Che cosa è, come funziona:

Algoritmi e tecnologie per il riconoscimento vocale

Stato dell'arte e sviluppi futuri

Andrea **Falletto**

Rai
Centro Ricerche e
InnovazioneTecnologica
Torino

1. Agli inizi

Il Centro Ricerche e Innovazione Tecnologica è da sempre impegnato nello studio di nuove tecnologie in grado di fornire agli utenti disabili opportunità di inclusione e di fruizione dei programmi televisivi.

Le tecnologie basata sul riconoscimento vocale trovano impiego in questo ambito, per esempio facilitando la sottotitolazione automatica grazie al riconoscimento del parlato in un programma.

Questo articolo ha lo scopo di offrire una visione d'insieme dei processi coinvolti nel riconoscimento vocale tramite computer.

Anche se le maggiori innovazioni nel campo del riconoscimento vocale si sono sviluppate negli ultimi due decenni, la storia di questa tecnologia ha radici lontane.

Sommario
Le tecnologie, sviluppatesi a partire dagli anni '50, per consentire il riconoscimento vocale si sono evolute nel corso degli anni e, grazie anche all'accresciuta capacità di elaborazione dei computer, trovano oggi sempre più ampia applicazione. In particolare possono costituire un valido ausilio anche in campi di stretto interesse degli enti televisivi, ad esempio per facilitare la sottotitolazione dei programmi, ed in prospettiva per consentirla anche in modo automatico e con basso tasso di errore. L'articolo è un'introduzione per meglio comprendere la complessità e le potenzialità di queste tecniche.

Il riconoscimento vocale

Fig. 1 - Olivetti M20 – Commercializzato dalla Olivetti nel 1982 – Disponeva di un piccolo altoparlante per dare conferme acustiche e segnalare malfunzionamenti.

Fig. 2 - Un Notebook multimediale con Windows XP – I moderni software di riconoscimento vocale a dettatura continua – speaker dependent, possono funzionare in real time su una macchina come questa.

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 NSA (National Security Agency) e il dipartimento della difesa. 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 Carneige Mellor Univeristy 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 settanta i personal computer emettevano, tramite un piccolo altoparlante, alcuni suoni per dare all'operatore indicazioni sullo stato delle procedure e come feedback per indicare se la macchina stava operando correttamente. I PC non emettevano altri suoni o musica e solo alla fine degli anni ottanta comparvero le prime schede audio che permettevano la riproduzione di brevi suoni o frasi musicali (figura 1).

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 suonare, riprodurre CD e DVD ed elaborare segnali audio/video. Anche i notebook dagli anni

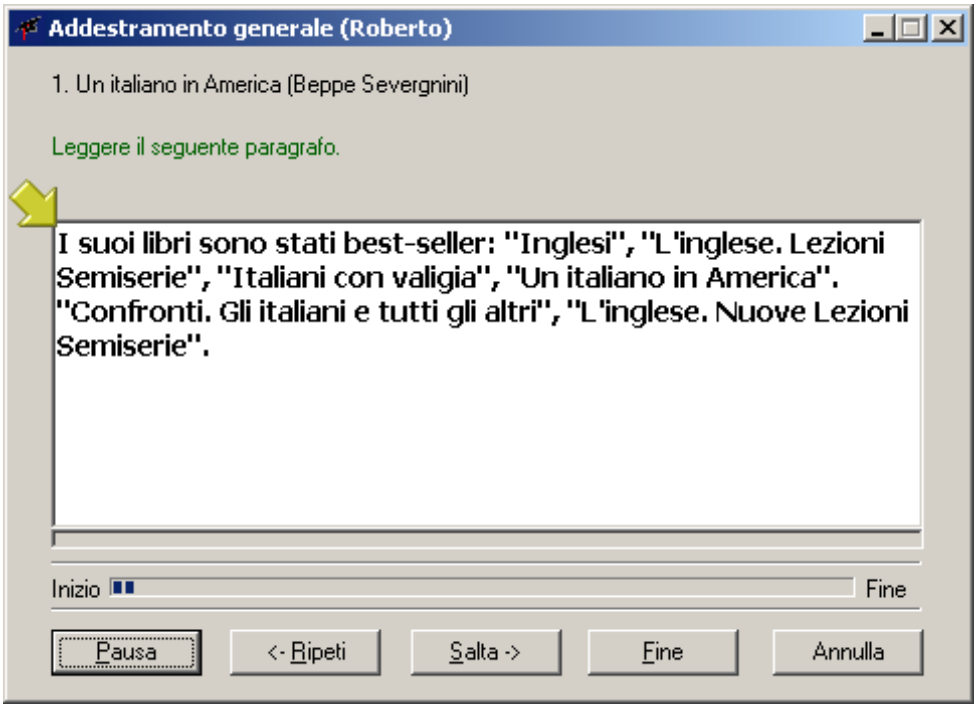

Fig. 3- Fase di addestramento (training) di un applicativo software di riconoscimento vocale: è richiesto all'utente di leggere un testo noto. In questo modo il software apprende le caratteristiche vocali e di pronuncia dell'utilizzatore.

novanta in poi diventano multimediali (figura 2), 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. Nel campo della telefonia, inoltre, i risponditori basati su riconoscitori vocali sono sempre più diffusi ed efficienti, si parla sempre più spesso di *call center* vocali automatici.

