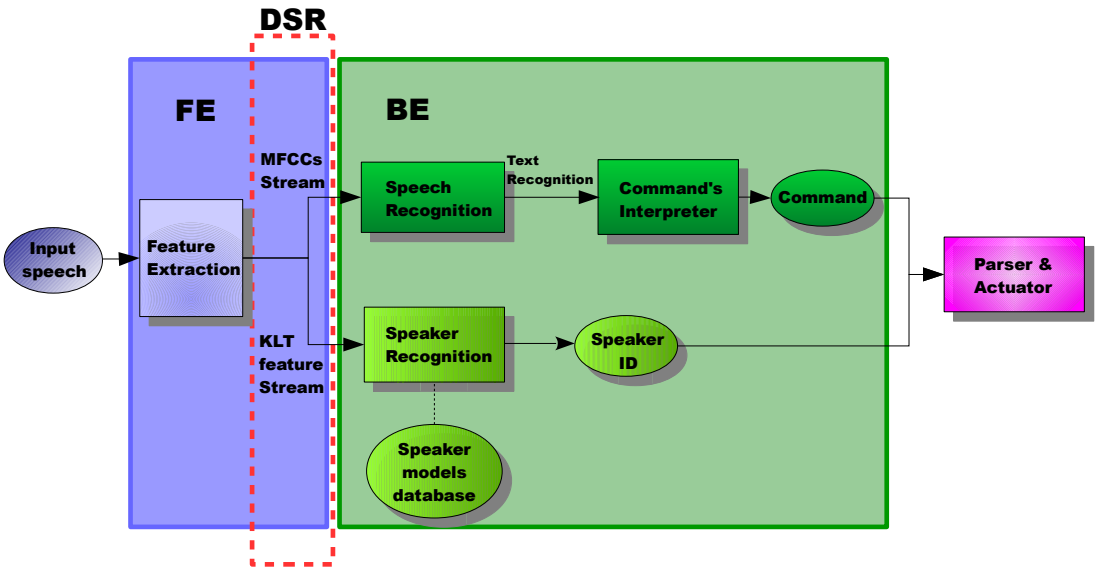


Figura 4.13: Overall system workflow.

del riconoscitore vocale, ovvero il comando, e l'uscita dell'identificatore, ovvero un identificativo associato ad ogni utente, vengono combinati e dati in pasto ad un programma di parsing che provvede, sulla base di queste due informazioni, ad associare al comando le preferenze dell'utente che lo ha pronunciato. Un dispositivo lato utente, tipicamente il FE stesso, effettua l'azione corrispondete al comando personalizzato.

Conseguentemente, anche lo schema della comunicazione illustrato nella sezione 3.4.4 del precedente capitolo, è stato modificato come illustrato in Fig. 4.14.

4.5.2 Risultati sperimentali

Gli esperimenti sono stati condotti usando i dati acquisiti da sei diversi parlatori italiani, quattro uomini (A, B, C, D) e due donne (E, F), ed un sistema composto da una semplice ed economica single board computer (Raspberry Pi®) dotata di un microfono USB standard. Per testare il sistema, il microfono è stato posizionato in un ufficio per registrare una sessione di 30 minuti di parlato (8 kHz, 16-bit), nella quale gli speakers svolgono le loro funzioni abituali intervallando al parlato continuo i comandi del sistema di controllo luci. Per valutare le performance del sistema, sono stati usati due diversi data sets per il training:

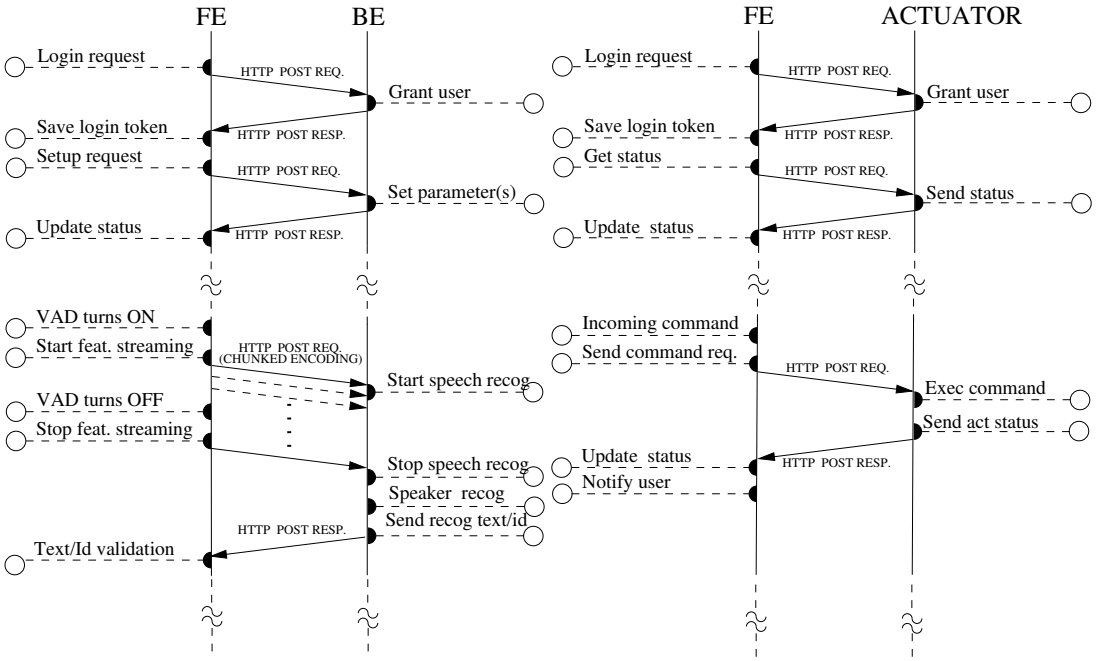


Figura 4.14: DSR-dialogue.

il primo contiene solo la registrazione di comandi appartenenti alla grammatica dei comandi stabilita (180 comandi per una durata complessiva di 10 minuti), mentre il secondo contiene continuous speech di contenuto generico (circa 20 minuti). Il testing set è composto dai singoli comandi pronunciati dall'utente (60 comandi).

La Fig. 4.15 e le Tabs. 4.17, 4.18 mostrano le performance del sistema. I parametri in tabella sono calcolati secondo le seguenti formule:

$$\begin{aligned} \text{Sensitivity} &= \frac{\text{TP}}{\text{TP+FN}} , \\ \text{Precision} &= \frac{\text{TP}}{\text{TP+FP}} , \end{aligned} \tag{4.27}$$

dove, per il processo di *speech recognition*, TP sono i veri positivi(elementi che appartengono alla grammatica dei comandi e sono riconosciuti come tali), FN sono i falsi negativi (elementi appartenenti alla grammatica ma scartati), FP sono i falsi positivi (elementi appartenenti al garbage e riconosciuti come appartenenti alla

grammatica), e TN i veri negativi (elementi che appartengono al garbage e che sono riconosciuti come tali). Per il processo di *speaker identification*, invece, TP sono i veri positivi (gli elementi della diagonale della matrice di confusione), FN sono i falsi negativi (la somma degli altri elementi sulla stessa riga della matrice di confusione), FP sono i falsi positivi (la somma degli altri elementi sulla stessa colonna della matrice di confusione), e TN sono i veri negativi (la somma degli elementi sulle altre righe e colonne della matrice di confusione). Le Figs. 4.16–

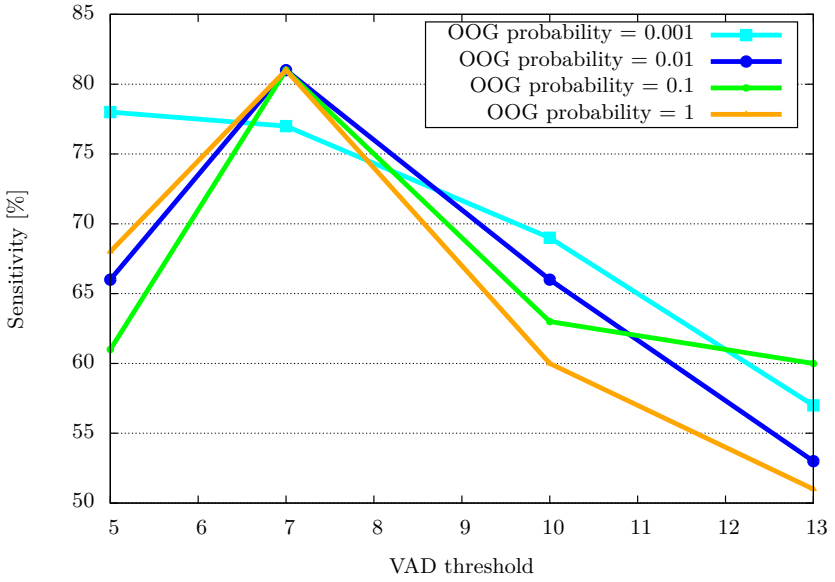


Figura 4.15: Speech recognition performance in funzione del valore di OOG probability e VAD threshold.

4.17 mostrano le performance del sistema di speaker identification, tramite il parametro sensitivity, in funzione della lunghezza della sequenza (numero di frames). Si osserva, come un incremento del numero di frames corrisponda ad un più alto tasso di identificazione.

Conclusioni Il sistema combinato di speech recognition e speaker identification, sebbene presenti ancora delle limitazioni nel rate di riconoscimento, dovuto ad aggiustamenti nel protocollo del sistema DSR, presenta buoni risultati ma soprattutto realizza un sistema versatile a basso-costo per la completa personalizzazione dell'ambiente domestico che può essere applicato in un largo numero di applicazioni.

Tabella 4.17: Speaker identification performance usando dati di training e testing appartenenti alla grammatica dei comandi.

Speaker	Gender	Sensitivity (%)	Precision (%)
		Sequence length = 200 frames	
A	M	100.00	100.00
B	M	100.00	100.00
C	M	100.00	100.00
D	M	100.00	100.00
E	F	100.00	100.00
F	F	100.00	100.00
		Sequence length = 100 frames	
A	M	95.45	95.45
B	M	100.00	100.00
C	M	100.00	100.00
D	M	95.45	95.45
E	F	92.59	100.00
F	F	100.00	97.06
		Sequence length = 70 frames	
A	M	93.55	90.62
B	M	100.00	100.00
C	M	100.00	100.00
D	M	96.77	96.77
E	F	89.47	97.14
F	F	97.87	93.88
		Sequence length = 30 frames	
A	M	85.14	81.82
B	M	98.75	92.94
C	M	97.56	96.39
D	M	86.30	86.30
E	F	76.67	90.79
F	F	90.99	87.07
		Sequence length = 1 frame	
A	M	48.10	45.99
B	M	71.83	65.86
C	M	77.89	64.95
D	M	54.92	51.14
E	F	47.54	60.14
F	F	58.35	66.47

Tabella 4.18: Speaker identification performance usando dati di training appartenenti al parlato continuo e dati di testing appartenenti alla grammatica dei comandi.

Speaker	Gender	Sensitivity (%)	Precision (%)
		Sequence length = 200 frames	
A	M	100.00	100.00
B	M	100.00	100.00
E	F	100.00	100.00
		Sequence length = 100 frames	
A	M	100.00	92.86
B	M	100.00	91.67
E	F	82.73	100.00
		Sequence length = 70 frames	
A	M	95.83	95.83
B	M	100.00	84.78
E	F	75.00	100.00
		Sequence length = 30 frames	
A	M	95.16	80.82
B	M	93.94	83.78
E	F	67.14	96.97
		Sequence length = 1 frame	
A	M	69.27	56.26
B	M	73.84	71.86
E	F	54.07	63.31

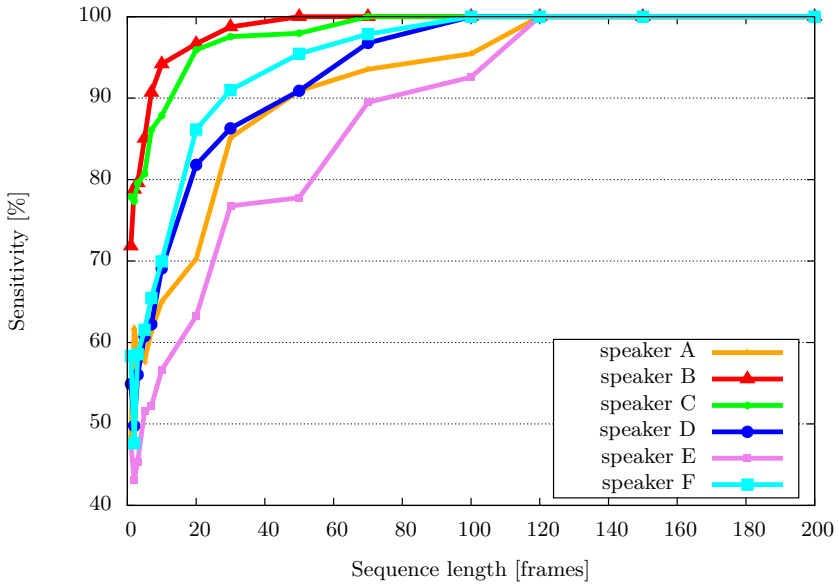


Figura 4.16: Speaker identification performance in funzione della lunghezza della sequenza usando sia training che testing data appartenenti alla grammatica dei comandi.

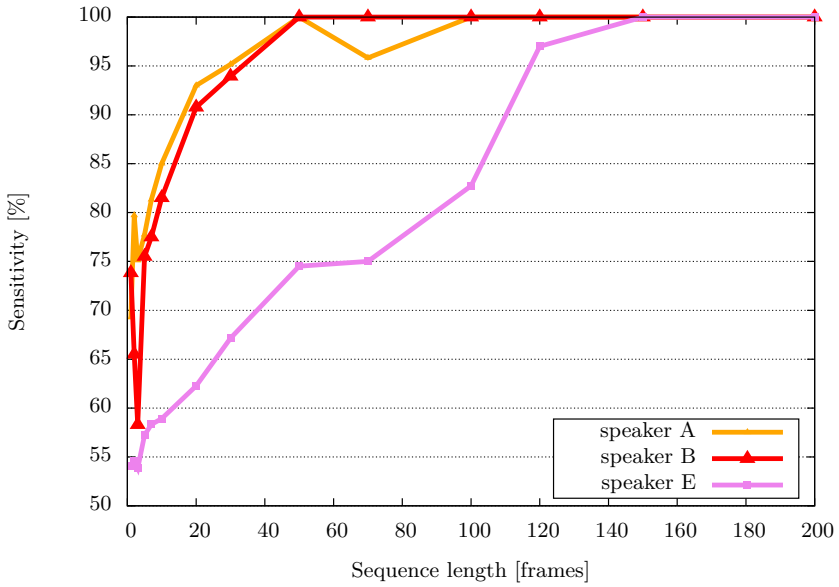


Figura 4.17: Speaker identification performance in funzione della lunghezza della sequenza usando training data dal continuous speech e testing data appartenenti alla grammatica dei comandi.

Capitolo 5

Speech Synthesis

Un sistema di interazione vocale richiede che il sistema sia in grado, in particolari situazioni, di rispondere a delle richieste. A tal fine il sistema deve essere dotato di un sintetizzatore vocale.

La sintesi vocale (*speech synthesis*) è la tecnica per la riproduzione artificiale della voce umana. Un sistema usato per questo scopo è detto sintetizzatore vocale e può essere realizzato tramite software o via hardware. I sistemi di sintesi vocale sono noti anche come sistemi *text-to-speech* (TTS) per la loro possibilità di convertire il testo in parlato [141]. La qualità di un sintetizzatore vocale si valuta sulla base sia della somiglianza con la voce umana (*naturalezza*) che con il suo livello di comprensibilità (*intelligibilità*). Un programma di conversione da testo a voce con una buona resa può avere quindi un ruolo importante anche nell'accessibilità. Per questo, fin dai primi anni ottanta molti sistemi operativi includono funzioni di sintesi vocale.

Sebbene siano state studiate diverse tecniche per la sintesi di segnali vocali, che verranno illustrate in questo capitolo, l'approccio basato sui modelli HMM si è dimostrato tra i più vantaggiosi in termini di performance ottenute [13, 14, 15]. Ciò nonostante un problema ancora aperto è rappresentato dalla ricostruibilità del segnale mediante features in grado di assicurare la perfetta ricostruzione del segnale vocale e superare quindi i limiti imposti dal modello HMM.

Il metodo proposto combina un *learning HMM* delle sequenze di stati associate ai fonemi (context-dependent) ad una trasformazione *Modified Discrete Cosine Transform* (MDCT) che assicura la perfetta ricostruzione del segnale. L'utilizzo della MDCT al posto della rappresentazione classica basata su MFCCs ha quindi il grande vantaggio di garantire la perfetta e diretta ricostruzione del segnale oltre che permettere un 50% di overlap tra i blocchi senza incrementare il data

rate, cioè senza aggiungere i costi derivanti dalla duplicazione del trattamento dei dati. Il metodo di sintesi proposto è un metodo di sintesi "ibrido" [142] che unisce l'approccio HMM e quello *unit-selection* per la generazione di forme d'onda. Questo metodo "ibrido" combina alcuni dei vantaggi di entrambe le tecniche, come la flessibilità e la buona qualità dello speech prodotto in termini di naturalezza ed intelligibilità.

In questo capitolo, verrà illustrato il processo di sintesi vocale, analizzando le tecniche utilizzate per realizzarlo ed i metodi di valutazione opportuni, fino a proporre e descrivere nel dettaglio il metodo di sintesi oggetto di questa tesi, la cui qualità verrà validata da risultati sperimentali.

5.1 Cenni storici

Dispositivi meccanici L'evoluzione storica dei sintetizzatori vocali è molto articolata, sebbene in molti credano che questi strumenti si siano sviluppati solo con l'era informatica, la loro nascita è ben più lontana dai giorni nostri: le prime apparecchiature furono costruite da Gerbert di Aurillac, Albertus Magnus e Roger Bacon, tra il X e il XIII secolo.

Nel 1779, lo scienziato danese Christian Kratzenstein, che si trovava a lavorare presso l'Accademia russa delle scienze, costruì modelli dell'apparato vocale umano che potevano riprodurre i cinque suoni lunghi delle vocali (ossia i suoni [*a*:], [*e*:], [*i*:], [*o*:] e [*u*:] secondo l'Alfabeto fonetico internazionale)[143]. A questi dispositivi seguì la Macchina acustica-meccanica vocale, un meccanismo a mantice realizzato dal viennese Wolfgang von Kempelen e descritto in un suo lavoro del 1791 [144] (Fig. 5.1). Questa macchina aggiungeva un modello delle labbra e della lingua consentendo così di sintetizzare oltre alle vocali anche le consonanti.

