Atti delle IX Giornate di Studio del G.F.S.
Venezia, 17-19 Dicembre 1998

CSLU Toolkit
Il riconoscimento automatico del linguaggio naturale alla portata di tutti

Piero Cosi
Istituto di Fonetica e Dialettologia – C.N.R.
Via G. Anghinoni, 10 - 35121 Padova (ITALY)
Tel: 049 8274421 Fax: 049 8274416
e-mail: cosi@csrf.pd.cnr.it www: http://www.csrf.pd.cnr.it

SOMMARIO

In questo lavoro sono descritte le principali funzionalità del software denominato CSLU-Toolkit ed è sottolineato come questo costituisca un insieme integrato di specializzate tecnologie di programmazione che rappresentano lo *stato dell'arte* negli strumenti per la ricerca, lo sviluppo e l'apprendimento dei sistemi di riconoscimento del linguaggio naturale. Al fine di descrivere in modo più dettagliato il sistema, sono presentati inoltre i risultati ottenuti, a fronte della collaborazione fra OGI-CSLU (Oregon Graduate Institute - Center for Spoken Language Understanding) e IFD-CNR (Istituto di Fonetica e Dialettologia – Consiglio Nazionale delle Ricerche), in un esperimento di riconoscimento automatico, indipendente dal parlante, di stringhe di numeri connessi in italiano, mediante un'architettura completamente realizzata mediante l'applicazione del software CSLU-Toolkit.

INTRODUZIONE

I sistemi di riconoscimento automatico del linguaggio naturale rendono possibile all'uomo di interagire con il computer mediante la voce, il metodo di comunicazione umana più naturale e comune. Questi sistemi sono studiati e realizzati mediante le conoscenze acquisite nel corso degli ultimi anni nel campo del riconoscimento automatico, dell'elaborazione del linguaggio naturale e delle tecnologie per l'interfaccia uomo-macchina. Essenzialmente si basano sul riconoscimento delle parole pronunciate, sull'interpretazione della loro sequenza al fine di ottenerne un

opportuno significato e sull'attuazione di un'adeguata risposta. Le potenziali applicazioni sono numerosissime e, sebbene questi sistemi siano sostanzialmente agli albori, è oltremodo facile intuire la loro enorme potenzialità nel poter rivoluzionare il modo in cui le persone nel futuro si rapporteranno con le macchine. Interagendo in modo naturale, senza cioè dover sottostare ad una specifica fase di addestramento, un sempre gran numero di persone, non necessariamente specializzate, sarà introdotto all'uso di queste tecnologie. Negli ultimi anni le tecnologie relative alla realizzazione di questi sistemi di riconoscimento automatico del linguaggio naturale hanno subito una fortissima accelerazione. Numerosi e notevoli sono stati i passi avanti compiuti nel campo della ricerca. Si possono, infatti, a tutt'oggi osservare un gran numero di sistemi funzionanti in compiti specifici, quali, la pianificazione di viaggi, l'esplorazione urbana ecc.. Ormai, non si può più parlare d'esclusivi prototipi di ricerca appannaggio di pochi laboratori scientifici, ma di vere e proprie applicazioni operanti in tempo-reale, su parlato continuo, indipendentemente dal parlante che non deve più sottostare a lunghe sedute d'addestramento, e supportati da vocabolari di 1000 e più parole. Questi sistemi devono essere assai più robusti degli iniziali prototipi di ricerca in quanto devono essere utilizzati in condizioni naturali quindi in presenza di rumore, sia di canale sia d'ambiente, in condizioni d'utilizzo che devono essere ugualmente soddisfacenti indipendentemente dal variare della velocità d'eloquio, dell'accento o del sesso dell'utilizzatore. Devono esibire inoltre un comportamento 'intelligente', devono in pratica essere in grado di saper reagire anche in condizioni di parziale riconoscimento, che può avvenire in seguito all'occorrenza di pronunce scorrette da parte dell'utente o a causa d'altri fenomeni indesiderati. Dovranno esibire inoltre la capacità di integrarsi efficacemente con altri modi di comunicazione, cercando di "capire" in anticipo le intenzioni dell'utente attraverso le sue espressioni facciali, il movimento delle labbra, degli occhi ecc. e sfruttando tutte le molteplici potenzialità multimediali offerte dalla tecnologia per elaborare le proprie azioni come risposta ai quesiti dell'utente rendendo l'interazione oltremodo naturale ed immediata.

Purtroppo lo sviluppo di un sistema di riconoscimento del linguaggio parlato è un'attività molto complessa che generalmente richiede, per la progettazione, la valutazione e la vera e propria implementazione del sistema, un lungo periodo che può facilmente durare parecchi mesi o meglio alcuni anni. Per poter sfruttare efficacemente questa nuova tecnologia, un sempre maggior numero di laboratori deve poter disporre di strutture informative adeguate ed è impensabile che le conoscenze necessarie allo sviluppo di una tale tecnologia siano parcellizzate e non comunemente utilizzabili. E' con questo obiettivo che all'*Oregon Graduate Institute* (***OGI***) di Portland, ed in particolare presso il *Center for Spoken Language Understanding* (***CLSU***)[1], è stato sviluppato il software denominato ***OGI-Toolkit*** [2] [1]

---

[1] Oregon Graduate Institute of Science and Technology (OGI) - Center for Spoken Language Understanding (CSLU), P.O. Box 91000, Portland Oregon 97291-1000 USA, http://cslu.cse.ogi.edu.