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 (riconoscimento vocale dipendente dal parlatore, *speaker dependent*, a dettatura continua). La precisione di riconoscimento di questi software è del 95 e 98 %. Il testo che state leggendo è stato scritto utilizzando un software di riconoscimento vocale.

2. Il riconoscimento vocale dipendente e indipendente dal parlatore

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 (figura 3). Il sistema si adatta così alle caratteristiche della voce e della pronuncia dell'utilizzatore. Questi sistemi offrono i migliori risultati in termini di precisione permettendo anche, dopo un po' di pratica, di correggere gli errori di interpretazione tramite il microfono, usando solo le funzioni vocali. Gli algoritmi su cui sono basati, prevedono che venga tenuta traccia delle correzioni, per consentire di imparare dagli errori.
- *Speaker independent*: permettono il riconoscimento di un parlato generico, senza es-

sere legati ad un determinato timbro di voce. La precisione di questi sistemi è inferiore rispetto a quelli dipendenti dal parlatore. La loro applicazione principale si individua nei servizi di informazione automatici, in cui, ad esempio, tramite il telefono ci viene chiesto di dire il nome della città da cui intendiamo partire

Risponditore: *"Dica solo il nome della città da cui si desidera partire"*

Utente: *"Ancona"*

Risponditore: *"Lei ha detto" - "Ancona"- ("?")- "dica si o no"*

Utente: *"si"*

Ogni individuo ha un proprio timbro vocale e un modo diverso di pronunciare le parole. 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.

3. Il Database - dizionario del software di riconoscimento

Il funzionamento di un sistema di riconoscimento vocale si basa sulla comparazione dell'audio in ingresso, opportunamente elaborato, con un database creato in fase di addestramento del sistema. In pratica l'applicativo software cerca di individuare la parola pronunciata dal parlatore, cercando nel database un suono simile e verificando a che parola corrisponde. Naturalmente è una operazione molto complessa, inoltre non viene fatta sulle parole intere ma sui fonemi che le compongono (figura 4).

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.

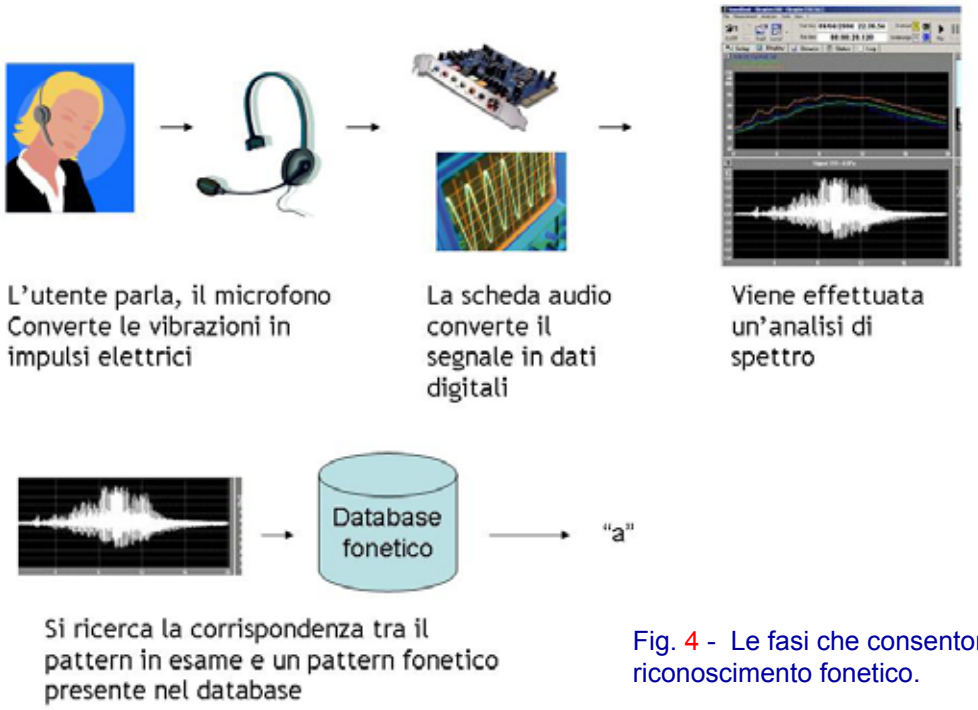

Fig. 4 - Le fasi che consentono il riconoscimento fonetico.

I sistemi *speaker independent*, come accennato, hanno una precisione inferiore perché non dispongono del modello vocale del parlatore. Per aumentare la precisione e operare su un database composto ad esempio da 200.000 lemmi, 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 di 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.

4. Come funziona il riconoscimento vocale

La materia che si vuole trattare è molto complessa. In questa sede si è scelto di affrontare l'argomento con l'ottica di spiegare solo i concetti chiave, ricorrendo ove necessario all'utilizzo di esempi.

