POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informatica e Bioingegneria

LE PRIMITIVE LINGUISTICHE: VERSO UN MODELLO BIOISPIRATO PER L'APPRENDIMENTO DEL LINGUAGGIO IN ROBOTICA COGNITIVA

AI & R Lab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

Relatore: Prof.ssa Giuseppina Gini
Correlatore: Alessio Mauro Franchi

Tesi di Laurea di:
Lorenzo Sernicola, matricola 799452

Anno Accademico 2013-2014

Sommario

L'utilizzo del linguaggio naturale in robotica è stato sempre visto come un problema a parte, nato con lo scopo di ottenere interfacce più dirette tra l'uomo e la macchina. Tuttavia il linguaggio non è unicamente un potente mezzo di comunicazione: esso è infatti profondamente legato all'organizzazione interna della mente stessa e ne guida lo sviluppo.
Lo scopo della tesi è quello di compiere un primo passo verso l'implementazione di un modello per l'apprendimento del linguaggio integrabile con le altre capacità dei robot, che possa guidare anche lo sviluppo cognitivo dell'automa ed accelerarne l'apprendimento.
Per ottenere questo risultato è stato progettato il LPM che sfrutta gli stessi principi delle primitive motorie per sviluppare le abilità linguistiche in modo bioispirato, imitando il processo di apprendimento che avviene nel cervello dei bambini di pochi mesi, nella fase di canonical babbling.
I risultati ottenuti, confrontati con i dati sperimentali ricavati dall'osservazione di bambini reali, validano il modello e ne confermano la correttezza anche da un punto di vista biologico. Il sistema così realizzato è stato infine integrato con IDRA, un'architettura bioispirata che simula il funzionamento del cervello e gestisce stimoli sensoriali e movimento, così da compiere un primo passo verso l'unione del sistema linguistico con le altre componenti della mente dell'agente.

Abstract

The use of natural language in robotics has always been seen as a separate problem, which was born with the aim to obtain a more direct interaction between man and machine. However, language is not only a powerful means of communication: it is in fact deeply linked to the inner organization of the mind and it guides its development.
The aim of this thesis is to take a first step towards a model of language which can be integrated with the robot's diverse abilities, thus leading to its cognitive development and to the speeding up of its learning capacity.
To achieve this result, the Model of Language Primitives has been elaborated which follows the same principles of the motor primitives, in order to develop language abilities with a biologically inspired approach, imitating the learning process which takes place in the brain of a few months old babies, in the phase of canonical babbling.
The obtained results, compared to the experimental data, extracted from the observation of real children, validate the model and confirm its correctness, even from a biological point of view.
The elaborated system has been finally integrated with IDRA, a biologically inspired architecture which simulates the functioning of the brain and manages sensory stimuli and movement, in order to unify the language system with the other components of the agent's mind.

A i miei genitori, che mi hanno permesso di arrivare sin qui, equipaggiandomi con un buon "software".
A mia sorella, con cui condivido lo sguardo, lo spirito e il sentiero.
A Silvia, che renderebbe umano ed entusiasta di vita persino un robot, meglio di un qualsiasi algoritmo cognitivo.

A tutti quelli che ho incontrato, amato, salutato, che ora fanno parte di me.
I robot sono immobili e inumani perché carenti di simpatia: l'anima si conquista, frammento dopo frammento, soltanto nelle forme degli altri. Per il resto non importa con quale materiale si è fatti, la vita è reale solo se condivisa.

Indice

Indice delle figure

Indice delle tabelle

Capitolo 1

Introduzione

"[...] should we not believe that He [God] has freedom to confer a soul on an elephant if He sees fit? We might expect that He would only exercise this power in conjunction with a mutation which provided the elephant with an appropriately improved brain to minister to the needs of this sort. An argument of exactly similar form may be made for the case of machines [...]"

Alan Turing da "Computing, Machinery and Intelligence"

Negli ultimi anni la robotica si è concentrata molto nello sviluppo di agenti intelligenti, dotati di capacità cognitive in grado di renderli autonomi nella selezione dei propri obiettivi e nella scelta delle azioni da compiere. Per tentare di superare le barriere tecnologiche e trovare soluzioni più efficienti, si è sviluppata la disciplina della robotica bioispirata, che trae spunto dalla natura e dai meccanismi biologici per risolvere problemi complessi.

In questo contesto di ricerca si è studiato spesso l'apprendimento motorio in relazione con lo sviluppo di abilità mentali, mentre la ricerca sull'utilizzo del linguaggio naturale da parte dei robot ha sempre fatto parte di un'area a sé stante, considerata unicamente per la risoluzione di problemi specifici e mai in una prospettiva generale. Tuttavia la capacità di parlare e di comprendere la lingua è profondamente legata all'organizzazione della mente, al modo in cui i concetti sono connessi alle esperienze e ai meccanismi logici che permettono di effettuare delle scelte.