[2] Per coloro che volessero replicare i risultati illustrati in questo lavoro oppure provare ulteriori esperimenti, il software CSLU-Toolkit può essere facilmente recuperato (gratuitamente per scopi accademici e di ricerca) all'indirizzo Internet: http://cslu.cse.ogi.edu/.

che ha proprio lo scopo di fornire ad un sempre più elevato numero di ricercatori, come anche di non addetti ai lavori, lo strumento necessario per creare e sviluppare personalmente in modo semplice ed interattivo nuovi sistemi di riconoscimento del linguaggio naturale sempre più orientati alle applicazioni [2]. Per gli utenti più esperti in tecnologie vocali i Toolkit rappresentano un vero e proprio banco di prova, efficacissimo per lo sviluppo e la verifica delle proprie ricerche, anche le più avanzate [3-5].

CSLU-TOOLKIT

Il software denominato CSLU-Toolkit è stato progettato per facilitare lo sviluppo della ricerca e delle sue possibili applicazioni, nel campo delle tecnologie vocali per un'ampia gamma d'utilizzatori e d'utilizzazioni. Fra le molteplici possibilità si possono elencare:

- l'abilitazione di esperti in specifici domini di applicazione all'utilizzo delle tecnologie vocali per la progettazione di sistemi di riconoscimento del linguaggio naturale, anche multi-lingue, con semplici strumenti di sviluppo;
- la produzione di sistemi di riconoscimento di elevate prestazioni a partire da specifiche di progetto di alto livello;
- l'apprendimento delle tecnologie vocali, con particolare riferimento ai sistemi di dialogo interattivo, mediante opportuni corsi introduttivi incorporati all'interno dei Toolkit;
- la realizzazione di ricerche sull'interazione uomo-macchina con particolare riferimento ai sistemi di dialogo;
- lo sviluppo delle ricerche sulle tecnologie vocali per una loro introduzione in applicazioni reali ed una loro valutazione.

Il software CSLU-Toolkit è un complesso e completo ambiente di sviluppo che integra in se un insieme di tecnologie vocali che comprendono il riconoscimento automatico del linguaggio naturale, anche in ambiente telefonico, la sintesi automatica della voce[3], e l'animazione video di facce parlanti[4]. Nel sistema sono inclusi vari strumenti di programmazione essenziali per una facile implementazione delle applicazioni. L'architettura generale, come illustrato in Figura 1, è composta da tre elementi principali: un insieme di librerie contenenti i moduli di base, generalmente generati utilizzando il linguaggio di programmazione C++, un'opportuna ed immediata interfaccia di programmazione (***CSLUsh*** [6]) e un sistema di sviluppo grafico per una più semplice realizzazione delle applicazioni

---

[3] Il software CSLU-Toolkit è strettamente integrato con il sistema software di sintesi da testo FESTIVAL, sviluppato presso l'Università di Edimburgh: http://www.cstr.ed.ac.uk/projects/festival.html

[4] Il software CSLU-Toolkit è strettamente integrato con il sistema software di animazione di volti parlanti BALDI sviluppato presso l'Università di California, Santa Cruz: http://mambo.ucsc.edu/

finali (***RAD**, Rapid Application Developer*), questi ultimi realizzati mediante l'utilizzazione del sempre più diffuso linguaggio ***Tcl/Tk*** [7].

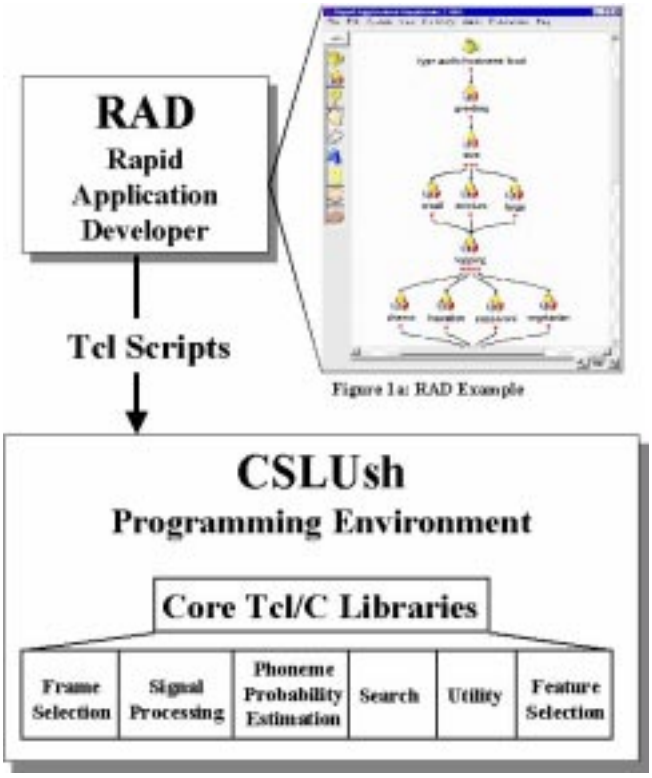

Figura 1. Architettura del software CSLU-Toolkit.