Nel 1837 Charles Wheatstone produsse una "macchina parlante" basata sul progetto di von Kempelen, e nel 1846 Joseph Faber costruì l'Euphonia (Fig. 5.2), in grado di riprodurre tra l'altro l'inno nazionale inglese. Il progetto di Wheatstone fu poi ripreso a sua volta nel 1923 da Paget [145].

Negli anni trenta, i Bell Labs svilupparono il Vocoder (Fig. 5.3), un analizzatore e sintetizzatore elettronico della voce comandato a tastiera con un risultato chiaramente intelligibile. Homer Dudley perfezionò ulteriormente questo apparecchio creando il VODER, di cui venne data una dimostrazione nel 1939 durante la Fiera Mondiale di New York. Questa invenzione risultava molto più compren-

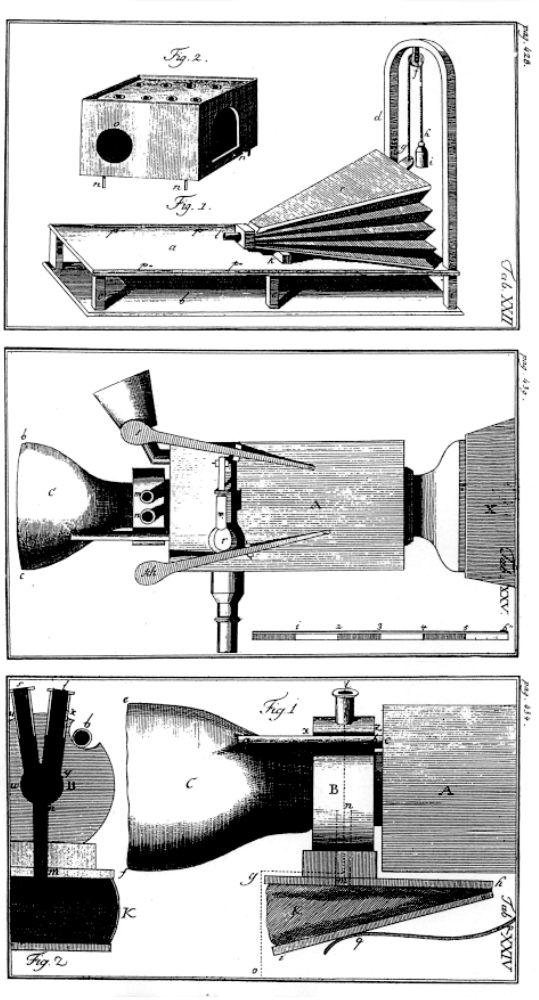


Figura 5.1: Macchina acustica-meccanica vocale di Von Kempelen (disegni del suo testo del 1791).

sibile rispetto ai prodotti precedenti, la sua struttura era molto simile a quella di un pianoforte: attraverso un pedale veniva controllata la potenza di un segnale che era modellato da una serie di filtri pilotati per mezzo di una tastiera.

Il ricercatore Franklin S. Cooper e i suoi colleghi dei Laboratori Haskins realizzarono alla fine degli anni quaranta il riproduttore di sequenze, completato nel 1950. Di questo dispositivo furono realizzate diverse versioni di cui soltanto una è arrivata fino ai nostri giorni. Il dispositivo converte in suono le immagini dello spettro acustico della voce e fu proprio grazie a questo meccanismo che Alvin Liberman e i suoi colleghi scoprirono le caratteristiche acustiche alla base della

Figura 5.2: Euphonia, macchina di sintesi vocale meccanica realizzata da Joseph Faber (illustrazione del 1846).

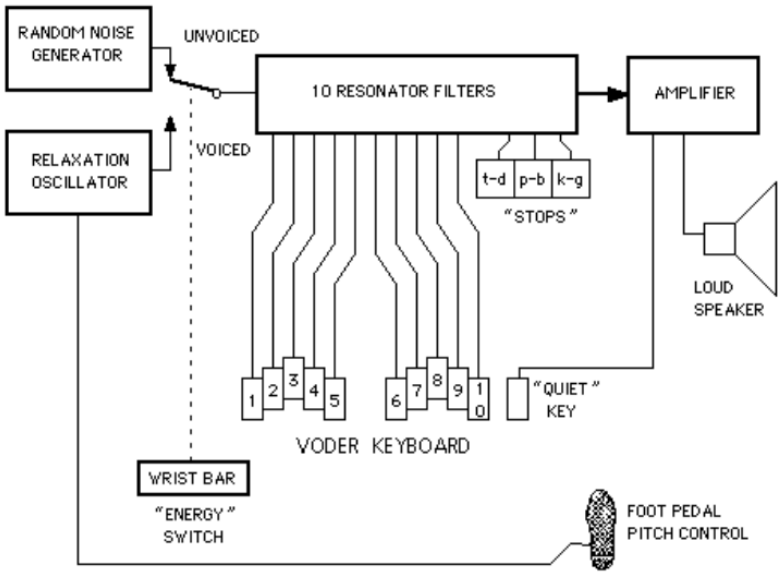


Figura 5.3: Lo schema di funzionamento di Vocoder.

percezione dei segmenti fonetici (consonanti e vocali).

Dispositivi elettronici I primi sintetizzatori vocali elettronici ricreavano una voce molto metallica ed erano spesso incomprensibili; da allora però la qualità è aumentata costantemente e la voce prodotta dai moderni sistemi di sintesi vocale è talvolta indistinguibile dalla vera voce umana.

I primi sistemi di sintesi vocale basati su computer furono creati sul finire degli anni cinquanta e il primo sistema di sintesi vocale text-to-speech (da testo a voce) completo venne realizzato nel 1968. Nel 1961 i fisici John Larry Kelly, Jr e Louis Gertsman utilizzarono un computer IBM 704 per sintetizzare la voce. Questo esperimento rappresentò uno dei momenti salienti dell'attività dei Bell Labs: il vocoder di Kelly riprodusse la canzone Daisy Bell, con l'accompagnamento musicale di Max Mathews. Lo scrittore Arthur C. Clarke si trovava casualmente ai Bell Labs in visita all'amico e collega John Pierce proprio nel momento di questa dimostrazione e ne rimase impressionato al punto da riprendere la scena in uno dei momenti cruciali del suo romanzo 2001: Odissea nello spazio, facendo eseguire la stessa canzone al computer HAL 9000 mentre viene disattivato dall'astronauta Dave Bowman, scena che fu poi riprodotta fedelmente dal regista Stanley Kubrick nell'omonimo film.

Il primo apparato di sintesi vocale in italiano, MUSA, è nato nel 1975 presso i laboratori CSELT (Gruppo STET); il prototipo era in grado di leggere un testo, con una caratteristica voce "metallica" e, nel 1978, anche di cantare il brano Fra Martino Campanaro. Nel 1978 il gruppo di ricerca CSELT sulle tecnologie vocali (nel 2001 divenuto lo spin-off Loquendo) era l'unica realtà industriale al mondo, oltre AT&T, a disporre di una tecnologia di sintesi vocale di interesse industriale.

Nonostante i successi ottenuti con i sintetizzatori elettronici, la ricerca sui sintetizzatori vocali di tipo meccanico non è stata abbandonata, specialmente in vista di un possibile impiego di tali sistemi per robot di tipo umanoide [146, 147].

5.2 Processo di sintesi vocale

Un sistema o motore di sintesi vocale si compone di due parti: *front-end* e *back-end*. Semplificando il processo, il *front-end* si preoccupa della conversione del testo in simboli fonetici mentre il back-end interpreta i simboli fonetici trascritti dal front-end per poterli trasformare in una voce artificiale. Lo schema ad alto livello di un sistema di sintesi vocale è mostrato in Fig. 5.4.

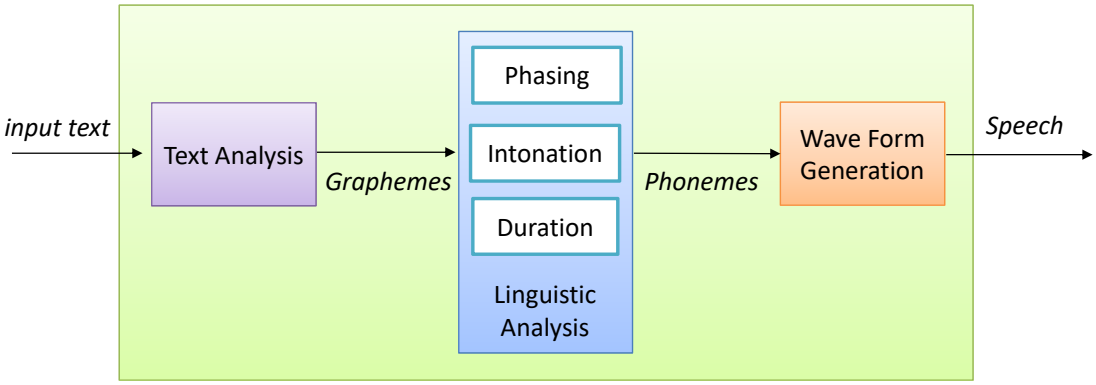


Figura 5.4: Schema a blocchi del processo di sintesi vocale.

Il front-end prevede due funzioni chiave: per prima cosa, viene eseguita un'analisi del testo scritto per convertire tutti i numeri, le sigle e le abbreviazioni in parole per esteso (es. il testo '2' viene convertito in 'due'). Questo fase di pre-elaborazione viene definita come normalizzazione o classificazione del testo. La seconda funzione consiste nel convertire ogni parola nei suoi corrispondenti simboli fonetici e nell'eseguire l'analisi linguistica del testo rielaborato, suddividendolo in unità prosodiche, ossia in proposizioni, frasi e periodi. Il processo di assegnazione della trascrizione fonetica alle parole è chiamato conversione da testo a fonema o da grafema[1] a fonema (text-to-phoneme, TTP)[148].

La trascrizione fonetica e le informazioni di prosodia[2] combinate insieme costituiscono la rappresentazione linguistica simbolica che viene utilizzata dal back-end per la conversione in suoni di tali informazioni ossia per il processo di sintesi vero e proprio.

Di seguito, si descrivono in dettaglio i moduli che compongono il sistema di sintesi vocale, detto anche sistema Text-To-Speech (TTS).

[1]Nella terminologia linguistica, la minima unità grafica di un sistema alfabetico o sillabico o ideografico ecc., cioè un segno che in un determinato sistema grafico si distingue da tutti gli altri segni del sistema e pertanto è in grado di far distinguere sul piano grafico una parola da altre.

[2]Definita come l'insieme delle regole ritmiche della produzione vocale. Presso i grammatici greci, ogni particolarità accessoria che appare nella realizzazione di un suono nella parola, indipendentemente dall'articolazione essenziale di esso: intonazione, aspirazione, quantità ecc. In senso ristretto e più comunemente, ogni particolarità che concerne la quantità o durata delle sillabe in sé e all'interno di una parola, soprattutto in rapporto alla versificazione (talvolta il termine è usato come sinonimo di metrica).

5.2.1 Sistemi Text-To-Speech (TTS)

Un sistema TTS è definito come: "The automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter." [149].

Spesso si tende a confondere i sistemi di sintesi vocale con i cosiddetti Voice Response System, che non sono propriamente sistemi TTS e si trovano ad esempio nelle stazioni ferroviarie o ancora nei navigatori satellitari. Questi si distinguono dai sistemi di sintesi veri e propri per:

- database molto limitati: vista la ristretta gamma di frasi da riprodurre;
- fedeltà di riproduzione molto elevata: vista, ancora, la ristrettezza del database delle frasi, che risultano prive dei problemi di fluidità del linguaggio;
- settori di utilizzo limitati: dato che si utilizzano principalmente nel settore dei trasporti (ferrovie, autostrade o navigatori).

Principalmente i sistemi TTS vengono adoperati nel settore delle telecomunicazioni, si pensi ad alcune applicazioni utilizzate nei telefoni cellulari o altri impieghi simili. Un ulteriore ed importante campo di applicazione è quello dell'accessibilità. Il famosissimo astrofisico Stephen Hawking, incapace di parlare perché affetto da gravi disabilità, da anni utilizza questi sistemi per tenere le sue conferenze e le sue lezioni; questo per fa comprendere come questi sistemi forniscano meccanismi molto validi che permettono di raggiungere un elevato grado di accessibilità.

Un sistema di sintesi vocale è composto principalmente da due macromoduli: il Natural Language Processing (NLP) ed il Digital Signal Processing (DSP). Il primo riceve in ingresso una serie di input testuali e, dopo averle elaborati sulla base delle regole grammaticali e prosodiche della lingua, produce in uscita la trascrizione fonetica assieme alle informazioni di intonazione e prosodia necessari per il macro-modulo successivo. Il DSP, combinando gli outputs (fonemi e prosodia) generati dal modulo NLP, è in grado di generare il segnale vocale.

Più in dettaglio, lo schema in Fig. 5.5 si può espandere come illustrato in Fig. 5.6.

Text Analyzer (Modulo NLP) Converte il testo dato in input in una sequenza di elementi machine-readable. Si compone dei seguenti moduli:

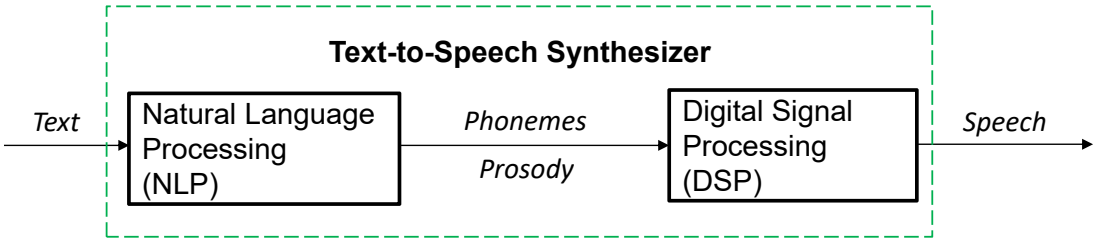


Figura 5.5: Schema di funzionamento di un sistema Text-To-Speech.

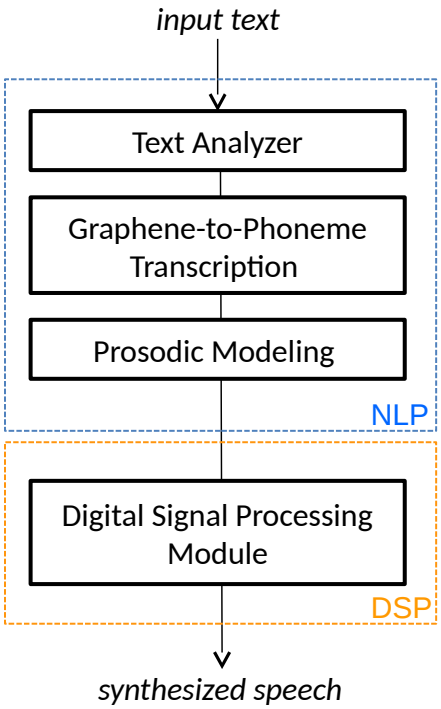


Figura 5.6: Schema di funzionamento di un sistema Text-To-Speech nel dettaglio.

- **pre-processing module**: ha il compito di studiare in prima battuta tutte le abbreviazioni, gli acronimi ed i numeri che dovranno essere poi convertiti in parole;

- **morphological analysis module**: il suo ruolo è classificare le parole in base alle possibili categorie grammaticali di appartenenza, studiando i morfemi[3] di ogni termine;

[3]Il morfema è l'unità elementare di ogni forma grammaticale.

- **contextual analysis module**: ha la funzione di scremare tutte le categorie create dall'analizzatore morfologico e stabilire, con un'alta probabilità, il contesto in cui è inserita ogni parola. Il compito è svolto cercando di interpretare tutte le parole che circondano il termine. Per raggiungere questi obiettivi vengono utilizzati dei metodi legati allo studio statistico della composizione delle frasi come le Catene di Markov o le Reti Neurali , vengono inoltre usati degli alberi di ricerca costruiti mediante delle regole linguistico-grammaticali.

- **syntactic parser**: può essere ridefinito come un analizzatore preprosodico, infatti cerca di analizzare le parti restanti del discorso, organizzandole in proposizioni e facilitando così i compiti successivi.

Grapheme-Phoneme Conversion (Modulo NLP) Converte il risultato della fase di text processing in sequenze di fonemi, tramite due metodi possibili:

- **dictionary-based**: sistema basato sull'utilizzo di dizionari. La strategia basata sui dizionari è chiamata così in quanto i fonemi vengono registrati in database detti appunto dizionari, in aggiunta vengono inserite delle regole che descrivono come vengono modificate le trascrizioni fonetiche mediante le regole derivazionali, flessive e composte.

- **rule-based**: strategia basata sull'utilizzo delle regole. Questa tecnica basata sulle regole è molto più raffinata ed inoltre ha l'appoggio dei fonologisti. Si basa su una serie di regole fonologiche utilizzate per modellare tutte quelle informazioni che arrivano in ingresso (sillabe, caratteristiche morfosintattiche ed i grafemi), tutte quelle parole che costituiscono in qualche modo un'eccezione sono registrate in un database molto limitato.

Prosodic Modelling (Modulo NLP) Combina la stringa di fonemi ottenuta nella fase precedente, con le informazioni che riguardano la durata delle singole unità linguistiche e l'intonazione delle frasi.

- **Durata**: *Metodo di Klatt*
  Dennis Klatt [150, 151] definì un ampio set di regole deterministiche per la durata dei singoli fonemi. Sebbene queste regole siano state definite nell'ambito del suo progetto di ricerca MITalk [141] (poi commercializzato

come DECtalk), queste regole possono essere applicate a qualsiasi sistema di sintesi. Si rimanda alla sezione 5.5.4.2 per i dettagli sull'argomento.

- **Intensità**: *Modello Fujisaki*
  Il modello Fujisaki è un superpositional model per la rappresentazione dell'andamento del pitch del segnale vocale [152].

Digital Signal Processing Module (Modulo DSP) Questo secondo macromodulo è il responsabile della produzione del segnale, corrispondente ai parametri ricevuti in ingresso dalla prima parte di analisi testuale. Viene tenuto conto delle caratteristiche articolatorie del tratto vocale per fare in modo che l'output vocale finale corrisponda all'input testuale iniziale. Per realizzare questi compiti il calcolatore deve cercare di rendere il segnale più intelligibile possibile, cercando così di ridurre tutti quei problemi che si riconducono alla coarticolazione delineando un suono molto meno robotico.