Lo scopo di questo lavoro è dunque quello di iniziare a studiare come l'apprendimento del linguaggio naturale possa essere integrato con le altre discipline della robotica bioispirata e come questo possa contribuire alle implementazioni di sistemi autonomi in grado di apprendere dalle proprie esperienze. Per compiere un primo passo verso questa integrazione, sarà presentato il LPM, che sfrutterà gli stessi meccanismi biologicamente ispirati già utilizzati per l'apprendimento motorio, per approcciare al linguaggio. Infine il modello sarà integrato con IDRA, un sistema cognitivo artificiale che imita il funzionamento delle componenti base del cervello biologico per identificare nuovi obiettivi a partire dagli stimoli sensoriali.

1.1 Inquadramento del problema di ricerca

Se si chiedesse a qualcuno di elencare, in pochi punti, le caratteristiche fondamentali di un robot, quella di saper parlare sarebbe immancabile. L'idea stessa di robot, che per i non addetti ai lavori coincide con l'idea di robot umanoide, non può prescindere dalla capacità di utilizzare in modo disinvolto, a volte ironico persino, il linguaggio naturale.

A questo punto ci si potrebbe chiedere quanto questa caratteristica sia frutto di un immaginario comune fomentato da film e libri di fantascienza, e quanto sia realmente una qualità necessaria: lo studio del linguaggio è davvero una disciplina imprescindibile nell'ambito della ricerca sulla robotica e sull'intelligenza artificiale? Lasciando da parte le esperienze cinematografiche e considerando soltanto l'esperienza con i prodotti attualmente presenti sul mercato, la caratteristica di comprendere il linguaggio non spicca di certo per importanza rispetto alle altre feature del prodotto come durata della batteria, display ad alta definizione, resistenza agli urti e molte altre.

Le tecnologie di riconoscimento vocale attualmente in commercio, come Apple Siri o Google Now, sono state accolte con entusiasmo dal pubblico, tuttavia, studiando le statistiche [1], si può notare che sono utilizzate dagli utenti perlopiù per compiti semplici come mandare un messaggio o effettuare una chiamata, dove l'interazione col dispositivo gioca un ruolo molto marginale; è dunque più importante che il telefono riconosca le parole per trascriverle in formato digitale, piuttosto che comprenda il loro significato.

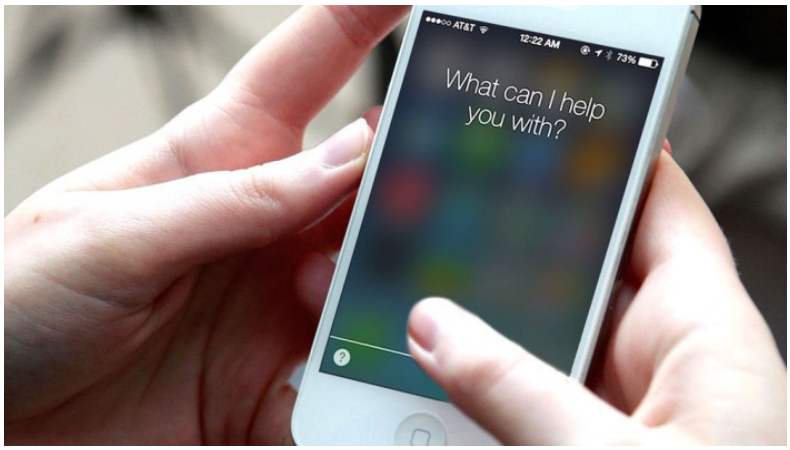


Figura 1.1 - Apple Siri (iOS8) è uno dei sistemi di riconoscimento vocale più utilizzato al mondo nell'ambito dei dispositivi mobile.

Altri esempi di applicazioni esistenti possono essere individuati in alcuni videogiochi di nuova generazione, o in automi, presenti in diversi ambiti, che riconoscono poche decine di parole relative ad un determinato compito, il che semplifica l'utilizzo, velocizza le operazioni, ma non pone comunque l'utilizzo del linguaggio in una posizione privilegiata rispetto ad altri requisiti.

Stando a quanto detto si potrebbe pensare che quella della parola non sia altro che una moda temporanea, per stupire e far acquistare, non differente da uno schermo dai colori più accesi o un audio più limpido. In realtà, contrariamente a questa visione, nell'informatica prima e nell'intelligenza artificiale poi, il linguaggio ha sempre rivestito un ruolo chiave.

Basti pensare che il padre indiscusso dell'informatica, Alan Turing, basò il suo test più celebre proprio sull'utilizzo del linguaggio naturale. Il test, denominato *Imitation Game* e introdotto per la prima volta nel 1950 in "*Computing Machinery and Intelligence*", cerca di definire un criterio di giudizio per poter capire se un eventuale automa sia o meno dotato di intelligenza [2]. Per farlo immagina un dialogo tra un intervistatore, che pone le domande tramite un terminale, e due partecipanti, non visibili dal primo, che sono rispettivamente un uomo e una donna. Entrambi devono convincere l'intervistatore di essere la donna e se l'uomo, mentendo, ci riesce, guadagna un punto. Nella fase successiva a giocare il ruolo dell'uomo è un robot. Alternando queste fasi diverse volte, se il punteggio totalizzato dall'uomo è comparabile con quello del robot, quest'ultimo sarà a tutti gli effetti considerato "intelligente".