Moduli o librerie di base

Il *cuore* del sistema consiste in un insieme di moduli che implementano gran parte degli aspetti fondamentali relativi alle tecnologie vocali. Queste librerie, generalmente realizzate in C++, rappresentano l'interfaccia di programmazione (**API**, *Application Programming Interface*) indipendente dall'*hardware* e dai diversi sistemi operativi utilizzati. I singoli moduli comprendono specifiche funzioni per l'elaborazione e l'analisi del segnale vocale, per l'apprendimento di reti neurali artificiali (**NN**, *Neural Networks)* e di catene di Markov Nascoste (**HMM**, *Hidden Markov Models*), per il riconoscimento automatico mediante algoritmo di ricerca di ***Viterbi*** [8] e per l'utilizzazione di un'interfaccia telefonica. Inoltre sono inclusi un robusto ***parser*** del linguaggio naturale [9], una versione aggiornata del sistema di sintesi da testo scritto denominata ***Festival*** [10], sviluppata presso l'Università di Edimburgo e un sistema di animazione di volti parlanti denominato ***Baldi*** [11] sviluppato presso l'Università di California, Santa Cruz, oltre a dizionari testuali di

pubblico dominio. L'insieme di questi moduli è molto flessibile in quanto i singoli moduli possono essere facilmente compilati all'interno di un singolo programma o richiamati individualmente all'interno dell'interfaccia di programmazione, secondo quanto richiesto dalle specifiche applicazioni.

CSLUsh: l'interfaccia di programazione

Il livello principale per lo sviluppo delle applicazioni è costituito dall'interfaccia di programmazione denominata CSLUsh, pronunciato come "*slush*", ed è interamente basata sul diffusissimo e portabilissimo linguaggio di programmazione interattivo denominato Tcl/Tk [7]. CSLUsh incorpora, infatti, i moduli di base API precedentemente descritti mediante specifiche funzioni interamente realizzate in Tcl/Tk. La funzionalità di ogni modulo di base è resa disponibile mediante opportuni comandi d'interfaccia regolati da convenzioni di chiamata standardizzate ed i dati sono individuabili come oggetti che possono essere trasferiti nella rete di comunicazione ed essere salvati su disco in modo completamente trasparente ai dispositivi e ai sistemi operativi utilizzati. Il codice che implementa quest'efficiente interfaccia è reso inoltre disponibile all'utente sia per estendere le funzionalità dei moduli di base sia per aumentare la fusione dei moduli stessi quando questa non sia sufficiente. Un'applicazione è quindi costruita fondendo insieme i moduli di base API e utilizzando funzioni aggiuntive d'interfaccia, quali, ad esempio, la gestione degli eventi relativi all'organizzazione dei vari *file*, degli eventi collegati ai problemi della rete, oppure degli eventi relativi all'interfaccia utente grafica. Le varie applicazioni possono essere eseguite anche su piattaforme *hardware-software* differenti e connesse ad una rete locale (LAN), oppure in Internet utilizzando le specifiche funzionalità *client-server* fornite dal linguaggio Tcl, quali i protocolli TCP e UDP, la capacità di funzionare come *daemon*, l'esecuzione remota di comandi Tcl ed il trasferimento dei dati-oggetti. I moduli di base API sono raggruppati in librerie funzionali che sono richiamate, dinamicamente, in fase d'esecuzione, rendendo le applicazioni scalabili in termini di risorse computazionali. Questo consente anche un'estensione delle funzionalità del software senza che questo debba essere ricompilato in seguito ad eventuali aggiunte o modifiche.

RAD (Rapid Application Developer): l'interfaccia di sviluppo grafica

Il terzo componente, a livello più "*alto*", è costituito dall'ambiente di sviluppo grafico denominato ***RAD*** (Rapid Application Developer). RAD integra i moduli per il riconoscimento vocale, la sintesi, l'animazione di agenti parlanti e gli strumenti di visualizzazione in un efficace ambiente grafico di sviluppo per la realizzazione e l'esecuzione di semplici applicazioni. RAD include una tavolozza d'oggetti di dialogo grafici ed una semplice interfaccia di tipo *drag-and-drop*. Gli oggetti della tavolozza sono dei veri e propri blocchi di programmazione grafica, che l'utente seleziona ed organizza collegandoli appropriatamente per realizzare un modello di dialogo a stati finiti. Durante la fase di esecuzione RAD fornisce invece una vista animata delle procedure di dialogo. L'utente può facilmente alternare alla fase di progettazione la