Il riconoscimento vocale automatico è basato su una sequenza di processi che si può così riassumere:

1. Trasformazione dei dati audio dal dominio del tempo al dominio della frequenza tramite FFT (Fast Fourier Transform)
2. Organizzazione dei dati ottenuti (tramite l'applicazione delle regole e del dizionario fonetico di una lingua)
3. Riconoscimento dei singoli fonemi
4. Composizione dei fonemi in parole e applicazione di un modello linguistico caratteristico della lingua in uso

Glossario

- Suono: Una vibrazione che si propaga in un mezzo (aria o solidi) come onda
- Fonema: l'unità minima che ha un valore distintivo all'interno di una lingua (a, t, e, z). Nella lingua italiana i fonemi sono circa trenta
- Parola: Un insieme di fonemi che uniti danno luogo a un vocabolo di senso compiuto in una certa lingua

Nel dominio del tempo il segnale audio si presenta come una forma d'onda periodica di frequenza compresa tra 300 e 3000 Hz (consideriamo la banda vocale). In questo formato le informazioni non permettono di individuare un pattern correlato a quello che viene detto. E' quindi necessario effettuare una conversione dei dati dal dominio del tempo al dominio della frequenza. Viene quindi realizzata una analisi di spettro del segnale, considerando una finestra di pochi campioni per volta e applicando la trasformata di Fourier. In questo modo è possibile identificare le frequenze che compongono il suono in esame e quale ampiezza ha ogni singola componente.

La FFT viene applicata tipicamente ad un segmento di audio della durata di un centesimo di secondo, dal quale si ricava un ipotetico grafico con l'ampiezza di ogni frequenza che compone il suono. Il riconoscitore vocale ha un database costituito da molte migliaia di questi "grafici" ognuno dei quali rappresenta l'enorme quantità di suoni diversi che la voce umana può produrre.

4.1 Dai grafici che rappresentano le frequenze componenti ai fonemi

Caso ideale

Procedendo per gradi, consideriamo ciò che avverrebbe in un caso ideale, immaginando cioè

che tutti abbiano la stessa voce e strumenti di analisi audio perfetti.

Il "grafico" del suono in analisi viene confrontato con tutto il database fino a che il sistema individua quello più simile. In questo modo è possibile stabilire che si trattava, ad esempio, di una "a". In realtà il sistema dalla FFT ricava dei valori in base ai quali, per ogni centesimo di secondo, viene calcolato un *feature number*. Il *feature number* è quindi un numero che rappresenta il suono nel centesimo di secondo in esame. Anche il database contiene i grafici o *pattern* di riferimento sotto forma di numeri.

In una ipotesi ideale, ci sarebbe una corrispondenza diretta tra un *feature number* e un fonema. Quindi se il segmento di audio analizzato mostrasse come risultato il *feature number* n° 52 significherebbe che il parlatore ha pronunciato una "h". Il *feature number* n° 53 corrisponderebbe ad una "f" e così via.

Sfortunatamente nel mondo reale le cose sono più complesse perché:

- Ogni volta che una persona dice una stessa parola, la dice in modo differente: quindi non produce mai lo stesso suono per ogni fonema. Per il nostro orecchio non costituisce un ostacolo, siamo infatti perfettamente in grado di capire un amico con il raffreddore, per il computer invece non è così immediato.
- I computer non dispongono dell'ascolto intenzionale (la caratteristica per cui il nostro sistema psicoacustico ci permette di "ascoltare" gli archi anche nel pieno d'orchestra) quindi i rumori di fondo, la musica e gli altri suoni sono un elemento fortemente disturbante e possono inficiare il riconoscimento del parlato in modo imprevedibile
- Il suono di ogni fonema cambia a seconda del fonema che lo precede e che lo segue. Il suono della "t" nella parola "tavolo" è molto diverso nella parola "antenna" o "treno"
- Il suono di un fonema cambia se si trova all'inizio o alla fine di una parola, ad esempio la "a" in "astice" è molto diversa dalla "a" in "roma", quindi le due "a" producono sequenze di *feature number* molto differenti.

Tutto questo ci porta a descrivere il riconoscimento vocale nel caso reale.

Caso Reale

Iniziamo dicendo che un fonema dura molto di più di un centesimo di secondo. Evidentemente ogni fonema produce più *feature number*: se vene analizzato un centesimo di secondo per volta, significa che ogni secondo vengono prodotti 100 *feature number*. Se in un secondo viene detta la parola "mano", dei cento *feature number* calcolati una parte rappresentano la "m", una parte rappresentano la transizione dalla "m" alla "a", una parte rappresentano la "a" poi la transizione dalla "a" alla "n" ecc.)

Visiti nel dettaglio, i passi per realizzare un riconoscitore sono.

- Istruire il software su come suona un fonema nelle varie pronunce e posizioni all'interno delle parole.
  - Un tool di training processa migliaia di registrazioni diverse dello stesso fonema.
  - Il sistema analizza ogni centesimo di secondo dell'audio e calcola un *feature number* in base all'ampiezza delle frequenze componenti.
  - Il sistema memorizza quindi migliaia di *feature number* per ogni fonema.
  - Maggiore è la quantità e l'eterogeneità del materiale analizzato, maggiore sarà la capacità del software di riconoscere le parole correttamente.
- Nel contempo, durante la fase di training, il software apprende anche una serie di dati statistici. Il dato più importante è costituito da quante probabilità ci sono che un determinato fonema generi una certa sequenza di