Ciò che si evince dal pensiero di Turing, al di là delle obiezioni che nel tempo sono state fatte a questo metodo, è proprio quanto la capacità di utilizzare il linguaggio naturale sia indispensabile a qualunque macchina dotata di intelligenza.

Tuttavia l'importanza del linguaggio non va cercata unicamente nella possibilità di interazione tra l'uomo e la macchina, in realtà esso riveste un ruolo ancora più basilare: è la radice del software stesso. I linguaggi di programmazione, che permettono di costruire il software, sono infatti composti da alfabeti, regole sintattiche e semantiche, tramite le quali si implementano gli algoritmi.

I linguaggi formali, ossia linguaggi logici composti da simboli e regole, sono stati utilizzati in linguistica e in matematica molto prima che i dispositivi elettronici fossero inventati; Frege li trattò per la prima volta nel 1879, nel "*Begriffsschrift*", descrivendoli come linguaggi di "puro pensiero" [3]. Anche prescindendo dalla nozione di linguaggio formale la logica, che fu definita in antichità dagli stessi greci come "l'arte di ragionare", utilizza comunque un linguaggio matematico per esprimere i propri costrutti. La matematica stessa è linguaggio.

Non si tratta dunque solamente di un mezzo di comunicazione, ma il linguaggio costituisce esso stesso informazione, senza la quale qualsiasi computazione, non solo in ambito umano, ma anche nell'ambiente fisico, sarebbe impossibile. Persino il DNA, mattone della vita, si basa su un linguaggio costituito da un alfabeto di quattro lettere, le quali, a loro volta, grazie alla traduzione e alla riorganizzazione svolta dagli enzimi, formano parole secondo regole sintattiche precise, consentendo così la trasmissione del

codice genetico attraverso le generazioni, meccanismo alla base dell'evoluzione [4].

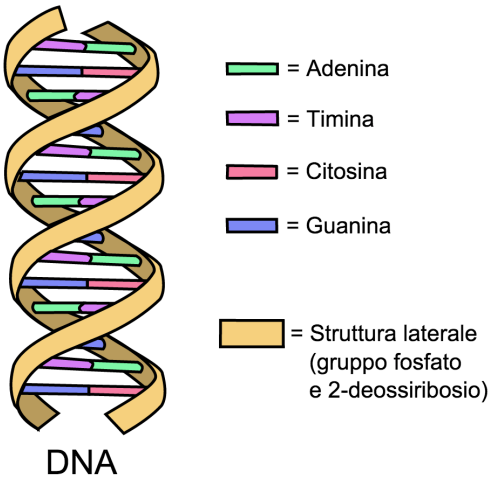


Figura 1.2 - Il DNA è formato da quattro basi azotate differenti, l'Adenina, la Timina, la Citosina, la Guanina, che costituiscono il contenuto informativo del DNA: ogni sequenza di queste lettere produrrà una diversa proteina.

Quanto visto fino ad ora chiarisce l'indubbia importanza che il linguaggio riveste in qualsiasi tipo di fenomeno; tuttavia, l'aspetto che più interessa ai fini di questa trattazione, è il suo rapporto con la mente. In termini evolutivi l'utilizzo del linguaggio è stata l'ultima caratteristica ad emergere nel cervello dell'uomo, ed è la caratteristica che più radicalmente ci distingue dai primati superiori.

In realtà l'abilità di parlare, o meglio, quella di comunicare, non c'è unicamente nell'uomo, ma è presente in molte altre specie che interagiscono tra di loro tramite gesti e suoni, secondo un insieme di regole dettate sia dalla biologia che dal contesto. Tuttavia solo nell'uomo il linguaggio ha subito un'evoluzione così rilevante, ed è dovuta al fatto che non si tratta solo di uno strumento esogeno usato per comunicare un concetto da un individuo ad un altro, ma è intrinseco alla mente stessa [5]. Si sa che la mente, seppur ancora inesplicabile nel suo mistero, dal punto di vista simbolico e computazionale è organizzata secondo categorie che raggruppano concetti, in relazione tra di loro; il ragionamento cosciente, proprio dell'uomo, segue regole sintattiche e principi grammaticali: la lingua parlata è perciò soltanto l'epifenomeno, la punta dell'iceberg che rappresenta la profonda relazione tra linguaggio e cervello.