fase d'esecuzione rendendo, così, più efficace lo sviluppo incrementale e il miglioramento interattivo delle applicazioni finali. L'insieme degli oggetti nella tavolozza comprende tutta una serie di funzioni tipiche di un sistema di dialogo quali ad esempio: la risposta al telefono, la sintesi vocale di messaggi, la registrazione di specifici comandi o messaggi vocali, oppure l'identificazione di un segnale d'ingresso telefonico ad impulsi (DTMF). L'interfaccia è stata progettata al fine di richiedere all'utente una conoscenza minima delle problematiche relative alle tecnologie vocali. Particolare cura è stata poi riposta nel cercare di semplificare al massimo la procedura di sviluppo delle applicazioni. Ad esempio, infatti, la specifica per l'utilizzazione di un sistema di riconoscimento è semplice al punto di specificare soltanto le parole o le frasi che devono essere riconosciute. Allo stesso modo per le procedure relative alla sintesi vocale, è richiesta esclusivamente l'immissione testuale dei messaggi da produrre, oppure l'indicazione di un eventuale collegamento con messaggi precedentemente registrati. L'agente parlante, Baldi [8] è integrato completamente nel sistema ed è automaticamente sincronizzato sia con la voce sintetica sia con quella registrata. Dopo aver progettato un'applicazione l'utente deve indicare il dispositivo vocale di ingresso da utilizzare in fase di esecuzione fra quelli disponibili in una lista di dispositivi d'ingresso, che possono essere, ad esempio, un microfono o un telefono. All'utente è successivamente richiesto di premere il pulsante "*Build*" mediante il quale costruire, vale a dire compilare, l'applicazione e finalmente il pulsante "*Run*", con il quale eseguire l'applicazione stessa. RAD racchiude in sé la potenza e la flessibilità del sottostante ambiente di programmazione. Inoltre all'utente è fornita la possibilità di creare applicazioni sofisticate utilizzando anche singoli moduli esterni, indipendentemente compilati, e richiamabili graficamente dall'interno dell'interfaccia grafica di RAD. Non appena l'utente diventa più esperto con l'ambiente di programmazione grafica, può immediatamente estendere l'insieme iniziale di funzioni fornite da RAD e creare, ad esempio, nuovi front-end di analisi per applicazioni già sviluppate quali la lettura automatica della propria posta elettronica o di pagine *www* testuali.

ARCHITETTURA DEL SISTEMA DI RICONOSCIMENTO

L'ambiente di riconoscimento principale del software CSLU-Toolkit, che pur consente diverse architetture, si basa essenzialmente sulle reti neurali artificiali. Il metodo per addestrare il sistema consiste nell'eseguire una sequenza di comandi o *script* CSLUsh utilizzando dei *file* di descrizione che specificano le caratteristiche dei corpora vocali utilizzati, delle condizioni di addestramento e dell'architettura del riconoscitore. Per l'addestramento di un nuovo riconoscitore, quindi, si devono inizialmente costruire i vari *file* di descrizione che vengono successivamente utilizzati dagli script CSLUsh per realizzare le varie fasi di sviluppo del sistema: l'individuazione ed organizzazione dei *file* corrispondenti al materiale vocale scelto per l'addestramento, la verifica e la valutazione finale, la trascrizione di questo materiale nelle categorie fonetico-acustiche scelte per il riconoscimento, la generazione dei vettori di analisi, la selezione dei dati, l'addestramento e la verifica

della rete neurale, la valutazione finale del sistema. Gli stessi *script* possono essere utilizzati sia per addestrare riconoscitori *general-purpose* oppure riconoscitori specializzati in specifici domini d'applicazione. I riconoscitori inoltre possono essere addestrati in differenti linguaggi. Infatti, attualmente sono stati sviluppati sistemi per l'inglese, l'italiano, il coreano, lo spagnolo messicano e il vietnamita.

Architettura di base

Il sistema di riconoscimento incorporato nei CSLU-Toolkit utilizza, nella sua architettura di base, un approccio "*frame-based*" come illustrato in Figura 2. Il segnale è diviso in *frame* ogni 10 ms e, per ogni *frame*, è calcolato un vettore combinazione di 13 coefficienti **PLP** (*Perceptual Linear Predictive*) [12] e di 13 coefficienti **MFC** (*Mel Frequency Cepstral*) [13]. Questo vettore rappresenta, quindi, l'inviluppo spettrale, opportunamente pesato per meglio evidenziare alcuni rilevanti aspetti percettivi, e la sua corrispondente energia in una finestra di analisi di lunghezza prefissata. Per cercare di normalizzare e quindi di ridurre l'effetto indotto sul segnale dal diverso canale di trasmissione e dalle diverse caratteristiche dei parlanti, il segnale è preventivamente pre-processato nei due casi, rispettivamente utilizzando la tecnica denominata **RASTA** (*RelAtive SpecTrAl analysis*) [14] e la tecnica denominata **CMS** (*Cepstral Mean Subtraction*) [15].