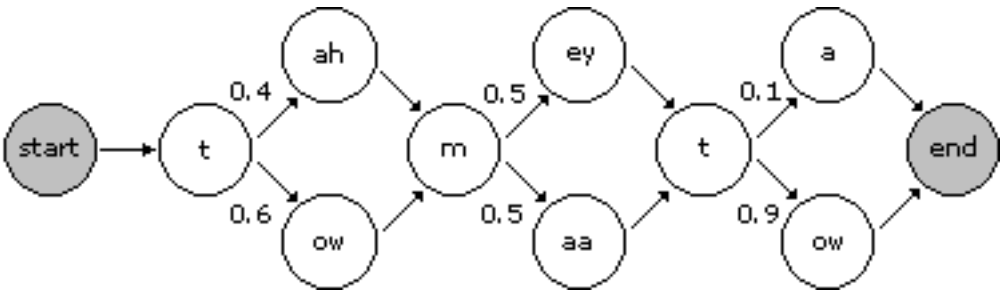

Fig. 5 - Esempio di Hidden Markov Model applicato alla pronuncia della parola inglese *Tomato*.
Questo modello prevede la pronuncia della parola in tre modi diversi
t ow m aa t ow - Inglese Britannico
t ah m ey t ow - Inglese Americano
t ah mey t a - Possibile pronuncia quando si parla rapidamente.
Si noti, dalle probabilità attribuite alle connessioni tra i fonemi, come il modello è leggermente sbilanciato verso il l'inglese britannico.

feature #. Prendiamo ad esempio il fonema "h". Il software dopo aver analizzato migliaia di "h"

- Apprende che in un centesimo di secondo del suono di una "h"
  - Ci sono il 55% di probabilità che compaia il feature #52
  - 30% di probabilità che compaia il feature #189
  - 15% di probabilità che compaia il feature #53
- Nel caso di una "f" ogni centesimo di secondo ci sono
  - 10% di probabilità che appaia un feature #52
  - 10% di probabilità che appaia il feature #189
  - 80% di probabilità che appaia il feature #53

Questa analisi delle probabilità è usata nella fase di riconoscimento; supponiamo che durante il riconoscimento i sei feature number calcolati durante sei centesimi di secondo siano:

…….52, 52 ,189 ,53 ,52 ,52…..

Il riconoscitore calcola le probabilità che sia uno dei trenta fonemi della lingua italiana.

Le probabilità che 52, 52 ,189 ,53 ,52 ,52 corrispondano ad una "h" sono

80% * 80% * 30% * 15% * 80% * 80% = 1.84 %

Le probabilità che il suono sia una "f" sono:

10% * 10% * 10% * 80% * 10% * 10 % = 0.0008 %

Il calcolo delle probabilità viene fatto per tutti i fonemi dalla "a" alla "z". Se tutti danno un risultato inferiore a 1.84%, molto probabilmente il suono analizzato è una "h".

Per mettere in pratica quanto esposto, i computer si affidano a strumenti matematici complessi. Tra i più usati a questo scopo c'è l' "Hidden Markov Model" HMM (figura 5).

In questo caso l'HMM viene usato per modellizzare una grossa matrice di fonemi, collegati tra di loro da "ponti" più o meno larghi, in base alle probabilità che un fonema sia correlato ad un altro.

I dati in input, come fossero un flusso di auto in marcia prendono preferenzialmente una strada o l'altra nella matrice a seconda di quanto i ponti tra un fonema e l'altro sono larghi e permettono il passaggio del traffico. La larghezza dei ponti viene modulata dalle statistiche calcolate dagli altri blocchi del sistema. Nel caso prima descritto il ponte che collegava il fonema precedente alla "h" era più grande (1,84) rispetto a al ponte che lo collegava alla "f" (0,0008).

4.2 La suddivisione dei fonemi

Il riconoscitore vocale ha anche bisogno di sapere quando un fonema finisce e inizia il successivo. A questo scopo vengono ancora impiegati gli "Hidden Markov Models".

E' possibile formarsi un idea di come funziona la procedura in questo esempio: supponiamo che il motore di riconoscimento individui una "h" seguita da un "ee". In base alle statistiche il fonema che corrisponde al suono "ee" ha:

- il 75% di possibilità di produrre un feature #82 per ogni centesimo di secondo,
- il 15% di possibilità di produrre un *feature* #98 e
- il 10% di dare origine a un *feature* #52.

Teniamo presente che il *feature* #53 appare anche nel fonema "h". Mettendo in fila i *feature number* di "hee" otteniamo:

……..23,52,52, 52, 189, 53, 52, 52, 82, 52, 82,82 ……..

. . . . . h . . . . ……... . . . . . .ee. . . . . .

Quindi dove finisce il suono "h" e inizia "ee" ?

Gli Hidden Markov Models

Sono modelli statistici che si possono applicare a sistemi che presentino la proprietà di Markov. Si dice che un sistema dispone della proprietà di Markov quando gli stati che può assumere in futuro dipendono solo dallo stato presente e sono indipendenti dagli stati passati.

Detto in un altro modo è un processo in cui le condizioni in un dato momento dipendono solo dalla situazione nello stato precedente e non da come si è giunti a tale stato. Come nel caso di un automobilista che decide di prendere una strada solo in base a quella che ha lasciato un momento prima, senza tener conto della mappa generale. Ciò che conta, alla fine è il punto in cui arriva e, visto a posteriori, il percorso che ha compiuto.

Hidden Markov Model hanno la caratteristica di poter determinare i parametri ignoti (hidden) in base a parametri osservabili.

Un esempio: un amico vive lontano e ama andare in moto. L'amico vi dice che un certo giorno

- andrà in moto verso la montagna,
- andrà in moto verso il mare
- oppure passerà la giornata al cinema.