1.2 L'utilizzo del linguaggio naturale in robotica cognitiva

Nonostante l'importanza che il linguaggio riveste in informatica e nelle discipline che studiano il funzionamento della mente umana, i meccanismi alla base dello sviluppo del linguaggio nei primi mesi di vita non sono ancora stati chiariti del tutto.
Inoltre, nell'ambito della robotica, sono poche le applicazioni che cercano di simulare l'apprendimento linguistico così come accade negli esseri umani: ci si è concentrati molto di più sulla soluzione di problemi specifici, come ad esempio la realizzazione di software in grado di trascrivere le parole dettate, o macchine in grado di rispondere ad un limitato set di comandi vocali [6].

Queste applicazioni, con il passare degli anni, stanno diventando sempre più efficienti, ma continuano a costituire un campo a parte: non cooperano insieme agli altri meccanismi cognitivi dell'agente artificiale per l'apprendimento generale, ma costituiscono un modulo a sé stante, in grado di ricevere set limitati di comandi e tradurli, senza comprendere in alcun modo l'oggetto del messaggio e senza poterne trarre informazioni riutilizzabili per il futuro. Il linguaggio rappresenta dunque unicamente un'interfaccia tra uomo e macchina, senza poter contribuire allo sviluppo della mente dell'agente, così come avviene invece nella natura.

Inoltre, anche i risultati ottenuti tramite l'utilizzo del linguaggio naturale nella comunicazione con gli esseri umani sono piuttosto limitati, nonostante gli sforzi che grandi aziende come Google, Apple, Microsoft stanno facendo in questo senso (in particolare nell'ambito mobile). Questo è dovuto al fatto che si tratta solamente di una simulazione di linguaggio naturale, basata sulla trascrizione dei segnali audio e sull'analisi sintattica delle frasi, meccanismi che tuttavia sono difficili da perfezionare a causa di problemi molto complessi dovuti all'ambiguità intrinseca del linguaggio naturale e ai disturbi nell'analisi dei segnali audio. Infine la gestione dei concetti collegati alle parole ascoltate è solitamente basata su conoscenza fornita alle macchine a priori e non fondata sulle proprie esperienze sensoriali: questa è un'ulteriore conseguenza di costruire i sistemi dedicati al linguaggio totalmente scollegati dagli altri apparati cognitivi [7].

Negli ultimi anni, proprio per superare queste limitazione, alcuni ricercatori hanno iniziato a studiare l'evoluzione della lingua nei bambini nei primi mesi di vita, per cercare di individuare ed utilizzare gli stessi meccanismi anche nei robot. Questa analisi si è rivelata molto complessa, dato che i soggetti sono difficili da osservare, soprattutto se in un contesto naturale; in particolare le interazioni tra i bambini, gli adulti e il modo circostante, che sono alla base dell'apprendimento e dello sviluppo mentale, non possono in alcun modo essere riprodotti in ambiente controllato.

Con lo scopo di ottenere osservazioni valide su cui basare gli studi futuri, nel 2006 è nato lo Speechrome Project [8], che ha previsto l'osservazione continua di un bambino dalla nascita fino ai tre anni di età, tramite una

particolare infrastruttura di microfoni e telecamere montate direttamente in casa e la trascrizione di tutte le parole pronunciate e delle modalità di apprendimento. L'analisi dei dati è ancora in corso, tuttavia non è stato ancora proposto un modello che permetta di imitare l'acquisizione naturale della lingua nei bambini, né la possibilità di integrare un tale sistema con gli apparati cognitivi già sviluppati.

1.3 Scopo di questo lavoro

Sono innumerevoli i problemi ancora da superare nell'ambito dello sviluppo del linguaggio naturale nei robot e negli agenti artificiali. Studiando attentamente quanto realizzato fino ad ora e le carenze di ogni tipologia di sistema, si evince come uno dei limiti più marcati delle tecnologie attuali sia il totale isolamento dei moduli dedicati al linguaggio rispetto a tutti gli altri.

Invece, nell'apparato biologico e in particolare negli esseri umani, le aree cerebrali dedicate al linguaggio sono strettamente connesse con le altre e spesso per l'interpretazione di una frase vengono attivati anche centri che solitamente sono utilizzati per altre attività, come il movimento [9]. Questo è dovuto al fatto che molti meccanismi mentali vengono riutilizzati per diversi compiti, per numerose tipologie di apprendimento, che sia motorio o linguistico o logico. Inoltre il linguaggio non è un mondo a sé stante: per la corretta comprensione delle parole e delle frasi il cervello ha bisogno, oltre che del lavoro dei centri specializzati come l'area di Broca, di attingere anche dalla memoria così da fondare il significato di un'espressione linguistica sulle proprie esperienze sensoriali e logiche [9]. Grazie a questa associazione tra oggetti astratti (le parole) e stimoli reali dal mondo circostante, il cervello è in grado di comprendere quello di cui si sta parlando.

Naturalmente in robotica si è ben lontani dal raggiungere un risultato del genere, che esula anche dagli scopi di questa trattazione. E' di fondamentale importanza tuttavia individuare un punto di partenza nuovo, che si distacchi dall'idea di un apparato linguistico isolato e abbracci una prospettiva d'insieme, in cui diversi meccanismi mentali biologicamente ispirati siano in grado di cooperare ai fini dello sviluppo cognitivo generale dell'agente.