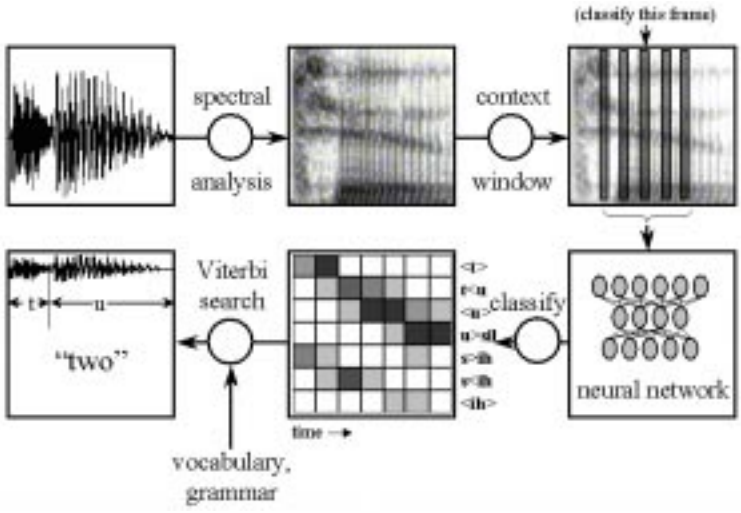

Figura 2. Illustrazione grafica della procedura di riconoscimento del sistema di base del software CSLU-Toolkit, relativo alla parola inglese "two".

Il vettore d'analisi, calcolato ogni 10ms, è fornito all'ingresso della rete neurale per la successiva classificazione nelle specifiche categorie fonetico-acustiche mediante l'algoritmo di ricerca di *Viterbi*. La rete neurale, non riceve in ingresso il vettore caratteristico di un dato *frame,* ma un vettore più esteso considerato come l'insieme dei vettori relativi al *frame* in esame e ad un prefissato numero di frame

adiacenti[5]. Questo "insieme di vettori" è utilizzato al fine di fornire alla rete neurale un'informazione utile sulla particolare dinamica del segnale vocale.

Le singole categorie fonetico-acustiche utilizzate per il riconoscimento possono essere indipendenti o dipendenti dal contesto e sono determinate dal lessico di pronuncia delle parole da riconoscere, come indicato in Tabella 1 nel caso specifico delle cifre inglesi, dal numero di parti con cui dividere ogni unità, a seconda della sua lunghezza e della possibilità che l'unità stessa possa essere più o meno influenzata da fenomeni coarticolatori, e dalla scelta del tipo di raggruppamento in particolari cluster di unità simili, come illustrato in Tabella 2. Fonemi composti da due parti hanno la parte sinistra dipendente dal fonema precedente e quella destra dipendente dal fonema successivo, mentre nei fonemi di tre parti la parte centrale è considerata più stabile ed indipendente dai fonemi adiacenti. Le uscite della rete neurale sono utilizzate come stime della probabilità, per ognuna di queste categorie fonetiche, che il frame in esame appartenga ad una determinata categoria, e la matrice delle probabilità assieme al lessico di pronuncia sono poi utilizzati all'interno di un algoritmo di ricerca di *Viterbi* per determinare la sequenza delle parole più probabili.

word	pronunciation
zero	{dz E r o}
uno	{u n o}
due	{d u e}
tre	{t r E}
quattro	{k w a tt r o}
cinque	{tS I n k w e}

word	pronunciation
sei	{s E I}
sette	{s E tt e}
otto	{O tt o}
nove	{n O v e}
separ	{.pau [.garbage] .pau}

$digit	zero | uno | due | tre | quattro | cinque | sei | sette | otto | nove
$grammar	[separ%%] < $digit [separ%%] > [separ%%]

Tabella 1. Lessico e grammatica per le sequenze di cifre dell'italiano.

phone	parts
.pau	1
n	2
r	2
s	2
v	2
w	2
d	2
t	2
k	2
tt	2

phone	parts
tS	2
dz	2
u	3
o	3
O	3
a	3
E	3
e	3
I	3

group	phones in group	description
$sil	.pau, .garbage	silence
$udp_l	t, tt	unvoiced burst to the left
$udp_r	t, tt, tS	unvoiced closure to the right
$vdp_l	d	voiced burst to the left
$vdp_r	d, dz	voiced closure to the right
f_l	s, tS, dz	frication to the left
f_r	s	frication to the right
$bck	u, o, O	back vowels
$mid	a, E	mid vowels
$frn	i, e	front vowels

Tabella 2. Unità fonetiche, numero di parti per ogni unità e raggruppamenti di unità simili, nel caso di sequenze di cifre in italiano.

[5] Generalmente il vettore risultante è di dimensione 130, in quanto sono considerati 13 coefficienti PLP e 13 coefficienti MFC, nel *frame* corrente e nei 4 *frame* adiacenti relativi a -60, -30, 30, e 60 ms..