La scelta di cosa farà è determinata esclusivamente dalle condizioni atmosferiche del giorno in esame. Non avete informazioni definitive sul meteo della città dove abita l'amico ma avete sentito per radio le previsioni per la sua regione. In base a cosa vi dirà che ha fatto, dovete indovinare com'era il tempo nella città del vostro amico nel giorno in questione.

Guardando i numeri appare che i 52 sono raggruppati all'inizio della sequenza gli 82 sono verso la fine. "Ad occhio" possiamo dire che la divisione si trova da qualche parte in mezzo ai due gruppi. Il computer per stabilirlo fa uso degli HMM: in questo caso ci sarà una matrice di *feature number*, i ponti avranno larghezze diverse in base alle probabilità (75% - 15% - 10%). In un primo istante il segnale di input percorrerà la strada che collega i feature number caratteristici della "h", poi i ponti saranno più agevoli nella zona della ee. Il segnale avrà effettuato dentro la matrice il percorso "hee". Le informazioni in uscita dalla matrice tengono traccia di tutto il processo e sarà possibile spere quanti millisecondi è durata la "h" e quanti millisecondi è durata la "ee")

4.3 Il silenzio

Il software sfrutta anche le pause del parlato per identificare l'inizio di un nuovo fonema, inoltre si occupa di analizzare in fonema "silenzio". In realtà le pause del parlato vengono sfruttate per prendere dei "campioni" del rumore di fondo e del fruscio che generano pattern di *feature number,* esattamente come il resto dell'audio.

Queste sequenze di dati vengono usate dal software per "depurare" l'informazione sonora utile dal pattern di rumore.

4.4 Uniamo i fonemi insieme, formando delle parole

Per l'algoritmo di riconoscimento tutti i processi descritti sopra rimangono nella sfera del possibile: tutto può essere rimesso in discussione e ri-calcolato fino a quando non si prova a mettere i fonemi insieme per formare le parole.

Il suono di un fonema cambia in relazione a quello che viene detto prima ed a quello che viene detto dopo. La "a" di "Aldo" "suona" diversamente da quella di "Andrea". Nella A di Aldo c'è una parte di "l" e nella A di "Andrea" c'è un po' di "n".

I software di riconoscimento risolvono questo problema creando dei tri-foni che sono terne di fonemi, composte tenendo conto del contesto. Quindi esisterà un tri-fono per il suono "silenzio-a-llll" e uno per il suono "silenzio-aaa-nnn". Considerando che devono essere contemplate tutte le combinazioni e che in italiano ci sono circa 30 fonemi, otteniamo 303 = 27.000 tri-foni.

Quelli simili vengono raggruppati allo scopo di ridurre i calcoli.

Durante i processi descritti il riconoscitore ipotizza più concatenazioni, basate sulle possibili combinazioni dei diversi fonemi. Calcola la probabilità che ciascun fonema sia nel posto giusto rispetto agli altri e stabilisce quale concatenzione ha maggiori probabilità di essere quella giusta. Ogni centesimo di secondo un nuovo *feature#* si aggiunge e integra le informazioni precedenti, fonema dopo fonema e silenzio dopo silenzio. Le combinazioni meno probabili vengono scartate, come pure le combinazioni impossibili di fonemi per la lingua in esame e viene scartata anche la possibilità che ogni centesimo di secondo inizi un nuovo fonema.

Riduzione dei calcoli e aumento della precisione

Dopo aver riconosciuto una parola dal punto di vista fonetico, sembra semplice trovare il temine corrispondente nel database. Può però accadere che: il parlatore non abbia pronunciato la parola chiaramente, che un rumore imprevisto abbia inficiato il contenuto dell'audio o che la divisione dei fonemi non sia avvenuta correttamente. Può succedere che "sono sotto casa anch'io" sia stato diviso male: "sonoso ttocasa anch'io".

Non trovando le parole corrispondenti nel dizionario, il sistema deve provare ad elaborare i fonemi in un altro modo ripetendo parti della procedura. Deve essere fissato un *time-out* o un numero di cicli, oltre il quale viene usata la parola più probabile, anche se sotto la soglia di sicurezza, oppure viene saltata la parola non riconosciuta per passare alla prossima. Alcuni

software indicano le parole non identificate con un simbolo, es. "…." o "???".

Per ridurre i tempi di calcolo e aumentare la precisione, vengono impostate delle regole per restringere il campo di ricerca. Si consideri che:

- Ci sono milioni di parole, ma normalmente ne vengono usate poche migliaia nella lingua corrente. La ricerca inizia da quelle più ricorrenti.
- Le regole grammaticali e linguistiche possono permettere di scartare combinazioni probabilmente sbagliate. Tra "eccomi sono qui" "ecco mi sono qui" invece di ricominciare confrontando entrambe le frasi con i feature # e i risultati degli HMM, viene, in base al modello linguistico, semplicemente scartata la seconda
- Con opportuni criteri, le sequenze di parole più comuni vengono memorizzate: la sequenza "Il presidente del consiglio dei ministri" è molto più probabile di "il residente del consiglio dei ministri"

Inoltre quando le parole da riconoscere non sono parte del lessico comune ma di un linguaggio specifico, deve essere caricato un dizionario appropriato. Per i software di riconoscimento consumer esistono dizionari con termini del linguaggio medico, giuridico, scientifico, tecnico ecc.

5. La modalità di composizione dei fonemi in parole in base ai contesti

5.1 Riconoscimento privo di grammatica

È il campo in cui i sistemi di riconoscimento vocale operano con maggiore efficienza.