Per individuare questa strada si è scelto di studiare approfonditamente l'evoluzione del linguaggio nei bambini di pochi mesi e di creare un sistema che simuli questo processo in modo bioispirato. Il meccanismo utilizzato, che è stato chiamato Modello delle Primitive Linguistiche (LPM), sfrutta inoltre concetti già utilizzati in robotica nell'ambito del movimento. Questo è importante da una parte ai fini implementativi dato che la letteratura nell'ambito del movimento è molto più approfondita che in quello linguistico, dall'altra ai fini dell'imitazione del sistema biologico dato che in natura i meccanismi cerebrali utilizzati per l'apprendimento motorio e linguistico sono molto simili.

Il sistema così realizzato, a partire da un dataset iniziale di primitive linguistiche fornite a priori (perché fisiologiche nel bambino), tenta di imitare una parola fornita in input componendo i diversi suoni, fino a raggiungere un risultato accettabile. Una volta imitata correttamente, la parola viene appresa ed è subito disponibile nel caso si voglia di nuovo imitare la stessa parola.

Infine abbiamo studiato come integrare il nostro LPM, che si occupa dell'apprendimento del linguaggio, con IDRA, un sistema artificiale bioispirato che simula il funzionamento di tre componenti del cervello: corteccia, talamo e amigdala. Grazie a questa integrazione sarà possibile non solo imitare una porzione del cervello umano dal punto di vista dei meccanismi, ma anche dell'organizzazione strutturale e logica. Inoltre, dato che IDRA gestisce sia l'apparato motorio che quello sensoriale, con questa integrazione si è mosso un primo passo verso un sistema bioispirato in grado di gestire sia l'apprendimento motorio che quello linguistico dati gli stimoli esterni, utilizzando meccanismi condivisi.

I risultati ottenuti, benché difficilmente confrontabili con altri esperimenti precedenti data la novità del lavoro svolto, dimostrano la correttezza delle ipotesi enunciate e del modello realizzato e sembrano indicare che la strada intrapresa sia corretta.

I dati evidenziano inoltre la possibilità di sfruttare un modello fino ad ora utilizzato unicamente per i movimenti, anche per il linguaggio, lasciando ben sperare riguardo ai possibili sviluppi futuri.

1.4 Struttura della tesi

Nel **Capitolo 2** valutiamo lo stato attuale delle tecnologie che utilizzano il linguaggio naturale e ne vengono analizzati differenti utilizzi, mettendo in luce diverse aree di miglioramento. Trattiamo inoltre innumerevoli aspetti teorici che saranno utilizzati nei capitoli successivi e nella parte sperimentale.

Nel **Capitolo 3** studiamo l'approccio bioispirato riportando le numerose motivazioni che ci hanno spinto a sceglierlo come modello implementativo per il nostro sistema. Analizziamo inoltre alcuni aspetti delle primitive motorie che saranno sfruttati per costruire il LPM.

Nel **Capitolo 4** approfondiamo lo studio dei meccanismi cognitivi biologici che portano i bambini di pochi mesi ad apprendere correttamente il linguaggio, rivolgendo particolare attenzione alla fase del canonical babbling, i cui principi saranno utilizzati per costruire il nostro modello.

Nel **Capitolo 5** si studiano diversi metodi di elaborazione ed analisi dei segnali audio tramite Matlab e Praat, che saranno poi utilizzati per gestire le primitive linguistiche all'interno del sistema.

Nel **Capitolo 6** trattiamo in modo approfondito l'architettura del sistema, i processi e i vari moduli che cooperano per far funzionare correttamente il LPM, con particolare attenzione alle diverse fasi di trasformazione dei dati iniziali che saranno utilizzati per l'apprendimento. Sarà infine discussa la struttura che permette di integrare il sistema con IDRA.

Nel **Capitolo 7** si descrivono gli esperimenti preliminari utilizzati per determinare i parametri migliori di funzionamento per il sistema e gli esperimenti di apprendimento svolti, analizzando il significato dei risultati ottenuti anche da un punto di vista biologico.

Nel **Capitolo 8** riportiamo le conclusioni a cui ci hanno condotto i risultati sperimentali analizzati in precedenza e descriviamo diverse possibilità di sviluppo futuro per il nostro lavoro.

Capitolo 2

Stato dell'Arte

"Le storie che si scriveranno, i quadri che dipingeranno, le musiche che si comporranno, le stolte pazze e incomprensibili cose che tu dici, saranno pur sempre la punta massima dell'uomo, la sua autentica bandiera [...] quelle idiozie che tu dici saranno ancora la cosa che più ci distingue dalle bestie, non importa se supremamente inutili, forse anzi proprio per questo. Più ancora dell'atomica, dello sputnik, dei razzi intersiderali. E il giorno in cui quelle idiozie non si faranno più, gli uomini saranno diventati dei nudi miserabili vermi come ai tempi delle caverne."