5.3 Tecniche di sintesi vocale

Le tecniche di sintesi vocale possono essere suddivise in quattro gruppi principali:

- **Sintesi articolatoria**;
- **Sintesi basata su regole**;
- **Sintesi concatenativa**;
- **Sintesi parametrica** o Markoviana.

5.3.1 Sintesi articolatoria

La sintesi articolatoria ricorre a tecniche computazionali basate su modelli biomeccanici dei tratti vocali umani e dei loro processi di articolazione; è quindi in teoria il metodo migliore per generare parlato di elevata qualità in maniera artificiale. In realtà, è uno dei metodi più complessi e più onerosi a livello computazionale, in quanto modellare il sistema con un insieme limitato di parametri richiede una notevole capacità di elaborazione. Inoltre presenta un'altra importante difficoltà: ottenere modelli tridimensionali del tratto vocale; Per questi

motivi, la sintesi articolatoria ha ricevuto negli anni meno attenzione rispetto alle altre tecniche, non raggiungendo quindi lo stesso livello di successo.

5.3.2 Sintesi basata su regole

La sintesi basata sulle regole ricrea la voce per elaborazione basandosi su un modello acustico e per tale motivo viene detta anche *sintesi per formanti*[4]. Vengono simulate le caratteristiche spettrografiche dei diversi suoni tramite composizioni di vari risonatori, ciascuno accordato su frequenze specifiche di formanti diversi. Le posizioni delle formanti nei diversi fonemi e le loro variazioni nel passaggio da un fonema al successivo sono descritte da specifiche regole. Tale tecnica rappresenta, l'apparato vocale umano attraverso una funzione di trasferimento e la produzione del suono viene quindi realizzata configurando dei parametri come ampiezza e frequenza nella legge matematica. Ciò è realizzato attraverso una sorgente o un generatore di rumore, che eccitano una serie di risonatori finché non viene rappresentato lo spettro desiderato. Controllando queste sorgenti è possibile anche modellare le fasi del discorso parlate e non. La sorgente simula le corde vocali mentre il generatore di rumore raffigura dei possibili restringimenti nel tratto vocale; il filtro, o meglio il set di risonatori, riproduce mediante una funzione di trasferimento alcune parti dell'apparato fonatorio umano come cavità orale, faringe e cavità nasale, le ultime correzioni del segnale tengono conto delle labbra. Molti sistemi di sintesi rule-based generano una voce dal suono artificiale e molto metallico che non può essere scambiata per una voce umana. Questa tecnica di sintesi non ha però come obiettivo la massima naturalezza e presenta una serie di vantaggi: la sintesi basata sulle regole infatti è decisamente intelligibile anche ad alte velocità, non presentando i piccoli stacchi acustici tipici di altri sistemi (es. sintesi concatenativa). Per questo motivo, la sintesi ad alta velocità è molto usata per i sistemi di lettura dello schermo per l'uso dei computer da parte delle persone ipovedenti o persone affette da dislessia. Inoltre i sistemi di sintesi basata sulle regole sono gestiti da programmi di dimensione più contenuta non dovendo utilizzare un database di campioni vocali. Questa caratteristica ne consente l'impiego in sistemi embedded, dove la capacità di memoria e la potenza di calcolo del microprocessore possono essere limitate. Infine, i sistemi di sintesi basata sulle regole possono controllare tutti gli aspetti del linguaggio vocale, ge-

[4]Frequenze caratteristiche del parlato grazie alle quali i vari suoni vengono identificati dall'ascoltatore.

nerando un'ampia varietà di prosodie e intonazioni e veicolando così non soltanto il contenuto del testo ma anche effetti emotivi e toni di voce. D'altra parte il metodo presenta una qualità del suono troppo scadente ed inoltre l'imprescindibile dipendenza dai parametri fa sì che spesso il modello sia impreciso e si ricorra all'ottimizzazione manuale per modificare gli errori ed i disturbi.

5.3.3 Sintesi concatenativa

La sintesi concatenativa è la più largamente usata tra tutti i sistemi TTS. La filosofia di questo approccio è di riprodurre il suono concatenando e combinando una serie di frammenti di voce registrati in precedenza e memorizzati in una base di dati, cercando di scegliere quelli che possono ridurre al minimo il tasso di coarticolazione. La creazione del database è quindi una fase importante del processo di sintesi: vengono scelte tutte quelle unità (fonemi, difoni[5] , sillabe, parole, ..) che coprono la maggior parte delle parole di una lingua. La scelta di queste unità viene fatta anche cercando di ridurre al minimo il coefficiente di coarticolazione che renderebbe meno comprensibile il suono. Una volta scelte, le unità vengono memorizzate in base al nome, alla durata ed alla forma d'onda. Dopo la formazione dei segmenti, il compito del sintetizzatore è di concatenarli in modo da essere pronti per le fasi successive di adattamento prosodico e riduzione della coarticolazione.

Esistono tre sotto-tipi principali di sintesi concatenativa.

- Domain-specific synthesis: concatenazione di parole e frasi pre-registrate per generare emissioni complete.
- Diphone synthesis: sintesi concatenativa di difoni.
- Unit-Selection synthesis: sintesi concatenativa a selezione di unità variabili.

Uno degli aspetti più importanti di questa tecnica, è trovare la corretta lunghezza dell'unità. Se si sceglie una lunghezza maggiore, si avrà una maggior naturalezza e controllo della coarticolazione ma sarà richiesta una memoria maggiore. Se si sceglie una lunghezza piccola, si risparmierà in termini di memoria a scapito però della complessità nella classificazione delle unità e della naturalezza.

[5]Il difone è il segmento acustico che include la transizione tra due fonemi consecutivi, quindi il segnale che connette la seconda metà di un fonema con la prima metà del fonema successivo.

In generale questa metodologia produce sicuramente il risultato di sintesi più naturale ed intelligibile. Tuttavia la differenza tra le variazioni naturali della voce umana e le tecniche di frammentazione automatica delle forme d'onda può talvolta generare dei piccoli disturbi udibili. Inoltre, questo tipo di sintesi è generalmente limitato ad un parlatore e quindi ad una voce e richiede una capacità di memoria maggiore rispetto agli altri metodi.

5.3.4 Sintesi parametrica - HMM

Grazie alla capacità di rappresentare non solo le sequenze di fonemi e di produrre una voce abbastanza naturale, con l'utilizzo di un database relativamente limitato, la sintesi parametrica è attualmente un hot topic nella ricerca di sistemi vocali, sebbene siano ancora diversi i problemi da risolvere [153]. Questa tecnica si basa essenzialmente su HMM; lo spettro di frequenze (usato per il tratto vocale), la frequenza fondamentale (usata per la sorgente vocale) e la durata dell'emissione vocale (usata per la prosodia) sono modellate simultaneamente tramite modelli HMM basati su un criterio di massima verosimiglianza. Gli HMMs sono ampiamente utilizzati con successo nell'ambito del riconoscimento vocale [41]. La sintesi vocale HMM-based può essere considerata come il problema inverso del riconoscimento vocale HMM-based [154]. Nel caso della sintesi, la sequenza di parole è data e si cerca la sequenza ottima del vettore di features. Nel momento in cui la sequenza di parole è data, non si ha più bisogno del modello linguistico. In prima istanza, la sequenza di parole deve essere convertita in una sequenza di fonemi usando regole lessicali o una conversione grafema-fonema (così come viene fatto anche in altre tecniche di sintesi come la unit-selection). La sequenza di fonemi è poi applicata al modello acustico al fine di determinare la sequenza di features più probabile [155]. Rispetto però al riconoscimento vocale, nella sintesi, non è sufficiente considerare le features, perché l'obiettivo è quello di generare una forma d'onda udibile. Questo si ottiene generando un'eccitazione (voiced o unvoiced) e filtrandola per mezzo delle caratteristiche spettrali frame-by-frame. Per il modeling dell'eccitazione, esistono diversi approcci: harmonic + noise model [156], residual prediction [157], o mixed excitation with state-dependent filters [158].

La Fig. 5.7 mostra lo schema a blocchi di un sintetizzatore vocale HMM-based [155]. Il sistema consiste di due fasi: training stage e synthesis stage. Nella fase di training, i parametri spettrali (MFCCs) ed i parametri di eccitazione (fun-

damental frequency) vengono estratti dal database audio. I parametri estratti sono modellati tramite context-dependent HMMs. Nella fase di sintesi, il testo in ingresso viene analizzato per produrre una sequenza di fonemi. In base a questa sequenza di fonemi, la frase HMM, che rappresenta l'intero testo da sintetizzare, è costruita concatenando fonemi HMMs. Dalla frase HMM, usando l'algoritmo di generazione dei parametri, viene generata una sequenza parametrica. Infine, applicando il filtro di sintesi Mel Log Spectral Approximation (MLSA), lo speech è sintetizzato dai parametri estratti.

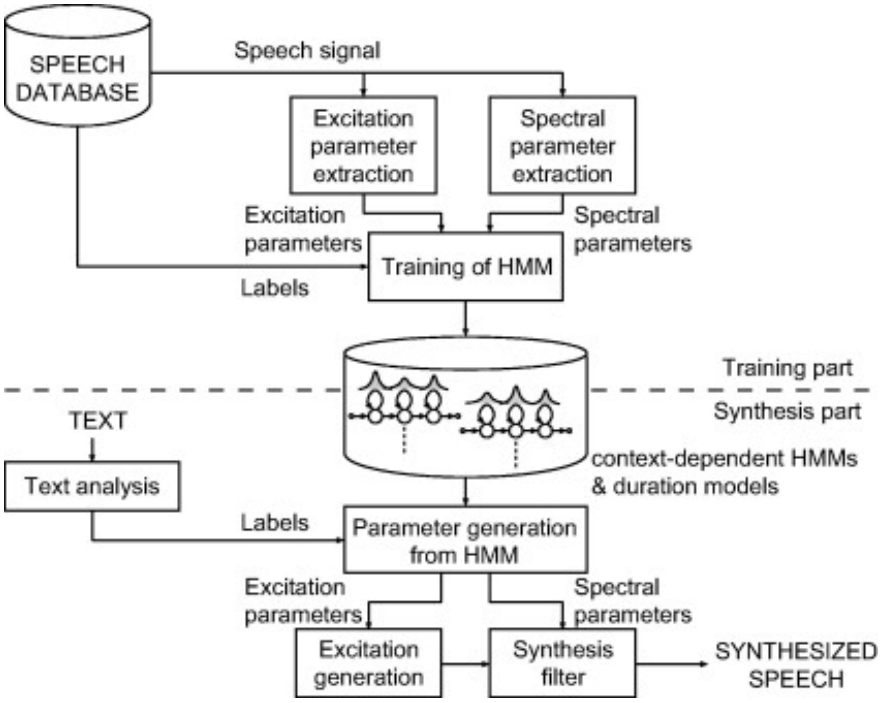


Figura 5.7: Schema a blocchi di un sintetizzatore HMM-based.

Sicuramente il vantaggio più grande di un sistema HMM è il basso impatto sulla memoria (pochi megabytes), dato che richiede di memorizzare solo i parametri del modello e non gli speech data. Inoltre, un sistema HMM-based richiede una minor computazione rispetto alla sintesi unit-selection. Questo perché quest'ultima ricerca in uno spazio che è tanto più grande quanto sono i dati vocali disponibili. Oltre a questi vantaggi, questa tecnica ha le potenzialità di produrre un segnale vocale ad alta qualità, non presentando la problematica delle discontinuità ai contorni come accade nella sintesi unit-selection.

La Blizzard Challenge[6] 2005 ha dimostrato che la sintesi HMM è in grado di superare la sintesi unit-selection, grazie al contributo dato dai buoni risultati raggiunti in termini di naturalezza ed intelligibilità [160], con un database (Arctic database[7]) relativamente limitato di 1.5 h di speech; questo anche perché, nel caso di un database ridotto, la sintesi unit-selection può avere problematiche nella ricerca di tutte le unità necessarie per effettuare la sintesi. Infatti nella Blizzard Challenge 2006, che forniva un corpus di 5 h, è stato un sistema basato su sintesi unit-selection ad attestarsi come vincitore [161].

In conclusione, HMM e unit-selection, sono attualmente i metodi più utilizzati nei sistemi di sintesi vocale.

5.4 Metodi di valutazione

Le qualità più importanti di una sintesi vocale sono la *naturalezza* e l'*intelligibilità* [151]. La naturalezza esprime quanto la voce sintetizzata si avvicina a quella umana mentre l'intelligibilità rappresenta la facilità di comprensione della voce sintetizzata. In alcune applicazioni, per esempio dispositivi reader per non vedenti, si predilige l'intelligibilità alla naturalezza; mentre nella applicazioni multimediali, la naturalezza è il fattore dominante. Un sintetizzatore ideale è allo stesso tempo naturale e intelligibile; nella realtà i sistemi di sintesi vocale approssimano tale comportamento tentando di ottimizzare entrambe le caratteristiche.

La procedura di valutazione di un sistema di sintesi vocale consiste in un set di test soggettivi, detti anche percettivi, volti a valutare la naturalezza e l'intelligibilità del parlato sintetizzato (la percezione, appunto, dell'ascoltatore) tramite l'ascolto di sillabe, parole, frasi. La valutazione può essere fatta quindi a più livelli, cioè su fonemi o parole o intere frasi, a seconda del tipo di informazione richiesta. Molti metodi di test sono stati definiti negli ultimi dieci anni. Di seguito, sono riportati i metodi di valutazione più comunemente utilizzati.

[6] La Blizzard challenge è stata fondata con lo scopo specifico di confrontare tecniche di sintesi vocale corpus-based [159].

[7] Si tratta del database specificatamente costruito per la Blizzard challenge dalla Carnegie Mellon University e rilasciato liberamente a tutti i partecipanti alla competizione (materiale non protetto da copyright).

5.4.1 Segmental Evaluation Methods

In questa tipologia di test è valutata solo l'intelligibilità di un singolo segmento o fonema.

Diagnostic Rhyme Test (DRT) Il Diagnostic Rhyme Test (DRT) usa un set di parole per testare l'intelligibilità della consonante nella posizione iniziale. Il test consiste di 96 coppie di parole che differiscono per una singola caratteristica acustica nella consonante iniziale. L'utente ascolta una sola parola e sceglie, tra le due soluzioni proposte, la parola che pensa sia stata effettivamente pronunciata. Il risultato finale consiste nella percentuale di errore totale, calcolata facendo una media delle risposte errate; ulteriori indagini possono essere fatte investigando il risultato delle confusion matrices.

Modified Rhyme Test (MRT) Il Modified Rhyme Test (MRT) è una sorta di estensione del DRT, in quanto riguarda la comprensione sia della consonante iniziale che di quella finale. Il test consiste in 50 sets ognuno dei quali contiene 6 parole monosillabiche per un totale di 300 parole. L'utente ascolta una parola alla volta, giudicando la parola ascoltata in un test a risposta multipla.

5.4.2 Sentence Level Tests

In questa tipologia di test è valutata la comprensione di un'intera frase.

Semantically Unpredictable Sentences (SUS) Il SUS consiste nella valutazione di un set di frasi costruite scegliendo le parole in modo random da una lista predefinita di parole, al fine di valutare l'intelligibilità delle singole parole nel contesto della frase. La scelta random delle parole fa sì che la struttura sintattica delle frasi non sia quella canonica della lingua analizzata. In particolare, sono state definite cinque strutture sintattiche specifiche per questo tipo di test [162], ognuna delle quali non supera otto parole, al fine di non saturare la memoria a breve termine dell'ascoltatore. Il parametro di giudizio è il Word Error Rate (WER) definito come:

$$WER = 100\% \frac{S + D + I}{N} \quad , \tag{5.1}$$

dove: S numero di sostituzioni, D numero di cancellazioni, I numero di inserimenti, N parole totali della sentenza corretta.

5.4.3 Comprehension tests

Nei test precedenti, viene valutata la qualità di un singolo fonema o parola o frase. Nei test di compressione, all'utente viene chiesto di ascoltare un certo numero di frasi o paragrafi e di rispondere a domande relative alla comprensione del testo ascoltato. In questo caso, non è importante quindi il riconoscimento di un singolo fonema, se viene compreso il senso generale del testo.

5.4.4 Overall Quality Evaluation

I metodi di Overall Quality Evaluation sono per lo più sviluppati per valutare le singole caratteristiche di qualità della voce (naturalezza, intelligibilità, somiglianza all'originale).

Mean Opinion Score (MOS) Il MOS è probabilmente il test più ampiamente utilizzato ed il più semplice per valutare la qualità dell'audio sintetizzato. Il MOS consiste in una scala a cinque livelli di valutazione crescente da 1 (pessimo) a 5 (eccellente), noto anche come ACR (Absolute Category Rating). All'utente viene chiesto semplicemente di ascoltare l'audio sintetizzato e valutarne la qualità secondo la scala in Tab. 5.1.

Tabella 5.1: Mean Opinion Score.

value	**MOS**	**DMOS**
5	Excellent	Inaudible
4	Good	Audible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Degradation Mean Opinion Score (DMOS) La Tab. 5.1 mostra anche il parametro DMOS, chiamato anche DCR (Degradation Category Rating), utilizzato

per valutare la percezione della degradazione del segnale sintetizzato, rispetto eventualmente alla traccia audio originale.

5.5 Implementazione di un sistema di sintesi vocale HMM/unit-selection tramite rappresentazione MDCT

Il sistema proposto, presenta una tecnica di sintesi vocale "ibrida" HMM/unit-selection basata sulla rappresentazione MDCT.

È dimostrato che la sintesi parametrica HMM realizza un framework particolarmente flessibile e robusto che permette di generare un segnale vocale sintetizzato con differenti stili ed espressioni [163, 164]. Grazie all'abilità di rappresentare non solo la sequenza di fonemi ma anche diversi contesti linguistici, la sintesi HMM è stata recentemente uno dei maggiori temi di ricerca nel campo della sintesi vocale [165, 166, 167, 168, 169].

Nelle tecniche convenzionali, basate sul source-filter model, si assume che le informazioni fonetiche e prosodiche siano principalmente contenute nell'inviluppo spettrale, frequenza fondamentale (F0) e durata dei singoli fonemi [170]. Nonostante i molteplici vantaggi, ci sono ancora delle limitazioni in questo approccio. In particolare, la difficoltà nel modeling di F0 data dalla sua specifica natura discontinua provocata dall'alternanza delle regioni vocali voiced ed unvoiced [171]. Inoltre, l'inviluppo spettrale definisce una trasformazione non invertibile e quindi il segnale vocale non può essere ricostruito perfettamente dalla sequenza di features [172, 173]. I parametri MFCCs, correntemente adottati nelle tecniche convenzionali di sintesi come features per il modeling acustico basato su HMM, non assicurano quindi la perfetta ricostruibilità del segnale vocale.