Come descritto nell'introduzione, si tratta di contesti in cui il vocabolario e la struttura sintattica delle frasi da riconoscere sono limitati e in cui le scelte possibili sono previste a priori. Il software di riconoscimento può scartare tutte le risposte non contemplate. Anche nel caso di pronuncia non chiara la scelta che il software deve effettuare è semplice: se la parola pronunciata non è riconducibile alle parole che si aspetta di "sentire" la scarta e, se previsto, può chiedere di ripetere.

Un esempio:

Supponiamo di analizzare il caso di un sistema di domotica che disponga di un telecomando vocale.

L'utente conosce a priori quali comandi il suo sistema può accettare: ad esempio

Elenco comandi:

(Accendi le luci | telefona a |
invia una mail a | Componi)

Elenco parametri:

(in salotto | in cucina | nella scala |
Giovanni Marzio Patrizia |
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)

Sintesi vocale: "*Dire un comando*"

L'utente pronuncia un comando.

Il sistema, deve solo riconoscere i comandi sopra elencati e imporre regole prestabilite.

Le frasi che potrebbero scaturire da questo esempio sono del tipo:

Telefona a Patrizia

Invia una mail a Marzio

Accendi le luci nella scala

Componi 0621456690

Il riconoscitore è anche programmato per applicare un filtro sintattico: se l'utente pronuncia "*Componi*", il resto della frase non può che essere una sequenza di numeri. In questo caso il riconoscitore dopo aver identificato la parola "*Componi*" si aspetta di sentire dieci possibili vocaboli: *zero*, *uno*, *due*, *tre* … *nove*.

5.2 Dettatura discreta e Modello linguistico

Ormai la potenza dei moderni personal computer permette di utilizzare software di dettatura continua. Maggiore è la velocità del processore più rapidamente si può dettare, maggiore è la dimensione della memoria RAM disponibile, più alta è la precisione nel riconoscimento (i calcoli vengono fatti su porzioni di audio più grandi).

Per completezza si espone anche il funzionamento a dettatura discreta anche se non vi sono più molte applicazioni per questo tipo di tecnologia. L'argomento però introduce il concetto di modello linguistico, usato anche dai più moderni algoritmi per il riconoscimento della dettatura continua.

Nella dettatura discreta, il parlatore è privo di vincoli e può dire qualsiasi parola compresa in un dizionario che può essere grande a piacere (solitamente qualche migliaio di lemmi). Durante la dettatura deve essere lasciato uno spazio tra le parole per indicare al software quando una parola finisce e un'altra inizia.

Come già indicato, la sequenza di processi che permettono il riconoscimento automatico del parlato sono:

1. Trasformazione dei dati audio dal dominio del tempo al dominio della frequenza tramite FFT
2. Organizzazione dei dati tramite l'applicazione delle regole proprie del dizionario fonetico e della lingua in uso.
3. Riconoscimento dei singoli fonemi
4. Composizione dei fonemi in parole e in sequenze di parole mediante applicazione di modelli linguistici

Il modello linguistico

Gli algoritmi (*tool*) che compongono i database di riferimento e realizzano il training del software, oltre ad analizzare i fonemi, effettuano le analisi statistiche anche sulle parole, elaborando una grande quantità di frasi. Si parla di numeri dell'ordine delle decine di Giga Byte di testo, alcuni milioni di parole: per rendersi conto della mole di dati contenuti in questi data base, si consideri che tutta la produzione letteraria di Dante Alighieri occupa qualche dischetto da 1,44 MB.

Da questa analisi di milioni di frasi vengono ricavate delle statistiche. Per esempio, data una parola, viene ricercata l'evidenza di schemi che indichino se è frequentemente seguita da un'altra (es. "Il Santo" è spesso seguita da "Padre"). In pratica, in fase di utilizzo, quando il riconoscitore individuerà un qualsiasi vocabolo, mentre sta lavorando sui fonemi di quella successiva, ipotizza una lista di parole che potrebbero seguirlo, in base al calcolo delle probabilità. Quando ha finito il lavoro sui fonemi e individua la parola, la confronta con la lista. Se la parola è presente, la può considerare riconosciuta e procedere oltre. Naturalmente il processo funziona nei due sensi, sia per la parola che precede che per quella che segue. In realtà il processo non viene fatto sulla singola parola ma su terne di parole:

Realizzazione del modello linguistico

Prendiamo ad esempio questa frase:

> *"Sono le 8: gli operai metalmeccanici entrano nei cancelli della fabbrica."*

Viene convertita in

> *"Sono le otto duepunti gli operai metalmeccanici entrano nei cancelli della fabbrica punto"*

La frase viene suddivisa in terne di parole.

> *<inizio frase>*
> *Sono le otto*
> *le otto duepunti*
> *duepunti gli operai*
> *gli operai metalmeccanici*

operai metalmeccanici entrano
metalmeccanici entrano nei
entrano nei cancelli
cancelli della fabbrica
della fabbrica punto
<fine frase>

Questo esempio, riguarda una frase sola per cui ogni terna di parole compare una sola volta. Su milioni di frasi invece, si delineano delle statistiche. Ad esempio la terna *"sono le otto"* o *"gli operai metalmeccanici"* compariranno più volte.

In un caso reale, guardando un segmento delle analisi vedremmo ad esempio statistiche di questo tipo:

….
….