Dino Buzzati da "Il Mago"

2.1 L'object labeling

Secondo gli studi di molti scienziati il complesso sviluppo cognitivo dei bambini avviene parallelamente a quello del linguaggio, che viene utilizzato come potente strumento per l'(auto)organizzazione dei concetti, ossia per sviluppare una rappresentazione interna del mondo esterno [5].

Diversi studiosi hanno analizzato il ruolo del linguaggio nell'acquisizione e nell'elaborazione delle esperienze senso-motorie, convalidando la teoria secondo la quale le parole apprese funzionerebbero come delle etichette (*label*) per gli oggetti del mondo: ogni volta che ad un oggetto, o meglio all'aspetto visuale di un oggetto percepito tramite i sensi, viene assegnata una parola nel cervello, si semplifica il processo di categorizzazione mentale e memorizzazione e dunque l'apprendimento di quel concetto [10]. L'*object labeling* tramite parole facilita, nello sviluppo del bambino (e nell'adulto anche se in modo minore), anche il processo di discriminazione tra oggetti differenti: parole diverse corrispondono alla rappresentazione mentale di oggetti diversi.

Questi studi, che sono svolti astraendo il cervello e trattandolo da un punto di vista computazionale, funzionale quindi, sono confermati dalle ricerche mediche che lo trattano invece come macchina biologica: si è visto infatti che le costruzioni grammaticali apprese tramite lo studio del linguaggio, interagiscono con l'apparato senso-motorio a livello neurale [11]. La lettura di una parola relativa a una parte del corpo, ad esempio, attiva i neuroni della corteccia celebrale motoria relativa a quella parte. Lo stesso accade in caso di

parole come "*camminare*" o "*vedere*". Dunque l'apparato mentale responsabile dei movimenti e quello responsabile di recepire ed elaborare gli stimoli sensoriali sono profondamente legati e si influenzano a vicenda: la comprensione e l'apprendimento del linguaggio sarebbero impossibili se non si basassero su esperienze fisiche e sensoriali.

2.2 Symbol Grounding Problem, intelligenza artificiale e analisi del linguaggio naturale

Il problema che prende in considerazione la comprensione mentale e il suo rapporto con gli oggetti reali, mette in evidenza quello che è stato, sin dalle origini della robotica e dell'intelligenza artificiale, un problema filosofico, oltre che matematico, di difficilissima soluzione: il *Symbol Grounding Problem* [12]. Il quesito riguarda la relazione che intercorre tra le parole (in senso più generico i simboli) con il loro significato. Se la sintassi, ossia le regole che legano tra loro i simboli, sono semplici da rappresentare in un ambiente simbolico astratto, lo stesso non si può dire della semantica, che collega il simbolo ad un concetto.

Pur tralasciando gli aspetti filosofici, la cui discussione esula dallo scopo di questa trattazione, è senza dubbio necessario affrontare il problema dal punto di vista dell'informatica, perché è alla base di molte difficoltà che sorgono in robotica quando ci si appresta a simulare un linguaggio naturale.

Il *Symbol Grounding Problem*, anche in informatica, nasce dal fatto che non è ben chiaro come i concetti che vengono elaborati all'interno della mente umana, siano collegati con le cose a cui si riferiscono. Non avendo a disposizione una soluzione per quanto riguarda il cervello biologico, è ancora più difficile trattare il quesito nell'ambito delle menti artificiali, il che rende impossibile, allo stato attuale, chiarire fino in fondo quanto in là possa spingersi l'emulazione della mente umana tramite un computer [13].

2.2.1 Chinese Room Argument

Per chiarire il concetto ed esplicitare il problema, si può utilizzare l'esempio della *Chinese Room Argument*, formulato da Searle per la prima volta nel 1980 [14]. Si tratta di un'argomentazione che mina dalle fondamenta la validità della metafora di una mente vista unicamente come computer, una visione tipica dell'*Intelligenza Artificiale Forte*. Secondo quest'ultima visione infatti, il cervello corrisponderebbe ad un hardware e la mente ad un software: il programma implementato basterebbe a garantire l'esistenza degli stati mentali. Dall'altro lato invece, l'*Intelligenza Artificiale Debole,* vede il computer come uno strumento molto potente, che può aiutarci nello studio della mente, ma che non può in alcun modo simularne una.

Mentre Searle accetta l'IA Debole ed è convinto dell'utilità dell'informatica, non crede invece che un computer, ossia un sistema formale che fondamentalmente manipola un linguaggio costituito da simboli, possa in alcun modo attribuire una *semantica*, ossia un significato, a quello che sta facendo.