In contrasto a questi approcci, si propone di seguito una tecnica di analisi e sintesi basata su MDCT che garantisce la perfetta ricostruzione del segnale, superando quindi i limiti imposti dalla tecnica basata su rappresentazione MFCC. In particolare, la tecnica proposta combina un learning HMM, ad una sintesi di tipo unit-selection, dove l'unità base scelta è il trifone context-dependent. Come già detto, la sintesi unit-selection ha la caratteristica di produrre un suono naturale di buona qualità a discapito però della memoria necessaria per memorizzare le unità scelte nel database utilizzato per la sintesi. Si è parlato infatti nella sezione

5.3.3 anche del trade-off nella scelta della lunghezza delle unità: scegliere come unità base il fonema, porta ad uno small-database ma ad una forte discontinuità nella concatenazione; mentre scegliere unità più lunghe porta ad una maggiore naturalezza ma ad un maggiore effort per quanto riguarda l'occupazione di memoria. Nel metodo proposto, si decide allora di utilizzare come unità base il trifone context-dependent, che con la sua proprietà di dipendenza dal contesto permette di mitigare il problema delle discontinuità richiedendo una quantità di memoria ragionevole. Grazie alle proprietà di invertibilità della trasformata, l'audio potrà poi essere ricostruito on-the-fly nella fase finale di sintesi, applicando la Inverse MDCT (IMDCT). I risultati sperimentali mostrano la validità dell'approccio proposto.

5.5.1 System overview

La Fig. 5.8 mostra lo schema a blocchi del sistema, che si compone principalmente di due fasi:

- **learning stage**, *off-line stage*: estrae dal database in ingresso (audio e testo allineati) le features MDCT e deriva tramite un modeling HMM i sottostati per ogni sequenza di fonemi in ingresso;

- **synthesis stage**, *off-line & on-line stage*: dato il testo in ingresso, restituisce l'audio sintetizzato del testo fornito. In particolare, sulla base delle sequenze di sottostati precedentemente classificate, memorizza le features MDCT associate per ogni trifone context-dependent. Il testo in ingresso è normalizzato in una sequenza di trifoni. Ad ogni trifone sono associati più MDCT feature vectors O; si sceglie il vettore ottimo O^* per ogni trifone, sulla base di regole di prosodia, durata e pitch. I features vector O^* determinati per ogni trifone che compone il testo in ingresso sono concatenati per determinare la sequenza ottima, e la trasformata inversa IMDCT più lo step overlap-and-add, restituiscono il segnale vocale sintetizzato corrispondente al testo in ingresso.

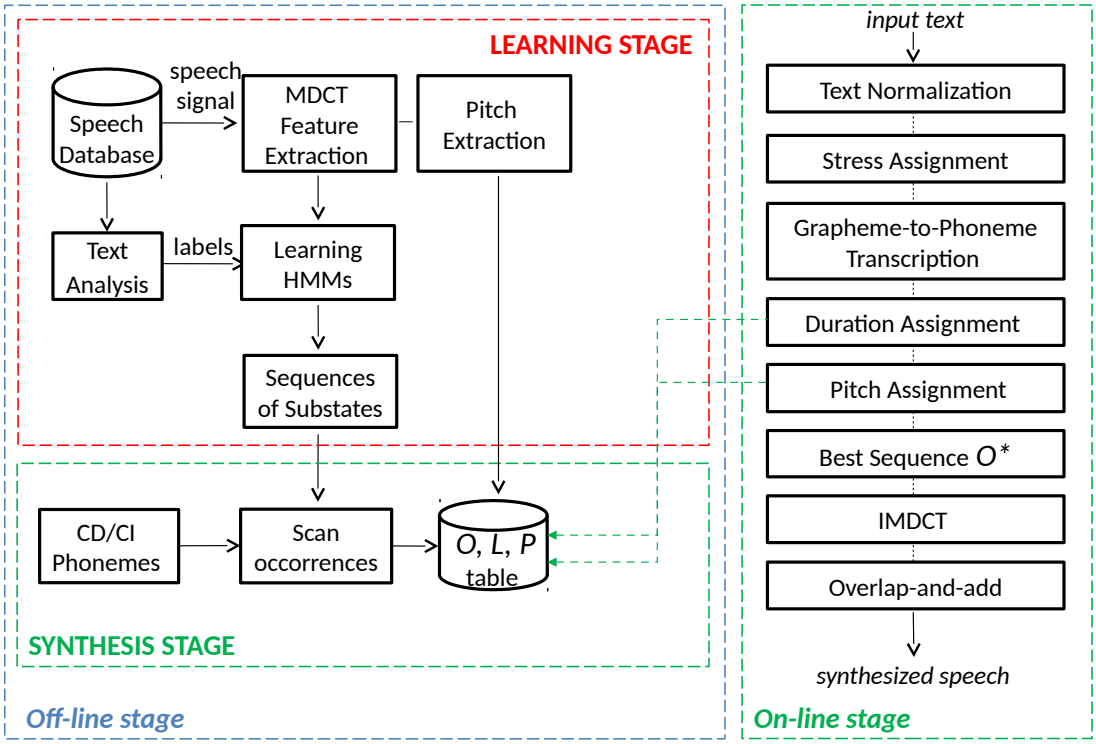


Figura 5.8: Schema a blocchi del sistema di sintesi vocale MDCT-based.

5.5.2 MDCT feature vector

Si rappresenti il segnale campionato S come una sequenza di $T+1$ blocchi di D campioni:

$$S = [s_1^T, s_2^T, \ldots, s_{T+1}^T]^T \in \mathbb{R}^{(T+1)D\times 1} \quad , \tag{5.2}$$

dove

$$s_t \in \mathbb{R}^{D\times 1} \tag{5.3}$$

è il singolo blocco di lunghezza D.

Nel campionamento con overlap, si ottiene una sequenza di frames

$$X = [x_1^T, x_2^T, \ldots, x_T^T]^T \in \mathbb{R}^{T(2D)\times 1} \quad , \tag{5.4}$$

dove

$$x_t = \begin{pmatrix} x_t^L \\ x_t^R \end{pmatrix} = \begin{pmatrix} s_t \\ s_{t+1} \end{pmatrix} \in \mathbb{R}^{2D\times 1}, \quad t = 1, \ldots, T \tag{5.5}$$

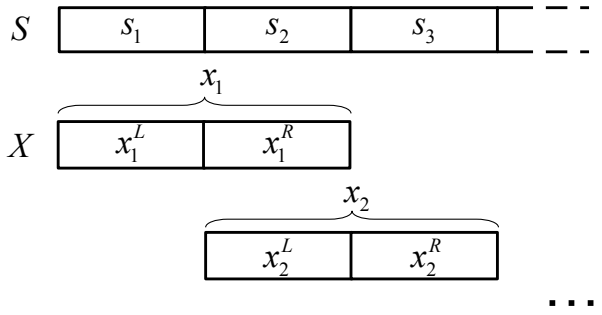


Figura 5.9: Le sequenze S, X, e le regioni di overlap tra i diversi blocchi.

è il singolo frame corrispondente ad una finestra di lunghezza $2D$.

Le sequenze S, X, e le regioni di overlap sono raffigurate in Fig. 5.9. I blocchi x_t e x_{t+1} presentano un overlap di lunghezza D, e vale la seguente condizione:

$$x_t^R = x_{t+1}^L \ . \tag{5.6}$$

Il metodo comunemente utilizzato per la parametrizzazione dello speech è il source-filter model che porta all'estrazione di parametri (features) quali: linear predictive coding (LPC), MFCCs, perceptual linear prediction (PLP) coefficients ecc. Tra questi, le features MFCCs si sono dimostrate essere quelle di maggior successo grazie alla loro particolare robustezza all'ambiente e flessibilità [105]. L'estrazione di features MFCC corrisponde ad una trasformazione F tale che

$$\hat{o}_t = F x_t \tag{5.7}$$

dove il vettore $\hat{o}_t$ rappresenta il feature vector appartenente ad un opportuno sottospazio.

Il problema principale nella sintesi vocale è che, dato un vettore $\hat{o}_t$ derivato dalla trascrizione, il frame signal x_t non può essere univocamente derivato da (5.7) perché la trasformazione F non è invertibile. Al fine di risolvere questo problema, la tecnica proposta va a sostituire la rappresentazione MFCC con una rappresentazione MDCT che garantisce la perfetta ricostruzione del segnale e permette un 50% di overlap tra i blocchi senza incrementare il data rate.

La MDCT è una trasformata lapped [174] che è un tipo di trasformazione lineare discreta a blocchi . La MDCT fu proposta da [175], in seguito al precedente lavoro di [176] per sviluppare il principio di base della cancellazione dell'aliasing

nel dominio del tempo.

La MDCT di un segnale di lunghezza finita $y(n)$, $n = 0, 1, \ldots, 2M-1$ è definita come segue. Dato un blocco in ingresso, $y(n)$, i coefficienti della trasformata, $Y(k)$, per $0 \leq k \leq M-1$, sono ottenuti per mezzo della MDCT, definita come:

$$Y(k) = \sum_{n=0}^{2M-1} w(n)y(n)h_k(n) \ , \quad k = 0, 1, \ldots, M-1 \tag{5.8}$$

dove i termini $h_k(n)$ della matrice di trasformazione sono:

$$h_k(n) = \sqrt{\frac{2}{M}} \cos\left(\frac{(2n+1+N)(2k+1)\pi}{4M}\right) \tag{5.9}$$

e $w(n)$ è una finestra simmetrica con $0 \leq n \leq 2M-1$.

La trasformata inversa IMDCT è data da

$$\widetilde{y}(n) = w(n) \sum_{k=0}^{M-1} Y(k)h_k(n) \ , \quad n = 0, 1, \ldots, 2M-1 \ . \tag{5.10}$$

I campioni ricostruiti $y(n)$, per $0 \leq n \leq M-1$, sono ottenuti per mezzo della IMDCT attraverso un processo di overlap-and-add definito come

$$y(n) = \sum_{k=0}^{M-1} \big[w(n)Y(k)h_k(n) + w(n+M)Y^P(k)h_k(n+M)\big] \ , \quad n = 0, 1, \ldots, M-1 \tag{5.11}$$

dove $Y^P(k)$ denota il blocco precedente dei coefficienti della trasformata.

I vincoli di perfetta ricostruzione sono

$$\begin{cases} w(2M-1-n) = w(n) \\ w^2(n) + w^2(n+M) = 1 \end{cases} \ . \tag{5.12}$$

La proprietà principale della MDCT è che questa non è una trasformazione ortogonale, e la ricostruzione perfetta del segnale può essere realizzata solo tramite un processo di overlap-and-add. Indicando con $(A_1 \ A_2) \in \mathbb{R}^{D \times 2D}$ e $W = \text{diag}(W_L \ W_R)$, $W_L, W_R \in \mathbb{R}^{D \times D}$ rispettivamente le matrici che rappre-

sentano la MDCT e la finestra [177], e con o_t il vettore delle features MDCT, risulta

$$o_t = (A_1\ A_2)\, W\, x_t = (A_1\ A_2)\, W \begin{pmatrix} s_t \\ s_{t+1} \end{pmatrix} = A_1\, W_L\, s_t + A_2\, W_R\, s_{t+1} \quad (5.13)$$

dove A_1 , $A_2 \in \mathbb{R}^{D\times D}$. In forma matriciale si ottiene

$$O = A_W\, S \quad (5.14)$$

con

$$A_W = \begin{pmatrix} A_1W_L & A_2W_R & \cdots & \cdots & 0 \\ 0 & A_1W_L & A_2W_R & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & A_1W_L & A_2W_R \end{pmatrix} \in \mathbb{R}^{TD\times(T+1)D} \quad (5.15)$$

e

$$O = [o_1^T\ ,\ o_2^T\ ,\ \ldots\ ,\ o_T^T]^T \in \mathbb{R}^{TD\times 1} \quad (5.16)$$

è il vettore delle features MDCT corrispondente al segnale S.

5.5.3 Learning stage

5.5.3.1 HMM acoustic model training

L'algoritmo di sintesi vocale proposto determina la sequenza X del segnale sintetico, data la sequenza O delle features corrispondenti alla trascrizione (sequenza di fonemi) H che deve essere sintetizzata.

In un modeling HMM, occorre innanzitutto derivare la sequenza di stati che genera la sequenza O. A questo scopo, si definisca

$$P(O, Q/\lambda) = \pi_{\theta_0} \prod_{t=1}^{T} a_{\theta_{t-1}\theta_t} b_{\theta_t}(o_t) \quad (5.17)$$

come la pdf congiunta di O e Q, dato il modello λ, dove

$$Q = \{\theta_1, \theta_2, \ldots, \theta_T\} = \{(q_1, i_1), (q_2, i_2), \ldots, (q_T, i_T)\}\ , \quad (5.18)$$

essendo $\theta_t = (q_t, i_t)$ il sottostato associato alla mistura Gaussiana i_t dello stato q_t all'istante di tempo t, cioè

$$b_{\theta_t}(o_t) = (2\pi)^{-D/2} \left|U_{\theta_t}\right|^{-1/2} \cdot \exp\left\{-\frac{1}{2}(o_t - \mu_{\theta_t})^T U_{\theta_t}^{-1}(o_t - \mu_{\theta_t})\right\} \tag{5.19}$$

con $\mu_{\theta_t} \in \mathbb{R}^{D\times 1}$, $U_{\theta_t} \in \mathbb{R}^{D\times D}$. π_{θ_0} è la initial-state probability, e $a_{\theta_{t-1}\theta_t}$ è la state-transition probability.

Dato che $H = \{h_1,\ h_2,\ \dots\}$ definisce la sequenza di fonemi, è possibile restringere la formulazione matematica ad un singolo fonema h. Dato il fonema h, le sequenze O e Q sono scelte in modo tale da massimizzare la pdf congiunta

$$P(O, Q/\lambda) = P(O/Q,\, \lambda)P(Q/\lambda)\ , \tag{5.20}$$

che rappresenta la likelihood del set $\chi = \{O, Q\}$. La sequenza Q è ottenuta durante la fase di learning in modo tale da soddisfare $\max P(Q/\lambda)$. Al termine del training, ad un dato h corrisponde un set $\{Q_1,\ Q_2,\ \dots\}$ di sequenze di sottostati, ad ognuna delle quali corrisponde un set di features $\{O_1, O_2, \dots\}$.

5.5.3.2 Maximum likelihood estimation

In questa sezione, si farà brevemente cenno al metodo di stima della maximum likelihood, applicato nel prima versione del framework di sintesi proposto e descritto dettagliatamente in [178].

Lo schema a blocchi dell'algoritmo è proposto in Fig. 5.10.

Lo schema mostra i due step fondamentali che compongono lo schema: il *learning stage* (off-line stage) ed il *synthesis stage* (on-line stage). Formalmente gli step sono gli stessi applicati nell'approccio attuale. In pratica però, mentre si è lasciato invariato il learning stage, il synthesis stage utilizza un metodo completamente diverso. In questo schema, infatti, lo stage di sintesi, sulla base delle sequenze di sottostati classificate nella fase di training HMM, determina la sequenza ottima di stati Q_{best} e la corrispondente sequenza ottima di features O^* per ogni fonema in ingresso, sulla base del criterio di massima verosimiglianza. Saranno queste le sequenze ad essere concatenate per ottenere il segnale sintetizzato in uscita tramite IMDCT.

Dal punto di vista matematico, al termine dell'addestramento, ad un dato h corrisponde un set $\{Q_1,\ Q_2,\ \dots\}$ di sequenze di sottostati; da qui si sceglie Q

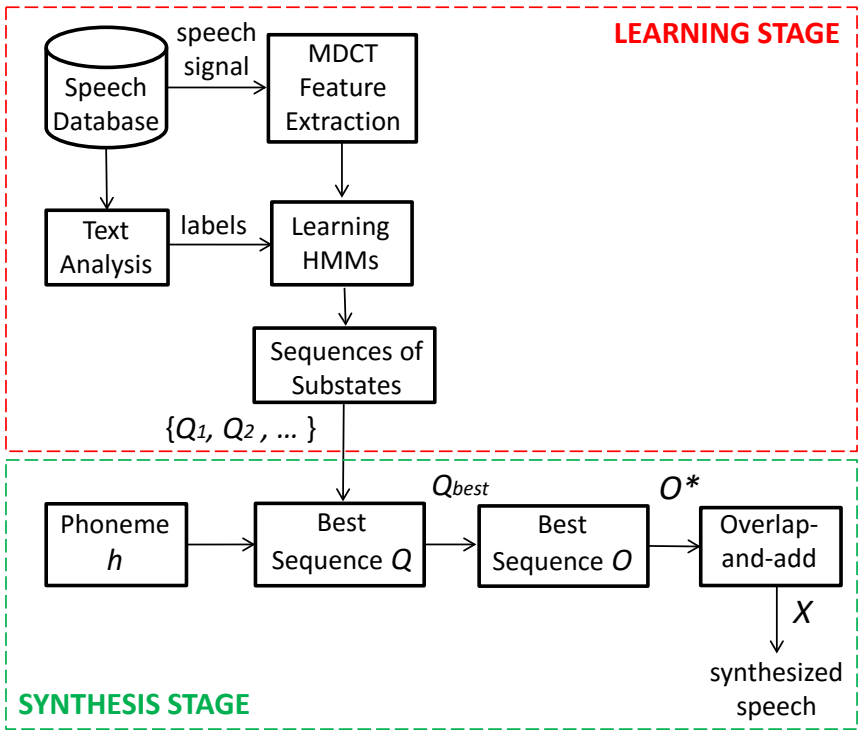


Figura 5.10: Schema a blocchi della prima versione del framework di sintesi vocale proposto.

come la sequenza che soddisfa

$$Q = Q_{best} = \arg\max_i P(Q_i/\lambda) \ . \tag{5.21}$$

Una volta ottenuta Q, la sequenze O è data dal massimo della likelihood $\log P(O/Q, \lambda)$ che può essere scritta come

$$\mathcal{L}(O) = \log P(O/Q, \lambda) = \sum_{t=1}^{T} \log b_{\theta_t}(o_t) \ . \tag{5.22}$$

Dopo alcune manipolazioni, si ottiene

$$\mathcal{L}(O) = -\frac{1}{2} O^T U^{-1} O + O^T U^{-1} M + k \tag{5.23}$$

dove

$$U^{-1} = \text{diag}\left[U^{-1}_{q_1,i_t}\,,\, U^{-1}_{q_2,i_t},\,\dots\,,\, U^{-1}_{q_T,i_t}\right] \in \mathbb{R}^{TD\times TD}\,,$$
$$M = \left[\mu^T_{q_1,i_t}\,,\, \mu^T_{q_2,i_t},\,\dots\,,\, \mu^T_{q_T,i_t}\right]^T \in \mathbb{R}^{TD\times 1} \quad (5.24)$$

e

$$k = k' + k''\,, \quad (5.25)$$

essendo

$$k' = \sum_{t=1}^{T} \log\left(2\pi\right)^{-D/2} |U_{\theta_t}|^{-1/2}\,, \quad k'' = \mu^T_{q_t,i_t} U^{-1}_{q_t,i_t} \mu_{q_t,i_t}\,. \quad (5.26)$$

La sequenza O può essere derivata in modo tale da massimizzare (5.23).