Terna di parole	ricorrenze
città del Vaticano	3125
città di Roma	6122
città di toma	2
città di tufo	22

….
…..
altre terzine

La terna *"città di Roma"* compare 6122 volte, mentre *"città di tufo"* solo 22. Quando viene pronunciato "citta di…" ci sono maggiori probabilità che la parola successiva sia *"Roma"* invece di *"tufo"*. Se il pattern fonetico non è chiarissimo ma "assomiglia" a *"Roma"*, il software interpreta *"Roma"* e si occupa della parola successiva.

Naturalmente può accadere che una combinazione di parole non sia mai stata "sentita" da sistema e non sia nel suo database. Se l'utente pronuncia chiaramente e il riconoscimento fonetico termina con un risultato probabilistico alto, la frase potrebbe essere *"la città di scatole"*. In questo caso il riconoscimento "puro" ha prevalso sul modello linguistico. Naturalmente il software, aggiorna il suo database con questa nuova combinazione, indicando di averla sentita almeno una volta (o una volta in più).

L'utilizzo del un modello linguistico riduce notevolmente il tasso di errore, mediamente portando la precisione dall' 80% al 95%.

Utilizzo del modello linguistico, vantaggi e svantaggi:

- Il riconoscimento è più veloce, in quanto combinazioni improbabili vengono scartate a priori.
- Vengono risolte le ambiguità legate alle parole omofone, rare in italiano ma frequenti in altre lingue: ad esempio *"vizi"* e *"Vizzi"* o *"l'oro"* e *"loro"* possono essere pronunciate esattamente allo stesso modo. Come in inglese accade per *"to"*, *"too"* e *"two"*. In questi casi è solo il modello linguistico a determinare la scelta.
- Esistono anche alcuni svantaggi: se due parole appaiono simili, il software di riconoscimento sceglie la più probabile in base al training ricevuto. Purtroppo ci sono spiacevoli effetti collaterali, spesso imbarazzanti.

5.3 Dettatura continua

Il riconoscimento a dettatura continua permette di parlare con velocità normale, senza limitazioni di vocabolario. E' un processo molto più complesso perché il riconoscitore deve stabilire autonomamente l'inizio e la fine di ogni parola. Viene anche richiesta la pronuncia della punteggiatura.

La dettatura continua impiega strumenti matematici più complessi e, rispetto alla dettatura discreta, tutti i processi della struttura (FFT, analisi dei *feature#* per ogni fonema, divisione dei fonemi, combinazione in parole e modello linguistico) richiedono maggior complessità nei calcoli e una finestra di campioni più grande. La dettatura continua in tempo reale è possibile, in modalità *speaker dependent*, con un moderno personal computer. Per cifre dell'ordine del centinaio di euro, esistono in commercio software che offrono un riconoscimento molto preciso.

Invece, i software che lavorano in modalità a dettatura continua *speaker independent* presentano, ad oggi, forti limitazioni nella qualità del riconoscimento. La percentuale media è del 70 – 80 %.

Il testo ricavato da un riconoscitore *speaker independent* è utilizzabile per scopi statistici e di documentazione, in cui sia importante avere una traccia testuale del contenuto dell'audio ma non sia richiesta la precisione delle singole parole.

A esempio, attualmente non si può usare la dettatura continua *speaker independent* per ricavare dei sottotitoli automatici. Quando necessario, la soluzione adottata a tale scopo è di utilizzare personale addestrato ad ascoltare audio, elaborare in tempo reale una sintesi e pronunciare il sottotitolo. Un software *speaker dependent* opportunamente addestrato scrive il sottotitolo che risulterà corretto al 95-98%.

Ci sono forti interessi in gioco per quanto riguarda le tecnologie di riconoscimento vocale, sia da parte dell'industria, sia da parte del terziario sia da parte delle società produttrici di software per l'informatica consumer. Molti esperti sono convinti che il riconoscimento vocale costituirà il nocciolo dell'interfaccia uomo-macchina del futuro. Per questo motivo nuovi algoritmi vengono introdotti sul mercato frequentemente, tuttavia i processi impiegati sono riconducibili a quelli descritti in precedenza.

Una menzione particolare in questo campo meritano le reti neurali.

Le reti neurali

La limitazione dei computer è la loro rigidità. Anche se sembrerebbe assurdo parlare di rigidità in un sistema così flessibile, in realtà i computer non fanno altro che eseguire dei programmi. Un programma è definito come "una sequenza ordinata di istruzioni che eseguite sequenzialmente risolvono un problema". Il computer però non è in grado di adattarsi a situazioni nuove non