Per spiegare questo concetto, egli suppone che un uomo sia rinchiuso in una stanza e abbia soltanto due canali di comunicazione con l'esterno, uno in entrata ed uno in uscita. All'uomo giungono dall'esterno dei fogli scritti in una lingua sconosciuta (come può essere il cinese per una persona che non lo conosce). All'interno della stanza ha solamente un libro in cui sono riportate, scritte in una lingua conosciuta (ad esempio in inglese) le istruzioni che indicano cosa fare con ciascun simbolo ricevuto e cosa rispedire fuori come risposta, senza dare alcun dettaglio sul loro significato. Queste regole legano semplicemente un input ad un output. Un osservatore esterno, che invece comprende i simboli in cinese, potrà immettere nella stanza una storia scritta in cinese e delle domande a cui rispondere, sempre scritte nella stessa lingua. Dopo un tempo più o meno lungo, dalla stanza usciranno dei fogli contenenti l'esatta risposta in cinese alle domande.

Si può dire che l'uomo chiuso nella stanza abbia compreso qualcosa della storia? No, naturalmente. Non ha compreso nemmeno che si trattasse di una storia. Si è solo limitato a seguire una sintassi, delle regole a lui fornite, riportate su un libro. Tuttavia queste, anche se scritte in una lingua a lui conosciuta, non lo aiutano in alcun modo ad attribuire ai segni cinesi un significato e quindi una semantica. I simboli da soli non possono garantire proprietà semantiche.

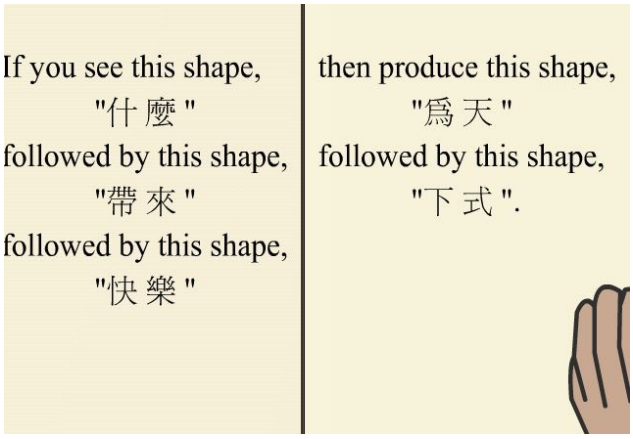


Figura 2.1 - Nel libro presente nella stanza cinese sono indicate soltanto le associazioni tra input ed output. Non contiene il significato dei simboli, né è possibile ricavarlo in alcun modo.

E' allora evidente come l'incapacità di risolvere, perlomeno tramite le conoscenze attuali, il *Symbol Grounding Problem,* influisca negativamente anche sulla possibilità di dotare i robot e, più in generale, gli agenti forniti di

intelligenza artificiale di capacità linguistiche avanzate, che li renderebbero in grado di sfruttare il linguaggio naturale. Allo stesso tempo, avendo chiarito come le capacità linguistiche siano profondamente collegate all'organizzazione interna del cervello e quindi, in ultima analisi, alle sue capacità di classificazione e generalizzazione, si evince come il *Symbol Grounding Problem* rappresenti un freno enorme allo sviluppo dell'intelligenza artificiale e della robotica cognitiva [13].

Tornando dunque alla domanda iniziale, oggetto di questa sezione, studiare come implementare il linguaggio naturale in una macchina non è importante soltanto ai fini dell'enorme passo avanti che una simile caratteristica garantirebbe dal punto di vista dell'interazione uomo-macchina, ma anche, e soprattutto, ai fini dello sviluppo di capacità cognitive nelle macchine, sviluppo che attualmente, come vedremo nella sezione successiva, è piuttosto limitato e soprattutto risulta inefficace se applicato in ambienti reali e non simulati.

2.2.2 Storia del Natural Language Processing in robotica

Prima di inoltrarsi nella discussione delle possibili soluzioni al *Symbol Grounding Problem*, ed illustrare l'argomento centrale della tesi che riguarda la possibilità di progettare un sistema in grado di supportare l'apprendimento del linguaggio in modo bioispirato, è bene esaminare quali siano stati gli step più importanti nell'ambito dello studio del linguaggio naturale nella robotica e quali siano i risultati attuali. In questo modo potremo valutare i punti di forza e gli aspetti problematici da risolvere e comprendere, infine, perché sia importante tentare un approccio differente.

Come illustrato in precedenza, uno dei primi scienziati a considerare il problema del linguaggio in informatica è stato Turing intorno al 1950 [2]. Non è una data casuale, infatti proprio in quegli anni nasce la disciplina del *Natural Language Processing (NLP)*, un campo dell'informatica e dell'intelligenza artificiale fortemente collegato alla linguistica, che studia l'interazione tra uomo e computer dal punto di vista del linguaggio naturale e prende in esame sia il processo di comprensione, che quello di generazione delle frasi.