Avendo ricavato la sequenza ottima Q_{best} di sottostati per un dato fonema h, a questa sequenza corrisponde un set $\{O_1, O_2, \dots\}$ di sequenze di features ed un set di likelihood values $\{\mathcal{L}(O_1), \mathcal{L}(O_2), \dots\}$. Al fine di massimizzare la pdf congiunta (5.20), viene scelta la sequenza $O^* = \{O_1, O_2, \dots\}$ tale che $\mathcal{L}(O^*) = \max\{\mathcal{L}(O_1), \mathcal{L}(O_2), \dots\}$. Al termine, una volta che è stata ottenuta la sequenza ottima di feature vectors O^* , si deriva la sequenza X dei frames del segnale sintetizzato tramite un processo di overlap-and-add.

Questo approccio, come mostrato in [178] ha portato a risultati oggettivi (analisi spettrale e distanze spettrali) positivi, se confrontato con altri tools di sintesi vocale. Questi test sono stati però condotti su unità piccole (fonemi, sillabe, parole). Nel momento in cui il metodo è stato applicato al fine di realizzare una sintesi continous-speech sono emerse alcune problematiche, sopratutto legate alla natura del suono (poco naturale e con forti discontinuità) a cui si è deciso di far fronte, modificando l'algoritmo con l'approccio "ibrido" proposto in questo capitolo (Fig. 5.8).

Di seguito, quindi, si prosegue con la descrizione dell'algoritmo corrente.

5.5.4 Synthesis stage

Non è possibile pensare di generare una frase semplicemente concatenando le features ottenute dal modello acustico; la fonetica e la prosodia sono essenziali nella produzione di un suono naturale ed intelligibile e come tali, questi argomenti vengono analizzati in dettaglio in tutti i principali testi dedicati alla sintesi vocale [179, 180], testi a cui si rimanda per un approfondimento specifico. Si il-

lustreranno, di seguito, gli elementi fondamentali della fonetica e prosodia di cui si è tenuto conto nella realizzazione dei tre steps principali della fase di sintesi dell'algoritmo sviluppato:

- Step 1: Determinazione dell'accento;
- Step 2: Determinazione della lunghezza per ogni sequenza di stati associata al trifone context-dependent;
- Step 3: Estrazione del pitch e tuning.

5.5.4.1 Analisi dell'accento

In linguistica, l'accento di una parola è un tratto prosodico, soprasegmentale, che permette, nella realizzazione fonetica di una parola, la messa in rilievo di una delle sillabe che la compongono (distingue tra le sillabe che vengono percepite come "più forti" da quelle "più deboli"). Si distingue tra accento primario, indicato nella trascrizione IPA[8] con il simbolo ˈ e l'accento secondario, indicato con il simbolo ˌ situato su una sillaba precedente quella su cui cade l'accento principale. In fonetica per accento secondario s'intende l'accento fonico (mai grafico, tranne in caso di trascrizione fonetica di alcuni dizionari ed enciclopedie). In italiano hanno due o più accenti tutte le parole composte: "accèndisìgari", "àlfabèto", "buònaséra", "pómodòro", "sàliscéndi", ... Quindi, ad esempio, per "pómodòro" la trascrizione IPA sarà: /ˌpomoˈdɔro/.

Le regole grammaticali delle varie lingue del mondo disciplinano in misura molto diversa la posizione dell'accento: esso dunque, a seconda della lingua, può essere completamente fisso (come nel caso del finlandese, del polacco, del francese o di parecchie lingue artificiali, come l'esperanto) o relativamente libero (come per l'italiano). In alcune lingue può cadere indifferentemente su qualsiasi parte della parola (prefisso, tema, suffisso, desinenza), mentre in altre deve sottostare ad alcune regole: in latino e greco, ad esempio, non può risalire oltre la terzultima sillaba, ed in latino non può nemmeno trovarsi sull'ultima. Ad ogni modo la

[8]International Phonetic Alphabet: è un sistema di scrittura alfabetico, basato principalmente sull'alfabeto latino, utilizzato per rappresentare i suoni delle lingue nelle trascrizioni fonetiche. L'IPA nasce su iniziativa dell'Associazione fonetica internazionale, con l'intenzione di creare uno standard per trascrivere in maniera univoca i suoni (tecnicamente, i foni) di tutte le lingue conosciute: questo è possibile poiché ad ogni segno IPA corrisponde un solo suono e viceversa, senza possibilità di confusione tra lingue diverse.

funzione principale dell'accento è quella di indicare le unità significative (parole), ciascuna delle quali ha di norma un solo accento.

In italiano possiamo dividere le parole in gruppi a seconda della posizione in cui cade l'accento:

- tronche: ultima sillaba (es. caf-fè)
- piane: penultima sillaba (es. ma-tì-ta)
- sdrucciole: terzultima sillaba (es. te-lè-fo-no)
- bisdrucciole: quartultima sillaba (es. cà-pi-ta-no)

Nell'algoritmo proposto, la determinazione dell'accento ha un ruolo fondamentale per la determinazione della durata. Quindi, nella normalizzazione del testo in ingresso al sintetizzatore, è stata considerata anche questa informazione, e la trascrizione fonetica contiene sia il simbolo dell'accento primario ˈ che dell'accento secondario ˌ . In particolare, la trascrizione fonetica è stata realizzata tramite l'ausilio del software "eSpeak" [82]. Il software "eSpeak" consiste in un software open source che realizza un sintetizzatore vocale ma che fornisce anche numerosi tools utilizzabili indipendentemente dal sintetizzatore, uno di questi è il trascrittore fonetico. Purtroppo, sebbene il software supporti numerosi linguaggi, la lingua di default è l'inglese e per quanto riguarda l'italiano, il tool presenta alcuni bugs a cui si è rimediato applicando numerose patches per ottenere una corretta trascrizione e determinazione dell'accento.

5.5.4.2 Analisi della durata

Klatt rules Come già detto, Dennis Klatt sviluppò un sistema di sintesi vocale, chiamato MITalk, e con esso definì un ampio set di regole deterministiche [150, 151, 141], applicabili indipendentemente dalla tecnica di sintesi adottata. In queste regole, vengono definiti due parametri fondamentali: una *minimum duration M* ed una *inherent duration*, dove la inherent duration è una sorta di media della durata del fonema nel modello acustico realizzato. La durata fonetica è definita come:

$$Duration = [(inherent\ duration - minimum\ duration) * A] + minimum\ duration \tag{5.27}$$

dove il termine A è una costante calcolata tramite successive applicazioni di un insieme di regole. Questa costante varia a seconda del fonema e del contesto linguistico considerato; così ad esempio si ha:

- *Allungamento pre-pausale*: la vocale o la sillaba che precede una pausa è allungata di un fattore 1.4
- *Accorciamento da mancanza di accento*: tutte le vocali non accentate devono essere ridotte di un fattore 0.7
- *Allungamento da presenza di accento*: tutte le vocali che portano accento devono essere allungate di un fattore 1.4
- *Accorciamento in sillaba chiusa*: tutte le vocali che in una sillaba sono seguiti da una consonante devono essere accorciate di un fattore 0.8
- *Accorciamento in contesto occlusivo*: tutte le vocali che precedono una occlusiva sorda sono accorciate di un fattore 0.7

Si rimanda a [141] per l'elenco completo delle regole.

Klatt rules modificate Nell'algoritmo proposto, la durata fonica viene stabilita seguendo l'approccio di Klatt ma modificando le regole per avere una dipendenza dalla durata media dell'accento primario, durata stimata dal modello acustico realizzato.

5.5.4.3 Analisi del pitch

Occorre distinguere tra pitch e frequenza fondamentale F0. Strettamente parlando, il pitch è ciò che l'ascoltatore percepisce, mentre la frequenza fondamentale F0, in un segnale periodico, può essere definita semplicemente come il reciproco del periodo, ma essendo il segnale vocale puramente aperiodico, deve essere definita in questo caso come la frequenza fondamentale del tratto vocale [179]. In prosodia, F0 è vista come la diretta espressione dell'intonazione; e spesso l'intonazione è definita come l'uso linguistico di F0. Chiaramente l'ascoltatore non percepisce distintamente tutte le frequenze della campana con centro a F0, ma piuttosto una versione processata di questa. Il meccanismo esatto non è noto, ma è come se l'ascoltatore interpolasse tutta la campana di frequenze in modo da produrre un andamento continuo ed interrotto della stessa.

Nell'algoritmo proposto, una volta applicate le regole fondamentali della prosodia, si è analizzato il segnale ottenuto. Il segnale sintetizzato con le sole regole sulla durata, si è dimostrato sì intelligibile, ma presentava delle forti discontinuità all'ascolto. Si è quindi deciso di indagare su queste discontinuità applicando l'*Enhanced Autocorrelation algorithm* (EAC); questo algoritmo mette in evidenza il contorno della frequenza fondamentale (F0 contour) dell'audio in ingresso e fornisce una rappresentazione matematica della variazione del pitch in modo tale da permettere una valutazione della somiglianza tra suoni generati anche da strumenti diversi. La Fig. 5.11 mostra lo spettrogramma (ottenuto applicando l'algoritmo EAC) del segnale sintetizzato con le sole regole della prosodia, e le successive manipolazioni sul pitch per ottenere uno spettrogramma sempre più "vicino" al segnale audio originale.

Dalla Fig. 5.11 si evince che manipolando il pitch, si riesce effettivamente a migliorare il segnale sintetizzato, mitigando le discontinuità ed ottenendo un segnale più gradevole all'ascolto e più simile all'originale. Queste manipolazioni consistono semplicemente nella scelta opportuna del pitch per ogni trifone sulla base del pitch medio del parlatore. All'algoritmo, si aggiunge quindi una nuova regola: oltre alla scelta della durata su basi prosodiche e accento, nella costruzione del testo in ingresso, occorre selezionare dal database il trifone corrispondente al pitch scelto, o comunque entro una soglia del pitch stabilita. Questo ovviamente prevede uno step precedente di estrazione del pitch che viene fatto in fase di modeling.

5.5.5 Risultati sperimentali

I test condotti sono stati realizzati nell'ottica di effettuare un'analisi completa del segnale vocale generato. Si è partiti da un'analisi preliminare del suono, realizzata su unità elementari, quindi su singole vocali e consonanti fino ad arrivare alla loro concatenazione per ottenere parole della lingua italiana; per poi giungere alla realizzazione di intere frasi. L'analisi delle unità elementari (fonemi e parole) è stata condotta utilizzando valutazioni oggettive, ossia analisi dello spettrogramma (sezione 5.5.5.2), distanze spettrali (sezione 5.5.5.2), oltre ovviamente a prove di ascolto condotte personalmente. Data la bontà dei risultati ottenuti, nel tool sviluppato sono state perfezionate le regole di tuning della prosodia e del pitch, precedentemente illustrate (sezione 5.5.4) per poter ottenere la sintesi di unità del linguaggio più complesse (frasi, pragrafi, ...). Di

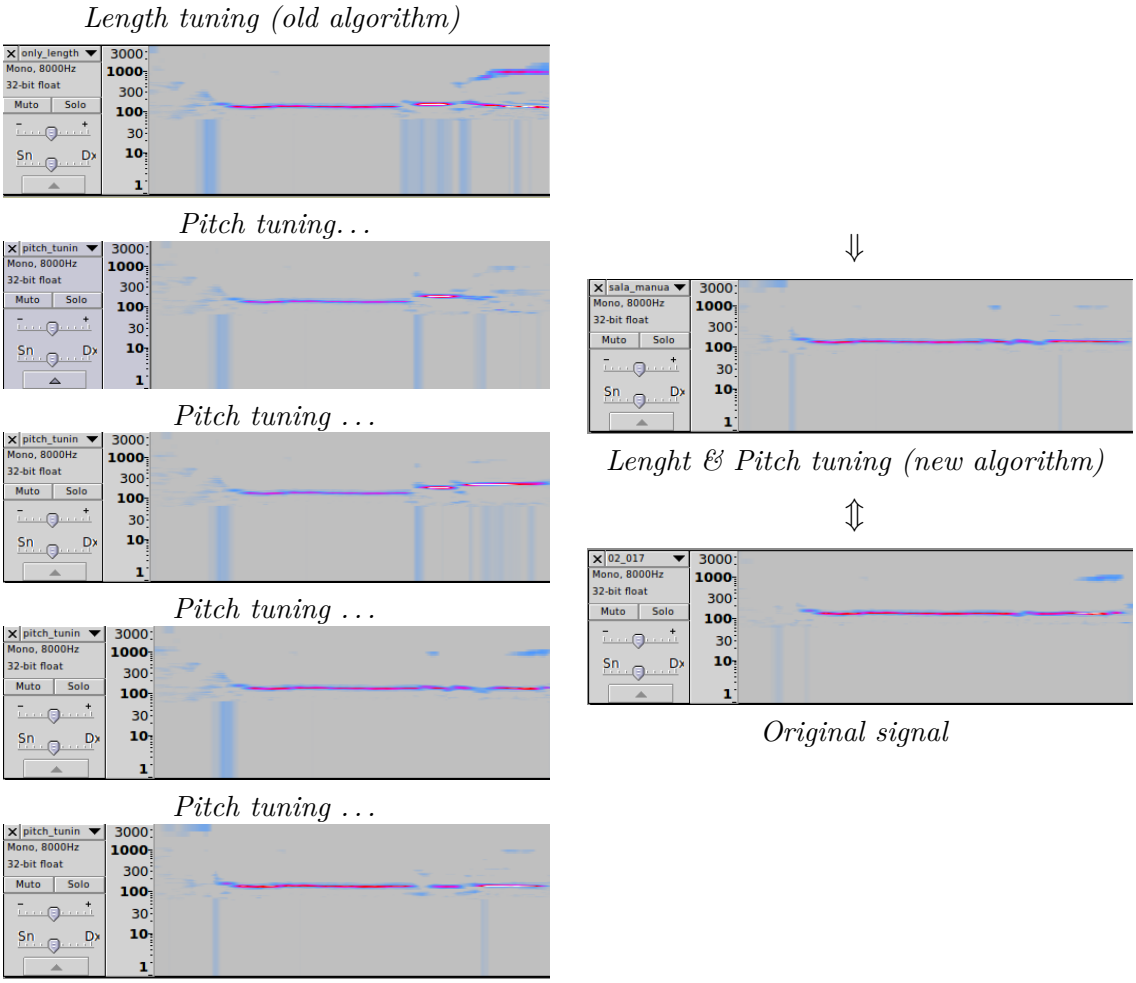


Figura 5.11: Successive manipolazioni del pitch del segnale sintetizzato tramite le sole regole della prosodia al fine di mitigare le discontinuità.

queste ovviamente, il semplice test oggettivo o l'ascolto personale (ormai viziato da innumerevoli prove di ascolto) non potevano essere sufficienti per valutarne la bontà. Come suggerisce quindi la letteratura, il metodo migliore per valutare la qualità dell'audio sintetizzato è quello di ricorrere ai test percettivi (sezione 5.4). Si è quindi scelto di condurre una campagna di test percettivi come descritto nella sezione 5.5.5.3.

training material:

language: Italian

speaker: female

book title: “Alice nel Paese delle Meraviglie”, by Lewis Carrol

duration: ~ 2 h

source: Liber Liber (`http://www.liberliber.it/`)

features:

Hanning window: 20 ms
overlap: 10 ms
feature type: 80 MDCT coefficient
audio: 8 kS/s, mono, 16 bit

acoustic model:

type: context dependent
gaussians per state: 8
phonemes: 46 - triphones: 6033
ci states: 138 - cd states: 18099
total HMM used: 46
states per HMM: 3

Tabella 5.2: Parametri utilizzati per il training HMM del modello acustico.

5.5.5.1 Acoustic model training

Il primo passo per realizzare gli esperimenti volti a validare l'approccio proposto, consiste nel realizzare il training del modello acustico HMM.

Il materiale adottato per il training è riassunto in Tab. 5.2: consiste di sole 2 h di registrazioni audio di voce femminile, estratte da un audiolibro in lingua italiana

Il vettore delle features è derivato applicando la MDCT ad un frame x_t di lunghezza $2D = 20$ ms, sovrapposto del 50% con il frame successivo. Quindi, con una frequenza di campionamento di 8 kHz, si ottiene una lunghezza del frame di 80 campioni (corrispondente alla lunghezza dell'overlap).

Il training è stato realizzato con l'ausilio dell'algoritmo di Baum-Welch, che realizza una stima EM dei parametri del modello audio.

5.5.5.2 Test oggettivi

Di seguito i test oggettivi realizzati: *analisi dello spettrogramma* e distanza spettrale tramite la *Itakuro-Saito Measure* (ISM) [181, 182].

Analisi spettrogramma

Vowel synthesis Per validare l'approccio proposto, lo schema descritto è stato utilizzato per sintetizzare le cinque vocali della lingua italiana ($|a|$, $|e|$, $|i|$, $|o|$, $|u|$), i cui spettrogrammi sono mostrati in Fig. 5.12. Per completezza, lo stesso fonema è stato sintetizzato usando il software "eSpeak" [82] selezionando una voce femminile derivata dal progetto MBROLA [183, 184] (it-4) a partire da audio appartenente al database ITC-irst. Il progetto MBROLA fornisce databases di difoni per un ampio numero di linguaggi. In particolare, il termine MBROLA si riferisce ad un algoritmo di sintesi vocale diphone-based [185]. "eSpeak" consiste in un software open source che realizza un sintetizzatore vocale e che può essere utilizzato come front-end per le MBROLA diphone voices.

Nella Fig. 5.12, il primo spettrogramma (a) rappresenta il segnale audio originale, il secondo (b) ed il terzo (c) rappresentano invece le vocali sintetizzate con l'approccio proposto e con la tecnica a difoni rispettivamente. Si evince che gli spettrogrammi realizzati con la tecnica suggerita si comportano in maniera molto simile a quelli del segnale originale, mentre la tecnica a difoni dà risultati abbastanza diversi da quelli attesi.