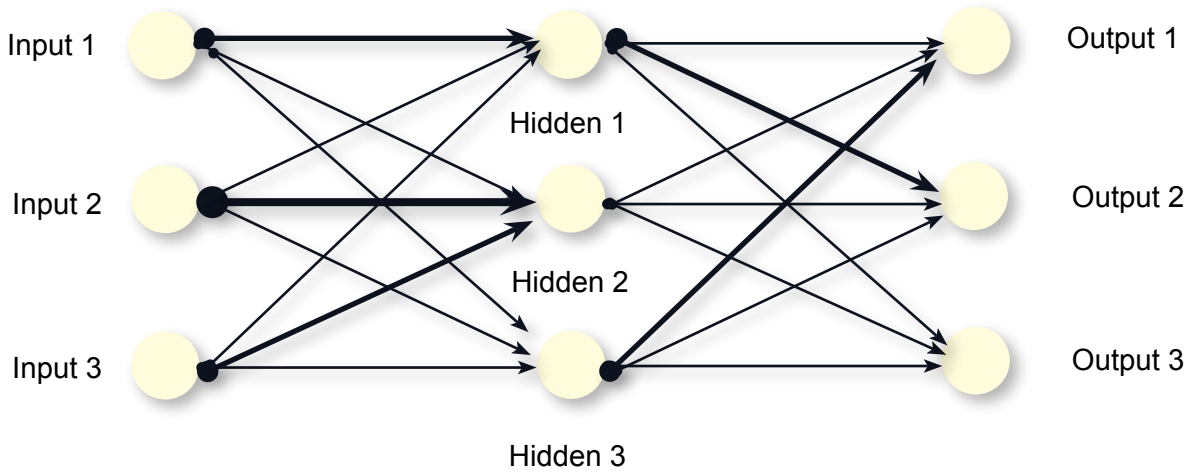

Fig. 6 - Esempio di rete neurale: la rete neurale si chiama in questo modo per similarità con le strutture che compongono il nostro cervello. I neuroni sono collegati da assoni e sinapsi. Nell'esempio il primo stadio è costituito dagli ingressi. I dati vengono convertiti in segnali compatibili con gli stadi successivi. Il secondo stadio, quello nascosto (hidden) elabora effettivamente i segnali. Il terzo stadio è quello di uscita e raccoglie i risultati adattandoli alle richieste del blocco successivo della rete neurale. Ogni collegamento e ogni stadio possono avere un peso ed una importanza maggiore degli altri. In questo modo la stessa rete può fornire risultati diversi con gli stessi imput, esattamente come due persone reagiscono in modo differente in una situazione analoga. Il modo in cui vengono trattati i dati è caratterizzato dall'algoritmo che regola il comportamento della rete.

precedentemente codificate. Può rispondere ad uno stimolo se è stato programmato per farlo e anche le risposte sono previste a priori nel programma.

Nelle reti neurali (figura 6) ci sono molte unità di elaborazione indipendenti collegate tra loro. Il nome "rete neurale" è dovuto all'analogia con il nostro cervello, in cui i neuroni sono in grado di funzionare singolarmente e sono collegati tra loro tramite gli assoni e le sinapsi.

Nel nostro cervello gli impulsi elettrici viaggiano attraverso i collegamenti. Un neurone si attiva quando riceve un impulso sufficientemente forte. A sua volta il neurone emette un impulso elaborato in base a quello ricevuto inviandolo a tutti i neuroni ad esso collegati. I collegamenti tra i neuroni possono attenuare l'impulso modulandone l'intensità, fino a far si che in certe direzioni esso si spegna del tutto. Le reti neurali, analogamente, sono composte da tante unità collegate e da un algoritmo che può modificare i pesi (l'attenuazione) dei singoli collegamenti, in modo che il segnale di input prenda una certa direzione e porti ad un certo output.

Per addestrare la rete e migliorare l'algoritmo, si invia un impulso all'ingresso della rete e si osserva l'output. Si modificano poi i pesi dei collegamenti fino ad ottenere un output più vicino a quello desiderato. Si ripresenta un input, si valuta l'output e si ripete il processo finché è necessario. Una rete neurale, dopo la fase di addestramento, è in grado di fornire un output coerente, anche se riceve un input che non era stato presentato in fase di addestramento. Proprio per questo motivo le reti neurali trovano applicazione nel riconoscimento vocale e nel riconoscimento dei caratteri. In quest'ultimo caso si addestra la rete a riconoscere i caratteri finché la rete non distingue perfettamente tutte le lettere dell'alfabeto.

A questo punto dato un simbolo in input, la rete è in grado di stabilire a quale carattere somigli di più.

Conclusioni

Solo pochi anni fa i sistemi per il riconoscimento vocale presentavano un tasso di errore così elevato da renderli inutilizzabili nella pratica. Oggi invece i software di riconoscimento dipendenti dal parlatore offrono una precisione del 95 - 99% e sono in grado di "apprendere" dai propri errori. I software indipendenti dal parlatore, invece, sono mediamente meno precisi. Vengono impiegati per scopi statistici e di documentazione oppure nei call-center vocali automatici.

Un elemento critico, in grado di peggiorare fortemente il riconoscimento è la qualità dell'audio in esame o la presenza di rumori e/o musica.

Molti esperti sono convinti che il riconoscimento vocale costituirà il nocciolo dell'interfaccia uomo-macchina del futuro, per questo ci sono forti interessi in gioco che inducono il mondo dell'industria e della ricerca a perfezionare le tecnologie e gli algoritmi che permettano il riconoscimento a dettatura continua immune da errori.

Bibliografia

1. A tutorial on Hidden Markov Models and selected applications in speech recognition, L. Rabiner, 1989, Proc. IEEE 77(2):257--286.
2. What HMMs can do, Jeff Bilmes, U. Washington Tech Report, Feb 2002
3. Markovian Models for Sequential Data, Y. Bengio, Neural Computing Surveys 2, 129--162, 1999.
4. Acoustic Modelling - Microsoft Reseach www.http://research.microsoft.com/srg/acoustic-modeling.aspx
5. The most comprehensive site on Artificial Intelligence on the net http://www.generation5.org/