Architettura "ibrida" NN/HMM

Un'architettura più complessa realizzabile con il software CSLU-Toolkit è anche quella denominata "***ibrida***" [16-17], risultante dalla combinazione di una rete neurale e di un'architettura basata su HMM, come schematicamente illustrato in Figura 3.

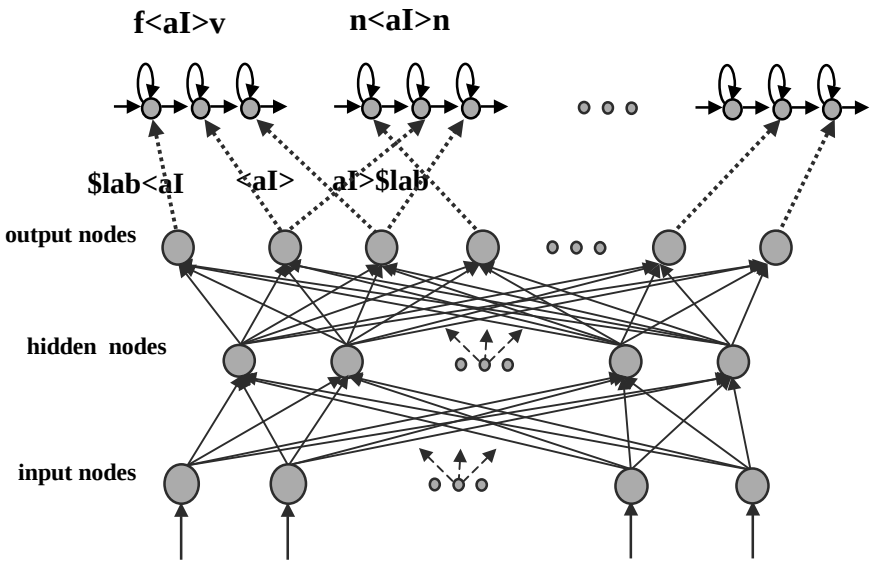

Figura 3. Architettura "ibrida" in cui una rete neurale multistrato è utilizzata per la stima delle probabilità di emissione dei singoli stati di apposite catene di Markov associate alle varie categorie fonetiche.

Analogamente al caso precedente, per ogni frame, la rete neurale classifica i vettori d'analisi in ingresso nelle prescelte categorie fonetico-acustiche, stimando la probabilità che ognuna di queste categorie sia rappresentata dal prefissato vettore d'analisi. Il risultato dell'elaborazione fornita in uscita dalla rete neurale è rappresentato quindi da una matrice CxF di probabilità, dove C è il numero di categorie e F è il numero dei *frame* corrispondenti alla particolare parola o frase in ingresso alla rete. La parola o le parole che meglio sono rappresentate da questa matrice di probabilità sono determinate utilizzando, come nel caso standard, l'algoritmo di ricerca di *Viterbi* vincolato al vocabolario e alla grammatica relativi alle parole da riconoscere. La ricerca è generalmente considerata come l'attraversamento di una sequenza di stati (illustrata in Figura 4 relativamente ad un caso di un semplice vocabolario di due parole inglesi), dove ogni stato rappresenta una particolare categoria fonetico-acustica e vi sono certe probabilità di transizione da uno stato all'altro.

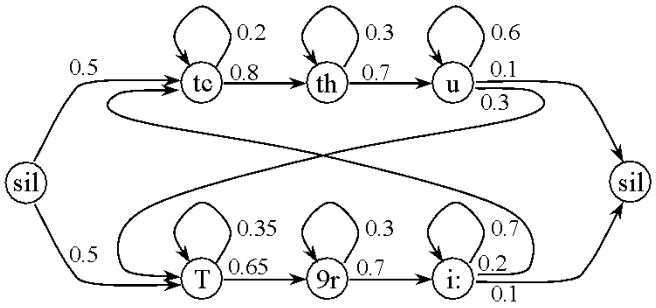

Figura 4. Sequenza di stati in HMM per un semplice vocabolario di due parole.

La differenza maggiore tra quest'architettura ibrida e un'architettura HMM classica risiede nel fatto che le probabilità delle singole categorie sono stimate utilizzando una rete neurale e non mediante l'utilizzazione di un insieme di gaussiane. L'utilizzazione di una rete neurale per questa stima, ha il vantaggio di non richiedere particolari ipotesi ed assunzioni circa la distribuzione di probabilità o l'indipendenza dei dati d'ingresso, e consente, inoltre, di realizzare facilmente un addestramento discriminante fra le singole categorie [18]. Oltre alla maggior velocità della procedura di riconoscimento, un'altra differenza fondamentale, con le architetture HMM standard, risiede nelle caratteristiche delle differenti unità dipendenti dal contesto. Infatti, nelle architetture HMM, le singole unità dipendenti dal contesto sono addestrate considerando il fonema precedente e quello successivo, mentre in quest'architettura ibrida ogni unità è divisa in più stati che sono dipendenti dal contesto destro e sinistro o sono anche assolutamente indipendenti dal contesto.