Word synthesis La Fig. 5.13 mostra gli spettrogrammi di tre parole del dizionario italiano *topo* (| t o p o |), *casa* (| k a z a |), *Alice* (| a l i ʧ e |). Anche in questo caso, gli spettrogrammi ottenuti con la tecnica proposta, sono molto vicini all'originale.

Itakura-Saito Measure La qualità del segnale sintetizzato è stata ulteriormente valutata utilizzando la distanza di Itakura-Saito Measure (ISM) [181, 182]. Si è misurata la ISM tra una popolazione di osservazioni estratte dal database di training e la realizzazione più probabile della stessa parola (parola target) e si è verificato con successo, come mostrato in Tab. 5.3, che la ISM tra la parola sintetizzata e la parola target si attestano nel range di valori determinato per la popolazione della parola originale, confermando la bontà del segnale sintetizzato.

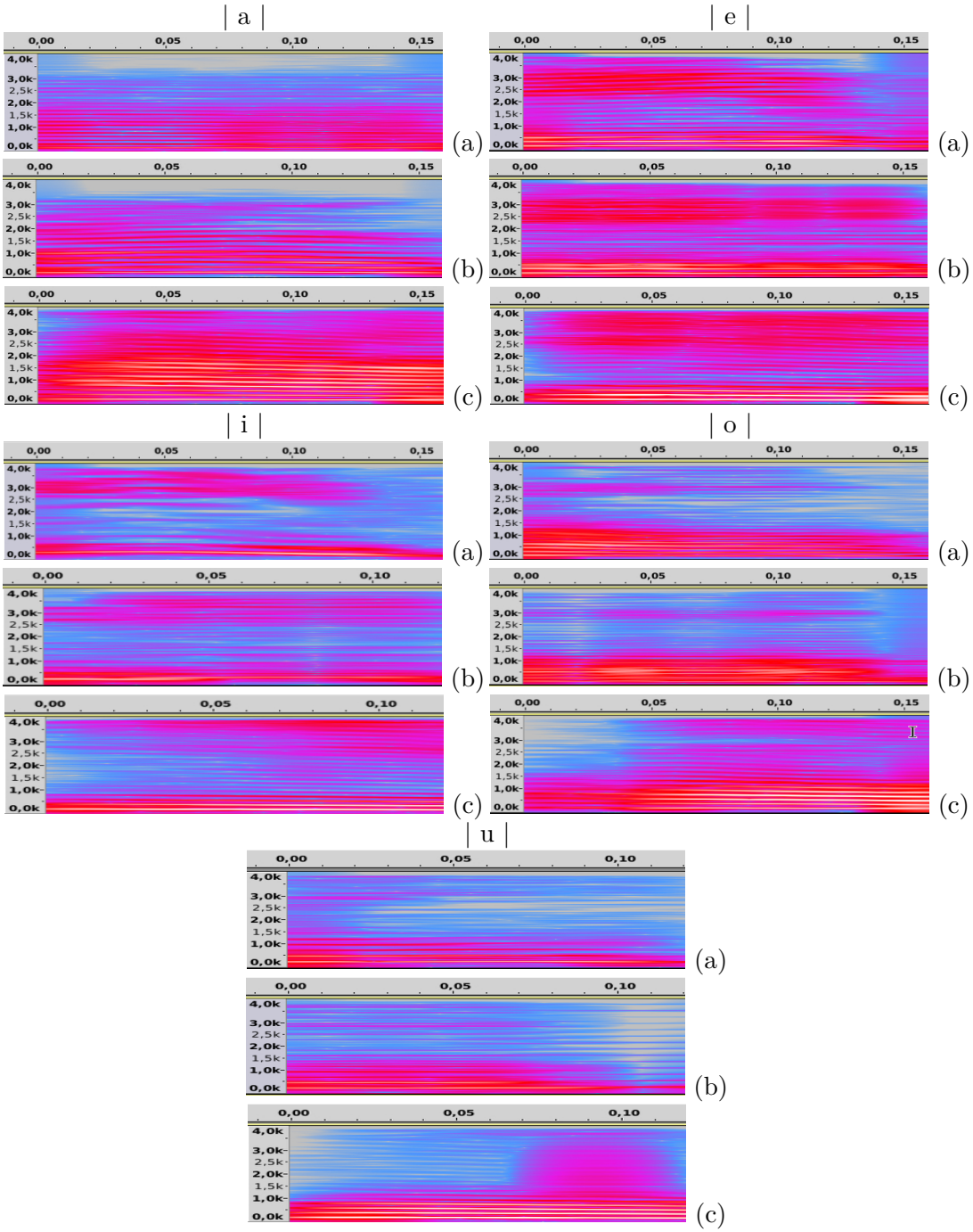


Figura 5.12: Spettrogrammi delle vocali italiane $|a|$, $|e|$, $|i|$, $|o|$, $|u|$ per: (a) segnale originale, (b) segnale sintetizzato con la tecnica proposta, e (c) segnale sintetizzato tramite la tecnica a difoni.

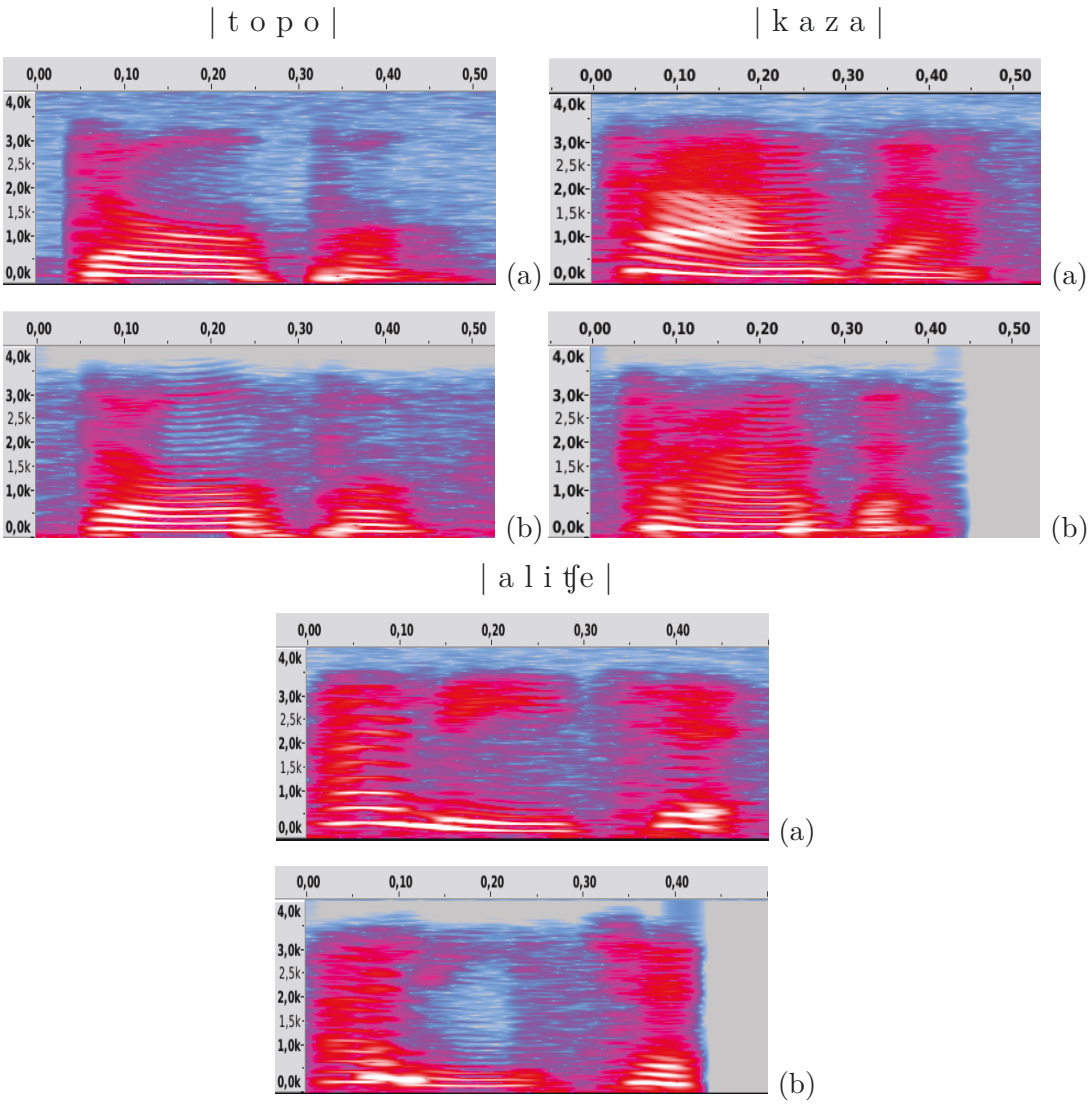


Figura 5.13: Spettrogrammi delle parole italiane *topo* (| t o p o |), *casa* (| k a z a |), *Alice* (| a l i ʧe |) per: (a) segnale originale, (b) segnale sintetizzato tramite la tecnica proposta.

5.5.5.3 Test soggettivi

Nel campo della sintesi vocale, come già illustrato nella sezione 5.4, il metodo di valutazione più efficace consiste nella realizzazione di test soggettivi in grado di valutare la bontà dell'audio sintetizzato a partire dalla percezione del singolo utente.

Tabella 5.3: Itakura-Saito measure per una popolazione di osservazioni della parola target e le parole sintetizzate.

Word	**Original Words**		**Synthesized Word**
	min	max	
| a l ˈi ʧe |	0.5463	29.0485	2.4101
| t ɛ r r a |	0.8861	4.2467	1.7028
| t ˈo p o |	1.2448	17.9420	3.5390
| t r ˈo pːo |	4.9798	30.6913	10.3629
| v ˈo ʧe |	0.6322	31.4948	14.0029
| b a m b ˈi n o |	12.3342	36.9731	5.7572
| s o r ˈe lːa |	1.6688	5.3084	1.7281
| p a r l ˈa r e |	0.9784	2.3326	1.7080
| m o m ˈe n t o |	0.8668	12.3031	4.2499
| k o n ˈi λo |	1.2304	8.2788	1.6444
| t ˈa v o l o |	0.9940	1.7989	1.6899

Nella campagna di test condotti, si è preso come riferimento lo standard definito dalla Blizzard Challenge, che nel corso degli anni (a partire dal 2005), pur variando di anno in anno alcune specifiche dei test, ha mantenuto principalmente due obiettivi:

- Costruire la voce sulla base di un dato database,
- Analizzare la bontà della voce, sintetizzando un numero variabile di utterances, valuate secondo i test:
  - **MOS**: *Naturalness* - *Intelligibility* (scale 1:5);
  - **DMOS**: *Naturalness* - *Similarity* (scale 1:5);
  - **SUS**: *Intelligibility* (WER).

  e da un numero variabile di ascoltatori ripartiti in tre gruppi:
  - **Speech experts**
  - **Undergraduates**
  - **Volunteers**

Seguendo questo approccio, si è scelto di condurre un test composto da tre sezioni di ascolto:

- *Test 1:* sezione contenente 15 utterances sintetizzate utilizzando parole non appartenenti al materiale di training e relative al topic "Conversation" [159], per valutarne la naturalezza e l'intelligibilità secondo la scala MOS;

- *Test 2:* sezione contenente 15 utterances sintetizzate utilizzando parole appartenenti al materiale di training per permetterne il confronto con l'audio originale in un test di tipo DMOS volto a valutare la naturalezza delle frasi e la loro somiglianza rispetto all'originale;

- *Test 3:* è il test più difficile per l'orecchio umano; riguarda la comprensione di 15 utterances di tipo SUS, generate quindi interpolando parole sintatticamente corrette ma in un ordine grammaticalmente non correlato; di queste utterances viene richiesta la trascrizione al fine di valutarne l'intelligibilità tramite il calcolo del WER.

Il set up utilizzato nei test condotti, è riassunto in Tab. 5.4

subjective test type:
naturalness: 15 MOS (testing material) + 15 DMOS (training material) sentences
similarity: 15 DMOS sentences (training material)
intellegibility: 15 MOS (testing material) + 15 SUS (testing material) sentences

listeners type:
speech experts: 5
undergraduates: 6
volunteers: 4

Tabella 5.4: Set up dei test soggettivi condotti.

I risultati ottenuti dalla campagna di test percettivi, sono riportati in Tab. 5.5.

Tabella 5.5: Valori del MOS, DMOS, SUS ottenuti da una campagna di test soggettivi su 15 soggetti.

MOS		**DMOS**		**SUS**
Naturalness	**Intelligibility**	**Naturalness**	**Similarity**	**WER**[%]
3.18	4.52	3.49	3.58	4.37

Dai risultati ottenuti, si evince che il parametro *intelligibilità*, rappresentato dal valore $MOS = 4.52$ e $WER = 4.37$ %, raggiunge una valore molto buono, dimostrando una completa comprensione del testo sintetizzato, mentre la *naturalezza*, sebbene come primo approccio presenti un valore già incoraggiante, necessita di ulteriori miglioramenti. Le soluzioni atte a migliorare questo parametro sono sicuramente:

- aumentare il materiale di training (come già detto il materiale di training utilizzato consiste di sole 2 ore di parlato),
- migliorare il tuning del pitch con algoritmi di modulazione del pitch, utili anche al fine di permettere al tool di sintesi di realizzare una sintesi emozionale.

Conclusioni Il metodo descritto realizza un nuovo framework basato su tecnica "ibrida" HMM/unit-selection per la sintesi vocale, che ha la particolarità di adottare la rappresentazione MDCT, invece delle classica parametrizzazione MFCCs, al fine di superare il limite di non invertibilità della trasformazione MFCC e garantire la perfetta ricostruzione del segnale dal vettore di features. I test condotti, oggettivi e soggettivi, mostrano la bontà dell'approccio proposto.

5.5.6 Sviluppi futuri

5.5.6.1 Rimozione delle discontinuità del pitch

Dai risultati ottenuti, come già detto, si evince che il parametro da migliorare è sicuramente quello della *naturalezza*. Il problema si può ricondurre principalmente alla presenza di discontinuità del pitch. Queste discontinuità sono legate in parte al modello acustico realizzato su un materiale limitato, che non contiene quindi tutte le realizzazioni necessarie per uniformare il pitch. Una soluzione semplice e immediata è quindi quella di aumentare il database di training. Una soluzione invece più complessa, ma anche più completa, perché permette di essere utilizzata ogni volta che si verifica una discontinuità (anche nel caso di ampio database) e permette di realizzare anche una sintesi emozionale, è quella di applicare al tool, algoritmi di *pitch-shifting*. Questo tipo di algoritmo, che può operare nel dominio del tempo o della frequenza in diverse modalità [186, 187, 188, 189, 190, 191, 192], permette di "shiftare" appunto il pitch al

valore desiderato. Deve essere utilizzato con estrema attenzione, perché modificare il pitch, può portare a cambiamenti anche drastici del segnale vocale, ma applicato a segmenti molto brevi e con uno spostamento in un range limitato di frequenza permetterebbe di allineare il pitch "indesiderato" al valore del pitch "desiderato", in maniera tale da collegare in modo più armonico le unità fonetiche che andranno a costituire la parola. L'idea che si sta sviluppando è quella di estrarre il pitch dal segnale tramite la tecnica sviluppata nel lavoro [193] basata su deconvoluzione omomorfica, per poi "stretchare" il pitch in base alle tecniche note di pitch shifting e ricostruire il segnale privo del pitch con il pitch desiderato. Attualmente, la tecnica proposta in [193] è stata applicata con successo per l'estrazione del segnale privo del pitch, come mostrato in Fig. 5.15 e si sta lavorando sulla ricostruzione del segnale con il pitch opportuno.

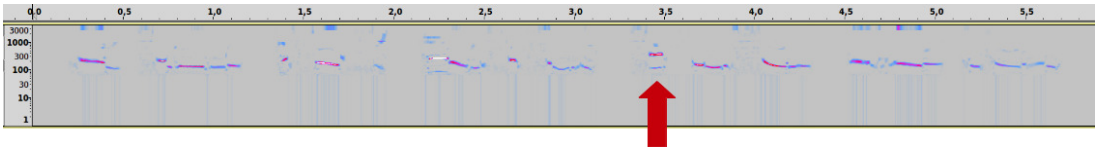


Figura 5.14: Pitch mismatch nella frase "Mia sorella aspetta sotto al sole il mio ritorno." in corrispondenza del segmento "al".

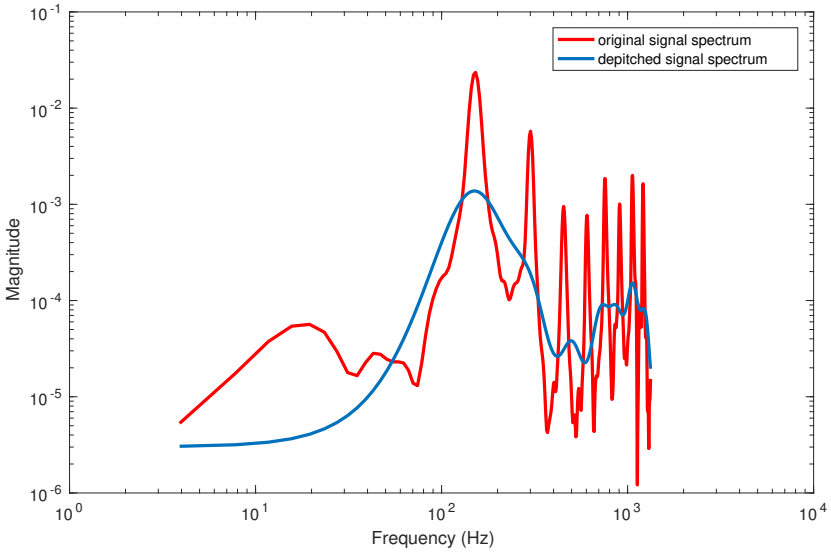


Figura 5.15: Deconvoluzione omomorfica applicata al fonema $|a|$.

Capitolo 6

Conclusioni

In questo lavoro di tesi sono state prese in analisi le diverse tecniche di interazione vocale: riconoscimento vocale (speech recognition), identificazione del parlatore (speaker recognition) e sintesi vocale (speech synthesis). Ciascuna di esse, o tutte, possono essere utilizzate in svariati ambiti di applicazione. In questo lavoro, il focus applicativo è stato l'ambito domotico.

Data la forte richiesta in ambito domotico di interfacce che abbiamo la proprietà di essere le più semplici e meno invasive possibili, e dato che la voce risulta certamente una modalità di interazione estremamente naturale per l'uomo; la scelta di utilizzare l'interazione vocale come input di un sistema domotico risulta certamente la più vantaggiosa. Sono state quindi analizzate le tre diverse modalità di interazione al fine di implementare frameworks di riconoscimento, identificazione e sintesi vocale che potessero essere integrati in un sistema domotico.

In particolare, per quanto riguarda l'ambito dello *Speech Recognition*, la ricerca si è focalizzata sullo studio delle tecniche e standard esistenti riguardanti i sistemi DSR e sull'implementazione di un sistema di riconoscimento vocale distribuito sia a livello di infrastruttura stabile e robusta sia a livello di improvement del rate del parlato continuo anche tramite l'ausilio di modelli "garbage" che permettono di distinguere i comandi dal parlato continuo. Un sistema di questo tipo, come già detto, deve poter essere facilmente accessibile. Per il raggiungimento di questo obiettivo, sono state implementate alcune interfacce ad interazione vocale lato client: una Web app che effettuata l'adattamento ed il riconoscimento on-line del parlato ed un'interfaccia Android che interagisce con un sistema wireless di controllo digitale della luce, per il controllo vocale dei punti luce.

Un requisito essenziale di una interfaccia vocale per l'home automation è rappresentata dalla capacità di riconoscere più parlatori. In prima analisi, l'attività

di ricerca si è concentrata sull'applicazione di algoritmi noti di adaptation lato back-end per adattare il modello acustico a qualsiasi parlatore. Successivamente l'attività di ricerca si è spostata verso l'implementazione di tecniche di compensazione del mismatch tra parlatori lato front-end, meno onerose dal punto di vista computazionale. I risultati ottenuti, purtroppo, non hanno portato ad un incremento significativo del rate di riconoscimento rispetto alle attuali tecniche lato back-end. Sviluppi futuri relativamente alla tecnica proposta sono orientati all'applicazione dell'algoritmo agli stati dei singoli fonemi anziché all'intero segnale del parlatore.

Un altro problema affrontato nell'ambito dello speech recognition è legato all'utilizzo del modello linguistico per l'estrazione di parole dal flusso di fonemi. L'utilizzo dei modelli linguistici richiede una larga quantità di materiale di training. Numerose tecniche sono state proposte in letteratura per risolvere questo problema; tuttavia nessuna di queste tecniche è stata inserita negli attuali sistemi LVCSR, in quanto devono essere integrate direttamente in fase di decodifica, al fine di sfruttare completamente le loro potenzialità. In questo ambito è stata studiata e implementata una tecnica "dictionary-based LVCSR", nella quale il task del decoder è quello di determinare la sequenza più probabile di fonemi, anziché ricercare la sequenza di parole che massimizza il prodotto di modelli acustici e linguistici, come di solito avviene negli attuali sistemi ASR, e trasformare la sequenza di fonemi in una sequenza di parole appartenenti ad un dato dizionario. I risultati sperimentali, mostrano la validità dell'approccio proposto. Sviluppi futuri si muovono nella direzione di ottimizzare l'algoritmo al fine di rimuovere l'offset del 5% nei risultati ottenuti. Il primo passo consiste nel concatenare più di due decoding windows, per incrementare la word recognition performance. Il secondo passo potrebbe essere quello di usare un criterio basato su soglia adattativa al fine di adattare la capacità di correzione dell'algoritmo all'errore stimato sull'input stream.

Per quanto riguarda invece, il campo della *Speaker Indentification*, la ricerca è stata orientata allo studio e sviluppo di nuove tecniche di identificazione del parlatore. Le tradizionali features MFCCs, sono state sostituite da una rappresentazione KLT troncata con classificazione tramite algoritmo di Figueiredo. I test sperimentali hanno dimostrato che su sequenze brevi di speech frames (utterance < 3.5 s), le performance della rappresentazione truncated KLT sono sempre migliori di MFCC in termini di classification accuracy, sia su un database di po-

chi parlatori (in lingua italiana) che sul più ampio TIMIT database, del quale sono stati presi in considerazione 100 speakers (in lingua inglese). Dimostrata la validità dell'approccio proposto, l'algoritmo è stato testato anche in condizioni di multiple speakers con overlap ed integrato nel sistema DSR realizzato, per ottenere un sistema combinato di speaker identification/speech recognition per la personalizzazione dell'ambiente domestico.

Infine, si è presa in analisi la modalità di interazione opposta al riconoscimento vocale, ovvero la *Speech Synthesis*. Sebbene siano state proposte in letteratura diverse tecniche per la sintesi di segnali vocali, l'approccio basato sui modelli HMM si è dimostrato tra i più vantaggiosi; ciò nonostante un problema ancora aperto è rappresentato dalla ricostruibilità del segnale con i modelli HMM. In questo ambito è stata implementata una tecnica di sintesi basata su una rappresentazione del segnale mediante features MDCT in grado di assicurare la perfetta ricostruzione del segnale e quindi superare i limiti imposti dal modello HMM e dall'utilizzo delle convenzionali features MFCCs. In particolare, il metodo proposto combina i vantaggi del modeling HMM, alla tecnica unit-selection per la concatenazione delle unità fonetica modellate. Le unità fonetiche scelte sono i trifoni context-dependent, che con la loro proprietà di dipendenza dal contesto permettono di mitigare il problema delle discontinuità dovuto alla concatenazione stessa. Test sperimentali hanno validato la bontà dell'approccio, soprattutto in termini di intelligibilità dell'audio sintetizzato. Sviluppi futuri sono orientati all'implementazione di algoritmi di pitch-shifting che permettano di perfezionare la naturalezza dell'audio sintetizzato ed all'implementazione del framework real-time proposto su dispositivi embedded.

Bibliografia

[1] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes—present state and future challenges," *Computer methods and programs in biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

[2] K.-A. Lee, A. Larcher, H. Thai, B. Ma, and H. Li, "Joint application of speech and speaker recognition for automation and security in smart home." in *INTERSPEECH*, 2011, pp. 3317–3318.

[3] S. B. Sangeetha *et al.*, "Intelligent interface based speech recognition for home automation using android application," in *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on*. IEEE, 2015, pp. 1–11.

[4] M. Vacher, F. Portet, A. Fleury, and N. Noury, "Challenges in the processing of audio channels for ambient assisted living," in *e-Health Networking Applications and Services (Healthcom'10), 2010 12th IEEE International Conference on*. IEEE, 2010, pp. 330–337.

[5] M. Vacher, B. Lecouteux, D. Istrate, T. Joubert, F. Portet, M. Sehili, and P. Chahuara, "Experimental evaluation of speech recognition technologies for voice-based home automation control in a smart home," in *4th workshop on Speech and Language Processing for Assistive Technologies*, 2013, pp. 99–105.

[6] M. Wölfel and J. McDonough, *Distant Speech Recognition*. John Wiley & Sons, Ltd, Chichester, UK, 2009.

[7] P. A. Naylor and N. D. Gaubitch, *Speech dereverberation*. Springer Science & Business Media, 2010.

[8] D. Anastasiou, "Survey on speech, machine translation and gestures in ambient assisted living," in *Tralogy, Session 4 - Tools for translators*, Mar. 2012.

[9] J.-C. Wang, H.-P. Lee, J.-F. Wang, and C.-B. Lin, "Robust environmental sound recognition for home automation," *IEEE transactions on automation science and engineering*, vol. 5, no. 1, pp. 25–31, 2008.

[10] M. Ferras, C.-C. Leung, C. Barras, and J.-L. Gauvain, "Comparison of speaker adaptation methods as feature extraction for SVM-based speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1366–1378, 2010.

[11] W.-L. Zhang, W.-Q. Zhang, B.-C. Li, D. Qu, and M. T. Johnson, "Bayesian speaker adaptation based on a new hierarchical probabilistic model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2002–2015, 2012.

[12] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, Aug 2000.

[13] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 50–58, 2008.

[14] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[15] Z.-H. Ling and L.-R. Dai, "Minimum kullback–leibler divergence parameter generation for HMM-based speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1492–1502, 2012.

[16] F. Portet, M. Vacher, C. Golanski, C. Roux, and B. Meillon, "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects," *Personal and Ubiquitous Computing*, vol. 17, no. 1, pp. 127–144, 2013.

[17] M. Vacher, B. Lecouteux, J. S. Romero, M. Ajili, F. Portet, and S. Rossato, "Speech and speaker recognition for home automation: Preliminary results," in *Speech Technology and Human-Computer Dialogue (SpeD), 2015 International Conference on.* IEEE, 2015, pp. 1–10.

[18] C. Chen and D. J. Cook, "Behavior-based home energy prediction," in *Intelligent Environments (IE), 2012 8th International Conference on.* IEEE, 2012, pp. 57–63.

[19] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: Current status and lessons learned," *Technology and Health Care*, vol. 17, no. 3, pp. 183–201, 2009.

[20] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.

[21] The SWEET-HOME project. [Online]. Available: http://sweet-home-data.imag.fr/

[22] B. Lecouteux, M. Vacher, and F. Portet, "Distant speech recognition in a smart home: Comparison of several multisource ASRs in realistic conditions," in *Interspeech 2011 Florence*, 2011, pp. 2273–2276.

[23] M. Vacher, P. Chahuara, B. Lecouteux, D. Istrate, F. Portet, T. Joubert, M. Sehili, B. Meillon, N. Bonnefond *et al.*, "The SWEET-HOME project: Audio technology in smart homes to improve well-being and reliance," in *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13),.* IEEE, 2013, pp. 7298–7301.

[24] M. Vacher, B. Lecouteux, P. Chahuara, F. Portet, B. Meillon, and N. Bonnefond, "The SWEET-HOME speech and multimodal corpus for home automation interaction," in *The 9th edition of the Language Resources and Evaluation Conference (LREC)*, 2014, pp. 4499–4506.

[25] S. Bouakaz, M. Vacher, M.-E. B. Chaumon, F. Aman, S. Bekkadja, F. Portet, E. Guillou, S. Rossato, E. Desserée, P. Traineau *et al.*, "Cirdo: Smart

companion for helping elderly to live at home for longer," *IRBM*, vol. 35, no. 2, pp. 100–108, 2014.

[26] M. Vacher, S. Bouakaz, M.-E. Bobillier-Chaumon, F. Aman, R. A. Khan, S. Bekkadja, F. Portet, E. Guillou, S. Rossato, and B. Lecouteux, "The cirdo corpus: Comprehensive audio/video database of domestic falls of elderly people," in *10th International Conference on Language Resources and Evaluation (LREC 2016)*. ELRA, 2016, pp. 1389–1396.

[27] M. Hamill, V. Young, J. Boger, and A. Mihailidis, "Development of an automated speech recognition interface for personal emergency response systems," *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 1, p. 26, 2009.

[28] G. De Pauw, W. Daelemans, J. Huyghe, J. Derboven, L. Vuegen, B. Van Den Broeck, P. Karsmakers, and B. Vanrumste, "Self-taught assistive vocal interfaces: An overview of the aladin project," 2013.

[29] H. Christensen, I. Casanueva, S. Cunningham, P. Green, and T. Hain, "homeservice: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition," in *4th Workshop on Speech and Language Processing for Assistive Technologies*, 2013, pp. 29–34.

[30] A. König, C. F. Crispim Junior, A. Derreumaux, G. Bensadoun, P.-D. Petit, F. Bremond, R. David, F. Verhey, P. Aalten, and P. Robert, "Validation of an automatic video monitoring system for the detection of instrumental activities of daily living in dementia patients," *Journal of Alzheimer's Disease*, vol. 44, no. 2, pp. 675–685, 2015.

[31] L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmüller, and P. Maragos, "The DIRHA simulated corpus." in *LREC*, 2014, pp. 2629–2634.

[32] M. Vacher, S. Caffiau, F. Portet, B. Meillon, C. Roux, E. Elias, B. Lecouteux, and P. Chahuara, "Evaluation of a context-aware voice interface for ambient assisted living: qualitative user study vs. quantitative system evaluation," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 7, no. 2, p. 5, 2015.

[33] Amazon Echo. [Online]. Available: https://www.amazon.com/dp/B00X4WHP5E

[34] Google Home. [Online]. Available: https://madeby.google.com/home/

[35] G. Linares, P. Nocéra, D. Massonie, and D. Matrouf, "The lia speech recognition system: from 10xrt to 1xrt," in *International Conference on Text, Speech and Dialogue*. Springer, 2007, pp. 302–308.

[36] P. Chahuara, F. Portet, and M. Vacher, "Making context aware decision from uncertain information in a smart home: A markov logic network approach," in *International Joint Conference on Ambient Intelligence*. Springer, 2013, pp. 78–93.

[37] M. Gallissot, J. Caelen, F. Jambon, and B. Meillon, "Une plate-forme usage pour l'intégration de l'informatique ambiante dans l'habitat: Domus," *Technique et Science Informatiques (TSI)*, vol. 32, no. 5, pp. 547–574, 2013.

[38] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall, Apr. 1993, vol. 14.

[39] R. Stuckless, "Developments in real-time speech-to-text communication for people with impaired hearing," *Communication access for people with hearing loss*, pp. 197–226, 1994.

[40] G. B. Varile and A. Zampolli, *Survey of the state of the art in human language technology*. Cambridge University Press, 1997, vol. 13.

[41] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[42] J. Makhoul and R. Schwartz, "State of the art in continuous speech recognition," *Proceedings of the National Academy of Sciences*, vol. 92, no. 22, pp. 9956–9963, 1995.

[43] G. Saon and J.-T. Chien, "Large-vocabulary continuous speech recognition systems: A look at some recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 18–33, Nov 2012.

[44] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[45] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*, ETSI ES 201 108, 2000.

[46] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms*, ETSI ES 202 050 V1.1.5, Jan. 2007.

[47] *Speech Processing, Trasmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back end speech reconstruction algorithm*, ETSI ES 202 212, 2002.

[48] K. Samudravijaya and M. Barot, "A comparison of public-domain software tools for speech recognition," in *Workshop on spoken language processing*, 2003.

[49] K.-F. Lee, H.-W. Hon, and R. Reddy, "An overview of the sphinx speech recognition system," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 35–45, 1990.

[50] X. Huang, F. Alleva, M.-Y. Hwang, and R. Rosenfeld, "An overview of the sphinx-ii speech recognition system," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1993, pp. 81–86.

[51] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4: A flexible open source framework for speech recognition," Sun Microsystems Inc., Tech. Rep. SMLI TR2004-0811, 2004.

[52] M. Seltzer and R. Singh. Instructions for using the Sphinx3 trainer. [Online]. Available: http://www.speech.cs.cmu.edu/sphinxman/fr4.html

[53] G. Biagetti, P. Crippa, A. Curzi, L. Falaschetti, S. Orcioni, and C. Turchetti, "Distributed speech recognition for lighting system control," in *Intelligent Decision Technologies*. Springer, 2015, pp. 101–111.

[54] W. Zhang, L. He, Y.-L. Chow, R. Yang, and Y. Su, "The study on distributed speech recognition system," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 2000, pp. 1431–1434.

[55] M. Alessandrini, G. Biagetti, A. Curzi, and C. Turchetti, "A speech interaction system for an ambient assisted living scenario," in *Ambient Assisted Living*. Springer, 2014, pp. 233–239.

[56] S. Hirota, N. Hayasaka, and Y. Iiguni, "Experimental evaluation of structure of garbage model generated from in-vocabulary words," in *Proc. of the 2012 International Symposium on Communications and Information Technologies (ISCIT 2012)*, Gold Coast, Australia, Oct. 2012, pp. 87–92.

[57] M. Levit, S. Chang, and B. Buntschuh, "Garbage modeling with decoys for a sequential recognition scenario," in *Proc. of the 11th IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU 2009)*, Merano, Italy, Dec. 2009, pp. 468–473.

[58] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," in *AVIOS 2000: The Speech Applications Conference*, 2000, pp. 261–264.

[59] M. Mercuri, "Sviluppo di un sistema di riconoscimento vocale distribuito," Ph.D. dissertation, Università Politecnica delle Marche, 2011/2012. [Online]. Available: http://openarchive.univpm.it/jspui/handle/123456789/913

[60] M. Alessandrini, G. Biagetti, A. Curzi, and C. Turchetti, "Semi-automatic acoustic model generation from large unsynchronized audio and text chunks," in *12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*, Florence, Italy, aug 2011, pp. 1681–1684.

[61] J. Picone, "Continuous speech recognition using hidden Markov models," *ASSP Magazine, IEEE*, vol. 7, no. 3, pp. 26–41, 1990.

[62] I. Bazzi and J. R. Glass, "Modeling out-of-vocabulary words for robust speech recognition," in *Proc. of the 6th International Conference on Spoken Language Processing (ICSLP 2000 / INTERSPEECH 2000)*, Beijing, China, Oct. 2000, pp. 401–404.

[63] M. Alessandrini, G. Biagetti, A. Curzi, and C. Turchetti, "A garbage model generation technique for embedded speech recognisers," in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013*, 2013, pp. 318–322.

[64] C.-H. Lee, C.-H. Lin, and B.-H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden markov models," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 806–814, 1991.

[65] C. J. Leggetter and P. Woodland, *Speaker adaptation of HMMs using linear regression*. University of Cambridge, Department of Engineering, 1994.

[66] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[67] Liber Liber. [Online]. Available: http://www.liberliber.it/

[68] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A.Senior, V. Vanhoucke, P. Nguyen, T. N. S. B., and Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[69] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, Jan 2012.

[70] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[71] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, May 2015.

[72] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, Sept 2015.

[73] Y. Ma, P. Niyogi, G. Sapiro, and R. Vidal, "Dimensionality reduction via subspace and submanifold learning [from the guest editors]," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 14–126, March 2011.

[74] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–393, 1999.

[75] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[76] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4, pp. 403–434, 2001.

[77] M. Siu and M. Ostendorf, "Variable n-grams and extensions for conversational speech language modeling," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 63–75, Jan 2000.

[78] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Computer Speech & Language*, vol. 20, no. 4, pp. 589–608, 2006.

[79] B. Hutchinson, M. Ostendorf, and M. Fazel, "A sparse plus low-rank exponential language model for limited resource scenarios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 494–504, 2015.

[80] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "An algorithm for automatic words extraction from a stream of phones in dictionary-based large vocabulary continuous speech recognition systems," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2015, pp. 18–23.

[81] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals." *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, feb 1966.

[82] eSpeak text to speech. [Online] Available: http://espeak.sourgeforge.net.

[83] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[84] S. Bhardwaj, S. Srivastava, M. Hanmandlu, and J. R. P. Gupta, "GFM-based methods for speaker identification," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1047–1058, June 2013.