ESPERIMENTO DI RICONOSCIMENTO DI CIFRE CONNESSE IN ITALIANO

Per quest'esperimento, l'addestramento e la verifica del sistema sono stati effettuati utilizzando una versione filtrata (300-3700 Hz) del corpus ***SPK*** [19]. Questa scelta è giustificata dal fatto che si voleva verificare l'affidabilità del software, anche con un segnale di caratteristiche simili a quelle riscontrabili in ambiente telefonico, benché simulato. Un ulteriore test del sistema è stato effettuato su un sottoinsieme del corpus ***PANDA*** [20]. In particolare, per l'addestramento del sistema, sono state utilizzate 20 ripetizioni delle dieci cifre dell'italiano, pronunciate isolatamente, e 20 differenti sequenze di otto cifre connesse, selezionate in modo casuale, appartenenti a 40 parlatori (19 femmine e 21 maschi) quasi tutti originari del Nord Est dell'Italia. Per l'ulteriore test finale del sistema sono state utilizzate delle sequenze di 15 o 16 cifre connesse registrate su canale telefonico [10]. Il segnale è stato acquisito a 48 kHz e 16-bit di accuratezza e sottocampionato a 16 kHz. Il vettore d'analisi, come precedentemente accennato, è

di dimensione 130 e contiene, per ogni istante, 13 coefficienti PLP e 13 coefficienti MFC relativi al *frame* corrente e ai *frame* adiacenti rispettivamente a -60, -30, 30, e 60 ms.. La rete neurale considerata è a tre livelli, con 200 nodi nel livello intermedio, ed è stata addestrata a classificare 116 diverse categorie fonetico-acustiche risultanti dalle specifiche illustrate nelle Tabelle 1 e 2.

La segmentazione e la corrispondente trascrizione fonetica sono disponibili soltanto per 10 parlatori, che sono stati quindi utilizzati per l'addestramento del sistema di base (***baseline***). L'addestramento della rete è stato fatto per 30 iterazioni e la rete neurale corrispondente all'iterazione che ha fornito i risultati migliori (indicati in Tabella 3) ottenuti testando il sistema sul segnale corrispondente ai rimanenti 30 parlatori, è stata scelta come la rete neurale di base (**B**). Successivamente, mediante questa rete neurale, il materiale vocale corrispondente a tutti i 40 parlatori è stato forzatamente riallineato (***forced alignment***) [5] per ottenere il nuovo materiale vocale su cui addestrare nuovamente il sistema, che è stato poi testato utilizzando questa volta il materiale vocale telefonico vero e proprio contenuto in PANDA. Infine, la nuova rete corrispondente all'iterazione che ha fornito i risultati migliori dopo il riallineamento forzato (**FA**), sempre indicati in Tabella 3, è stata utilizzata per stimare le probabilità di emissione dei singoli stati di apposite catene di Markov associate alle varie categorie fonetiche, come illustrato nella Figura 4. Successivamente, è stato applicato l'algoritmo "***forward-backward***" [16] per stimare nuovi valori target da utilizzare come nuovi *pattern* di addestramento per la rete neurale, che è stata a sua volta testata (**FB**) sullo stesso database telefonico PANDA precedentemente introdotto, ottenendo i risultati sempre indicati nella Tabella 3.

	Itr #	Snts #	Wrds #	Sub %	Ins %	Del %	WrdAcc %	SntAcc %
B	**29**	**4950**	**8800**	**0.26**	**0.10**	**0.10**	**99.53**	**99.29**
FA	**22**	**990**	**15483**	**2.98**	**4.31**	**0.50**	**92.21**	**55.15**
FB	**21**	**990**	**15483**	**2.42**	**4.56**	**0.47**	**92.55**	**53.74**

Tabella 3. Percentuali di corretto riconoscimento a livello di parola (WrdAcc) e di frase (SntAcc) nel caso del sistema di base (**B**), del sistema ottenuto dopo force alignment (**FA**) e del sistema finale ibrido (**FB**). Sono indicati anche l'iterazione che ha fornito i migliori risultati (Itr), il numero di frasi (Snts) e di parole (Wrds) per il test e la percentuale degli errori di sostituzione (Sub), Inserzione (Ins) e cancellazione (Del).

Sia a livello di parola che di frase, le percentuali di corretto riconoscimento, nel caso in cui l'addestramento ed il test siano effettuati sullo stesso tipo di materiale vocale (SPK telefonico simulato), anche se ovviamente su sottoinsiemi disgiunti, sono ottime (>99%). Il test su materiale telefonico vero e proprio (PANDA) è incoraggiante, anche se a livello di frase il risultato risente ovviamente delle caratteristiche non omogenee del materiale di addestramento e di quello di *test* e potrà essere sicuramente migliorato utilizzando materiale vocale telefonico vero e proprio anche nella fase di addestramento.