[85] V. R. Apsingekar and P. L. De Leon, "Speaker model clustering for efficient speaker identification in large population applications," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 848–853, May 2009.

[86] L. Schmidt, M. Sharifi, and I. Lopez Moreno, "Large-scale speaker identification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 1650–1654.

[87] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23–61, Secondquarter 2011.

[88] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, Jan. 2010.

[89] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE International Conference on Acoustics, Speech, and*

Signal Processing (ICASSP), vol. 4, Orlando, FL, USA, May 2002, pp. IV–4072–IV–4075.

[90] H. Gish and M. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 18–32, Oct. 1994.

[91] J. P. Campbell, Jr., "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep. 1997.

[92] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 4, pp. 430–451, 2004.

[93] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[94] K. Chen, L. Wang, and H. Chi, "Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 03, pp. 417–445, May 1997.

[95] J. Ming, D. Stewart, P. Hanna, P. Corr, F. J. Smith, and S. Vaseghi, "Robust speaker identification using posterior union models," in *8th European Conference on Speech Communication and Technology (EUROSPEECH 2003 - INTERSPEECH 2003)*, Geneva, Switzerland, Sept. 2003, pp. 2645–2648.

[96] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.

[97] O. Thyes, R. Kuhn, P. Nguyen, and J.-C. Junqua, "Speaker identification and verification using eigenvoices," in *Sixth International Conference on Spoken Language Processing (ICSLP)*, vol. 2, Beijing, China, Oct. 2000, pp. 242–245.

[98] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in *Neural Networks for Signal Processing X — Proc. 2000 IEEE Signal Processing Society Workshop*, vol. 2, Sydney, NSW, Australia, Dec. 2000, pp. 775–784.

[99] R. P. Ramachandran, M. S. Zilovic, and R. J. Mammone, "A comparative study of robust linear predictive analysis methods with applications to speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 2, pp. 117–125, Mar. 1995.

[100] K. T. Assaleh and R. J. Mammone, "New LP-derived features for speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 630–638, Oct. 1994.

[101] M. S. Zilovic, R. P. Ramachandran, and R. J. Mammone, "Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 260–267, May 1998.

[102] V. Chandran, D. Ning, and S. Sridharan, "Speaker identification using higher order spectral phase features and their effectiveness vis–a–vis Mel-cepstral features," in *Biometric Authentication: First International Conference, ICBA 2004, Hong Kong, China, July 15-17, 2004. Proceedings*, D. Zhang and A. K. Jain, Eds. Berlin, Heidelberg: Springer, 2004, pp. 614–622.

[103] S. Furui, "50 years of progress in speech and speaker recognition," *SPECOM 2005, Patras*, pp. 1–9, 2005.

[104] D. A. Reynolds, "Experimental evaluation of features for robust speaker identification," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 639–643, Oct. 1994.

[105] A. P. Dobrowolski and E. Majda, "Cepstral analysis in the speakers recognition systems," in *Proc. Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA)*, Poznan, Poland, Sept. 2011, pp. 1–6.

[106] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[107] Q. Jin, A. R. Toth, A. W. Black, and T. Schultz, "Is voice transformation a threat to speaker identification?" in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2008)*, Las Vegas, NV, USA, Mar. 2008, pp. 4845–4848.

[108] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[109] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 210–229, Apr./July 2006.

[110] U. V. Chaudhari, J. Navrratil, G. N. Ramaswamy, and S. H. Maes, "Very large population text-independent speaker identification using transformation enhanced multi-grained models," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '01)*, vol. 1, Salt Lake City, UT, USA, May 2001, pp. 461–464 vol.1.

[111] M. Hossain, B. Ahmed, and M. Asrafi, "A real time speaker identification using artificial neural network," in *10th International Conference on Computer and Information Technology (ICCIT 2007)*, Dhaka, Bangladesh, Dec. 2007, pp. 1–5.

[112] C. W. Maina and J. M. Walsh, "Joint speech enhancement and speaker identification using approximate Bayesian inference," in *44th Annual Conference on Information Sciences and Systems (CISS 2010)*, Princeton, NJ, USA, Mar. 2010, pp. 1–6.

[113] L. Xu and Z. Yang, "Speaker identification based on sparse subspace model," in *19th Asia-Pacific Conference on Communications (APCC 2013)*, Denpasar, Indonesia, Aug. 2013, pp. 37–41.

[114] X. Zhao, Y. Shao, and D. Wang, "CASA-based robust speaker identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1608–1616, July 2012.

[115] H. Zeinali, H. Sameti, and B. BabaAli, "A fast speaker identification method using nearest neighbor distance," in *IEEE 11th International Conference on Signal Processing (ICSP 2012)*, vol. 3, Beijing, China, Oct. 2012, pp. 2159–2162.

[116] M. E. M. Weddin, "Speaker identification for hearing instruments," Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2005.

[117] S. Nakagawa, L. Wang, and S. Ohtsuka, "Speaker identification and verification by combining MFCC and phase information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1085–1095, May 2012.

[118] R. D. Zilca, B. Kingsbury, J. Navratil, and G. N. Ramaswamy, "Pseudo pitch synchronous analysis of speech with applications to speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 467–478, Mar. 2006.

[119] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[120] Y. Hu, D. Wu, and A. Nucci, "Fuzzy-clustering-based decision tree approach for large population speaker identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 762–774, Apr. 2013.

[121] J. Luque and J. Hernando, "Robust speaker identification for meetings: UPC CLEAR'07 meeting room evaluation system," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science, R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Springer Berlin Heidelberg, 2008, vol. 4625, pp. 266–275.

[122] G. Friedland and O. Vinyals, "Live speaker identification in conversations," in *Proc. 16th ACM International Conference on Multimedia*. Vancouver, BC, Canada: ACM, Oct. 2008, pp. 1017–1018.

[123] G. Biagetti, P. Crippa, A. Curzi, S. Orcioni, and C. Turchetti, "Speaker identification with short sequences of speech frames," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM 2015)*, Lisbon, Portugal, 2015, pp. 178–185.

[124] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "An investigation on the accuracy of truncated DKLT representation for speaker identification with short sequences of speech frames," *IEEE Transactions on Cybernetics*, 2016.

[125] M. A. F. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[126] N. Singh-Miller, M. Collins, and T. J. Hazen, "Dimensionality reduction for speech recognition using neighborhood components analysis," in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, Antwerp, Belgium, Aug. 2007, pp. 1158–1161.

[127] X. Jiang, "Linear subspace learning-based dimensionality reduction," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 16–26, Mar. 2011.

[128] I. T. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. Springer New York, 1986.

[129] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, Sept. 1933.

[130] C. W. Therrien, *Discrete Random Signals and Statistical Signal Processing*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1992.

[131] K. Fukunaga, *Introduction to statistical pattern recognition*, 2nd ed. San Diego, CA, USA: Academic Press, 1990.

[132] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[133] S. H. Yella and H. Bourlard, "Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1688–1700, 2014.

[134] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Robust speaker identification in a meeting with short audio segments," in *Intelligent Decision Technologies 2016*. Springer, 2016, pp. 465–477.

[135] NIST. (2000) 2000 speaker recognition evaluation - evaluation plan. [Online]. [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/spk/2000/spk-2000-plan-v1.0.htm

[136] O. Galibert, "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech." in *Proc. INTERSPEECH*, 2013, pp. 1131–1134.

[137] Ami meeting corpus. [Online]. Available: http://www.idiap.ch/dataset/ami/

[138] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, *The AMI meeting corpus: A pre-announcement*. Springer, 2005.

[139] G. Biagetti, P. Crippa, A. Curzi, L. Falaschetti, S. Orcioni, and C. Turchetti, "A distributed speaker identification system for personalized home control," in *Workshop on Mobile Networks for Biometric Data Analysis (mBiDA), 2014*, 2014.

[140] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Distributed speech and speaker identification system for personalized domotic control," in *Mobile Networks for Biometric Data Analysis*. Springer, 2016, pp. 159–170.

[141] J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni, *From text to speech: The MITalk system.* Cambridge University Press, 1987.

[142] S. Takamichi, T. Toda, Y. Shiga, H. Kawai, S. Sakti, and S. Nakamura, "An evaluation of parameter generation methods with rich context models in HMM-based speech synthesis." in *INTERSPEECH*, 2012, pp. 1139–1142.

[143] S. Lemmetty, "History and development of speech synthesis," *Retrieved from*, 2004.

[144] W. Kempelen, H. Füger, and J. G. Mansfeld, *Wolfgangs von Kempelen kk wirklichen Hofraths Mechanismus der menschlichen Sprache: nebst der Beschreibung seiner sprechenden Maschine: mit XXVII Kupertafeln.* Bei JB Degen, 1791.

[145] I. G. Mattingly, "Speech synthesis for phonetic and phonological models," *Haskins Labs. Stat. Rep. Speech*, vol. 23, pp. 117–149, 1970.

[146] K. Fukui, K. Nishikawa, S. Ikeo, M. Honda, and A. Takanishi, "Development of a human-like sensory feedback mechanism for an anthropomorphic talking robot," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 101–106.

[147] K. Fukui, Y. Ishikawa, E. Shintaku, K. Ohno, N. Sakakibara, A. Takanishi, and M. Honda, "Vocal cord model to control various voices for anthropomorphic talking robot," in *Proceedings of the 8th International Speech Production Seminar (ISSP), Straßburg*, 2008, pp. 341–344.

[148] J. P. Van Santen, R. Sproat, J. Olive, and J. Hirschberg, *Progress in speech synthesis.* Springer Science & Business Media, 2013.

[149] T. Dutoit, "High-quality text-to-speech synthesis: An overview," *Journal of Electrical and Electronics Engineering Australia*, vol. 17, pp. 25–36, 1997.

[150] D. H. Klatt, "Interaction between two factors that influence vowel duration," *The Journal of the Acoustical Society of America*, vol. 54, no. 4, pp. 1102–1104, 1973.

[151] ——, "Review of text-to-speech conversion for english," *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[152] K. Hirose, H. Fujisaki, and M. Yamaguchi, "Synthesis by rule of voice fundamental frequency contours of spoken japanese from linguistic information," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9. IEEE, 1984, pp. 597–600.

[153] J. Tao, K. Hirose, K. Tokuda, A. W. Black, and S. King, "Introduction to the issue on statistical parametric speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 170–172, 2014.

[154] T. Masuko, "HMM-based speech synthesis and its applications," Ph.D. dissertation, Institute of Technology, Tokyo, Japan, 2002.

[155] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[156] Y. Stylianou, J. Laroche, and E. Moulines, "High-quality speech modification based on a harmonic+ noise model," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[157] D. Sündermann, A. Bonafonte, H. Ney, and H. Höge, "A study on residual prediction techniques for voice conversion," in *ICASSP (1)*, 2005, pp. 13–16.

[158] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda, "An excitation model for HMM-based speech synthesis based on residual modeling," in *Workshop Proceedings*, vol. 95, 2007, p. a3.

[159] A. W. Black and K. Tokuda, "The blizzard challenge–2005: Evaluating corpus-based speech synthesis on common datasets," in *Proceedings of interspeech*, 2005.

[160] H. Zen and T. Toda, "An overview of nitech HMM-based speech synthesis system for blizzard challenge 2005," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[161] M. Kaszczuk and L. Osowski, "Evaluating ivona speech synthesis system for blizzard challenge 2006," in *Blizzard Workshop, Pittsburgh*, 2006.

[162] C. Benoît, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences," *Speech Communication*, vol. 18, no. 4, pp. 381–392, 1996.

[163] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, May 2013.

[164] R. E. Donovan and P. C. Woodland, "A hidden Markov-model-based trainable speech synthesizer," *Computer Speech & Language*, vol. 13, no. 3, pp. 223 – 241, 1999.

[165] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *EUROSPEECH*, 1997.

[166] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, (ICASSP'00)*, vol. 3, 2000, pp. 1315–1318.

[167] T. Toda and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis," in *9th European Conf. Speech Communication and Technology*, 2005, pp. 2801–2804.

[168] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems," Ph.D. dissertation, Nagoya Institute of Technology, 2002.

[169] J. Yamagishi, T. Nose, H. Zen, Z.-H. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive HMM-based text-to-speech synthesis," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1208–1230, Aug 2009.

[170] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 153–165, Jan 2011.

[171] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1071–1079, July 2011.

[172] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2129–2139, Oct 2013.

[173] J. P. Cabral, K. Richmond, J. Yamagishi, and S. Renals, "Glottal spectral separation for speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 195–208, April 2014.

[174] H. S. Malvar, *Signal processing with lapped transforms*. Artech House, Inc., 1992.

[175] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '87*, vol. 12, Apr 1987, pp. 2161–2164.

[176] J. Princen and A. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1153–1161, Oct 1986.

[177] M. Bosi and R. E. Goldberg, *Introduction to digital audio coding and standards*. Springer, 2003.

[178] G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Learning HMM state sequences from phonemes for speech synthesis," *Procedia Computer Science*, vol. 96, pp. 1589–1596, 2016.

[179] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[180] W. Holmes, *Speech synthesis and recognition*. CRC press, 2001.

[181] F. Itakura and S. Saito, "Analysis synthesis telephony based on the maximum likelihood method," in *Proceedings of the 6th International Congress on Acoustics*, vol. 17. pp. C17–C20, 1968, pp. C17–C20.

[182] G. Chen, S. N. Koh, and I. Y. Soon, "Enhanced Itakura measure incorporating masking properties of human auditory system," *Signal Processing*, vol. 83, no. 7, pp. 1445–1456, 2003.

[183] Copying the MBROLA bin and databases. [Online] Available: http://www.tcts.fpms.ac.be/synthesis /mbrola/mbrcopybin.html.

[184] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. Van der Vrecken, "The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes," in *Proc. Fourth International Conference on Spoken Language, (ICSLP 96)*, vol. 3, Oct 1996, pp. 1393–1396 vol.3.

[185] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, no. 5–6, pp. 453 – 467, 1990.

[186] S. Kawachale, S. Gengaje, and J. Chitode, "Spectral mismatch as the index of quality of naturalness in synthetic speech," in *2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. IEEE, 2009, pp. 808–813.

[187] A. Mousa, "Voice conversion using pitch shifting algorithm by time stretching with PSOLA and re-sampling," *Journal of electrical engineering*, vol. 61, no. 1, pp. 57–61, 2010.

[188] Y. Pantazis and Y. Stylianou, "On the detection of discontinuities in concatenative speech synthesis," in *Progress in nonlinear speech processing*. Springer, 2007, pp. 89–100.

[189] D. Singh and P. Singh, "Removal of spectral discontinuity in concatenated speech waveform," *International Journal of Computer Applications*, vol. 53, no. 16, 2012.

[190] S. H. Chen, H. Liu, Y. Xu, and C. R. Larson, "Voice f0 responses to pitch-shifted voice feedback during english speech," *Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 1157–1163, 2007.

[191] Y. Stylianou and A. K. Syrdal, "Perceptual and objective detection of discontinuities in concatenative speech synthesis," in *Acoustics,*

Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, vol. 2. IEEE, 2001, pp. 837–840.

[192] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 805–808.

[193] G. Biagetti, P. Crippa, S. Orcioni, and C. Turchetti, "Homomorphic deconvolution for MUAP estimation from surface emg signals," *IEEE journal of biomedical and health informatics*, 2016.

Lista delle Pubblicazioni

G. Biagetti, P. Crippa, A. Curzi, L. Falaschetti, S. Orcioni, and C. Turchetti, "A distributed speaker identification system for personalized home control," in *Workshop on Mobile Networks for Biometric Data Analysis (mBiDA)*, 2014.

A. Bacá, G. Biagetti, M. Camilletti, P. Crippa, L. Falaschetti, S. Orcioni, L. Rossini, D. Tonelli, and C. Turchetti, "CARMA: A robust motion artifact reduction algorithm for heart rate monitoring from ppg signals," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 2646–2650.

G. Biagetti, P. Crippa, A. Curzi, L. Falaschetti, S. Orcioni, and C. Turchetti, "Distributed speech recognition for lighting system control," in *Intelligent Decision Technologies*. Springer, 2015, pp. 101–111.

G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "A rule based framework for smart training using sEMG signal," in *Intelligent Decision Technologies*. Springer, 2015, pp. 89–99.

——, "An algorithm for automatic words extraction from a stream of phones in dictionary-based large vocabulary continuous speech recognition systems," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2015, pp. 18–23.

G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, N. Ortolani, and C. Turchetti, "Improvement of RS-485 performance over long distances using the ToLHnet protocol," in *Intelligent Solutions in Embedded Systems (WISES), 2015 12th International Workshop on*. IEEE, 2015, pp. 85–89.

P. Crippa, A. Curzi, L. Falaschetti, and C. Turchetti, "Multi-class ECG beat classification based on a gaussian mixture model of Karhunen-Loève Transform." *International Journal of Simulation–Systems, Science & Technology*, vol. 16, no. 1, 2015.

G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Discrete bessel functions for representing the class of finite duration decaying sequences," in *Signal Processing Conference (EUSIPCO), 2016 24rd European*. IEEE, 2016.

M. Alessandrini, G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Optimizing linear routing in the tolhnet protocol to improve performance over long RS-485 buses," *EURASIP Journal on Embedded Systems*, vol. 2017, no. 1, p. 7, 2016.

G. Biagetti, P. Crippa, L. Falaschetti, S. Orcioni, and C. Turchetti, "Motion artifact reduction in photoplethysmography using bayesian classification for physical exercise identification," in *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, 2016, pp. 467–474.

——, "An efficient technique for real-time human activity classification using accelerometer data," in *Intelligent Decision Technologies 2016*. Springer, 2016, pp. 425–434.

——, "Multivariate direction scoring for dimensionality reduction in classification problems," in *Intelligent Decision Technologies 2016*. Springer, 2016, pp. 413–423.

——, "Robust speaker identification in a meeting with short audio segments," in *Intelligent Decision Technologies 2016*. Springer, 2016, pp. 465–477.

——, "Distributed speech and speaker identification system for personalized domotic control," in *Mobile Networks for Biometric Data Analysis*. Springer, 2016, pp. 159–170.

——, "Learning HMM state sequences from phonemes for speech synthesis," *Procedia Computer Science*, vol. 96, pp. 1589–1596, 2016.

——, "Wireless surface electromyograph and electrocardiograph system on 802.15.4," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 3, pp. 258–266, 2016.

——, "An investigation on the accuracy of truncated DKLT representation for speaker identification with short sequences of speech frames," *IEEE Transactions on Cybernetics*, 2016.