CONCLUSIONI

Sono state descritte le principali funzionalità del software denominato CSLU-Toolkit ed è stato rilevato come questo costituisca un insieme integrato di specializzate tecnologie di programmazione, rappresentanti lo stato dell'arte negli strumenti per la ricerca, lo sviluppo e l'apprendimento dei sistemi di riconoscimento del linguaggio naturale. I risultati presentati in un esperimento di riconoscimento automatico, indipendente dal parlante, in ambiente telefonico simulato e non, di stringhe di numeri connessi in italiano, ha dimostrato inoltre, l'efficace affidabilità del sistema, in qualità di strumento integrato capace di facilitare l'implementazione di semplici applicazioni di riconoscimento automatico.

RINGRAZIAMENTI

Si ringraziano vivamente tutti i componenti del *Center of Spoken Language Understanding* di Portland, in particolare Ron Cole, Hyneck Hermansky, John Paul Hosom, Jaques de Villiers, Stephen Sutton e Johan Shalkwyck per il continuo supporto e gli importanti suggerimenti.

BIBLIOGRAFIA

[1] Sutton, S., Cole, R.A., de Villiers, J., Schalkwyk, J., Vermeulen, P., Macon, M., Yan, Y., Kaiser, E., Rundle, B., Shobaki, K., Hosom, J.P., Kain, A., Wouters, J., Massaro, D., and Cohen, M., "Universal Speech Tools: The CSLU Toolkit," ICSLP-98, vol. 7, pp. 3221-3224, Sydney, Australia, November 1998.
[2] Cole R., Sutton S., Yan Y., Vermeulen P., Fanty M., Accessible Technology for Interactive Systems: A new approach to spoken language research, Proc. ICASSP-98, II 1037-1040.
[3] Sutton S., Novick D., Cole R., Fanty M., Building 10,000 spoken-dialogue systems, Proc. ICSLP-96, II 709-712.
[4] Hosom J.P., Cole R.A., Cosi P., Evaluation and Integration of Neural-Network Training Techniques for Continuous Digit Recognition, Proc. ICSLP-98, III 731-734.
[5] Cosi, P., Hosom, J.P., Shalkwyk, J., Sutton, S., and Cole, R.A., "Connected Digit Recognition Experiments with the OGI Toolkit's Neural Network and HMM Based Recognizers," IVTTA-ETWR-98, pp. 135-140, September 1998.
[6] J. Schalkwyk, J.H. de Villiers, S. van Vuuren, and P.Vermeulen, "Cslush: An Extendible Research Environment," Proc. of Eurospeech 97, Rodhes, September 1997, pp 689-692.
[7] J.K. Ousterhout, Tcl and the Tk Toolkit. Addison Wesley, 1994.
[8] L. Rabiner and B-H Juang, Fundamentals of Speech Recognition, Signal Processing Series, Alan V. Oppenheim, Series Editor, Prentice Hall, Englewood Cliffs, New Jersey,1993.

[9] Kaiser, E.C., Johnston, M., and Heeman, P.A., “PROFER: Predictive, Robust Finite-State Parsing for Spoken Language,” ICASSP-99, vol. 2, pp. 629-632, Phoenix, AZ, March 1999.
[10] Black, A. and Taylor, P., “Festival Speech Synthesis System: System Documentation (1.1.1),” Human Communication Research Centre Technical Report HCRC/TR-83, Edinburgh, 1997.
[11] Massaro, D. W., Perceiving Talking Faces: From Speech Perception to a Behavioral Principle. MIT Press: Cambridge, MA, 1998.
[12] Hermansky H., Perceptual Linear Predictive (PLP) Analysis of Speech, JASA, 87- 4, 1738-1752.
[13] Davis S.B., Mermelstein P., Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE Trans. ASSP, 28- 4, 357-366.
[14] Hermansky H., Morgan N., RASTA Processing of Speech, IEEE Trans. SAP, 2-4, 578-589.
[15] Furui S., Cepstral Analysis Techniques for Automatic Speaker Verification, IEEE Trans. ASSP, 29-2, 254-272.
[16] Yan Y., Fanty M., Cole R.A., Speech Recognition Using Neural Networks with Forward-Backward Probability Generated Targets, Proc. ICASSP-97, 3241-3244.
[17] Hosom J.P.H, Cole R.A. and Cosi P., Improvements in Neural-Network Training and Search Techniques for Continuous Digit Recognition, AJIIPS Australian Journal of Intelligent Information Processing Systems, in fase di stampa.
[18] Bourlard, H., “Towards Increasing Speech Recognition Error Rates,” Eurospeech’95, vol. 2, pp. 883-894, Madrid, Spain, September 1995.
[19] ELRA web page: http://www.icp.grenet.fr/ELRA/cata/spee_det.html#spk
[20] Chesta C., Laface P., Ravera F., Connected Digit Recognition Using Short and Long Duration Models. Proc. ICASSP-99.