In realtà per alcune decadi non saranno sviluppate applicazioni pratiche dei principi teorici, data l'assenza di macchine digitali in grado di supportare una tale complessità; negli anni '60 ci furono alcuni tentativi, puramente sperimentali e dimostrativi, di gestire poche parole e alcuni limitati gruppi di frasi, tramite le tecniche di *pattern matching*: esperimenti di questo tipo sono SHRDLU, sviluppato da Terry Winograd al MIT nel 1968 [15], in grado di operare nel *block world*, ossia un ambiente virtuale al cui interno sono simulati vari oggetti, ed ELIZA (1964) [16], in grado di interpretare il ruolo di uno psicoterapeuta, ponendo domande sensate al "paziente" (ossia l'utente al terminale), basandosi sulle risposte e sulle frasi digitate precedentemente.

Negli anni '70, invece, la disciplina del NLP ha preso maggiormente piede col fine di sviluppare i primi correttori grammaticali con cui equipaggiare i tool di scrittura dei sistemi UNIX. Si tratta di programmi in grado di rilevare, ad un livello elementare, errori di punteggiatura ed inconsistenze grammaticali. Questi software funzionavano grazie ad enormi set di regole grammaticali scritte ed inserite manualmente, tramite le quali venivano processate le frasi: questo portava a sistemi molto complessi da progettare e con alte probabilità di errore [17].

I problemi legati all'elaborazione di costrutti in linguaggio naturale e l'alta complessità richiesta nascevano (e nascono tutt'oggi) dal fatto che la lingua parlata comunemente dagli esseri umani (di qualsiasi ceppo linguistico) è molto differente dai linguaggi logici sui quali sono basati i computer. I linguaggi formali delle macchine sono infatti linguaggi artificiali, ossia sottoinsiemi formalizzati di quello naturale, creati intenzionalmente dall'uomo col fine di avere un insieme predeterminato di costrutti, utilizzando un alfabeto finito che genera un numero solitamente finito di stringhe.

Inoltre le frasi hanno una costruzione ben precisa (devono essere "ben-formate"), priva dell'ambiguità che invece contraddistingue il linguaggio naturale.

Quest'ultimo, seppur rappresenti un mezzo potentissimo di espressione, non è completamente formalizzabile: dunque una traduzione simultanea e perfettamente corrispondente del linguaggio naturale in linguaggio formale è impossibile [18]. Questo fatto costituisce un problema non da poco dato che i computer, e quindi qualsiasi agente di intelligenza artificiale, ragiona utilizzando linguaggi formali.

La difficoltà nella traduzione è uno dei problemi principali da risolvere quando si cerca di implementare un'interfaccia uomo-macchina basata sul linguaggio naturale. Questo va ad aggiungersi al *Symbol Grounding Problem* e, come vedremo, al fatto che proprio la profonda ambiguità dei linguaggi naturali li rende fortemente dipendenti dal contesto e dalla conoscenza pregressa che comunemente è presente tra due interlocutori umani.

Un grosso passo avanti nella disciplina dell'NLP è stato fatto negli anni '80, con l'introduzione di algoritmi di *machine learning* che hanno accelerato e semplificato l'analisi del linguaggio naturale, basandosi sugli innumerevoli studi di linguistica, grazie a studiosi come Chomsky [17]. Questo nuovo approccio ha portato alla progettazione di modelli statistici basati sulla probabilità che sfruttano gli alberi decisionali e le Catene di Markov per superare, perlomeno parzialmente, i problemi legati all'ambiguità del linguaggio naturale. I sistemi più stabili, realizzati tramite questi nuovi metodi, hanno suscitato l'interesse di aziende leader nell'informatica come Microsoft che ha sviluppato, nell'ambito del programma *NLPWin* [19], dei correttori grammaticali più complessi dei precedenti, utilizzati poi, per la prima volta, in *Microsoft Word 97*. Questi software erano in grado di effettuare l'analisi lessicale (che include il riconoscimento di token), la segmentazione delle parole e l'analisi morfologica della frase. Allo stesso tempo il programma consulta dei dizionari integrati nel programma (*dictionary lookup*) e dei tesauri

che contengono le relazioni tra parole suddivise in categorie e sottocategorie. Oltre a controllare l'ortografia quindi, il software era in grado di produrre delle descrizioni sintattiche delle frasi inserite e di controllarne la correttezza.

Se l'implementazione dei sistemi *Microsoft Word Grammar Checker* continua dagli anni '90, ben più rapido è stato lo sviluppo dei sistemi Google (azienda nata nel '98) per l'analisi del linguaggio naturale, che hanno avuto, e continuano ad avere, un ruolo chiave nell'indicizzazione del web da parte del motore di ricerca primo al mondo. In particolare dal 2006 è stato lanciato il servizio *Google Translate* che oggi vanta di poter fornire traduzioni istantanee tra 57 lingue differenti, e che fa uso di un metodo chiamato *statistical machine translation* [20]. In pratica Google elabora le traduzioni analizzando, in modo statistico, i pattern contenuti all'interno di milioni di documenti, per decidere quale sia la traduzione migliore, senza basarsi unicamente su regole fisse e predeterminate; l'unico limite è che le frasi nelle lingue diverse dall'inglese devono essere prima tradotte in inglese e poi tradotte nuovamente nella lingua desiderata.

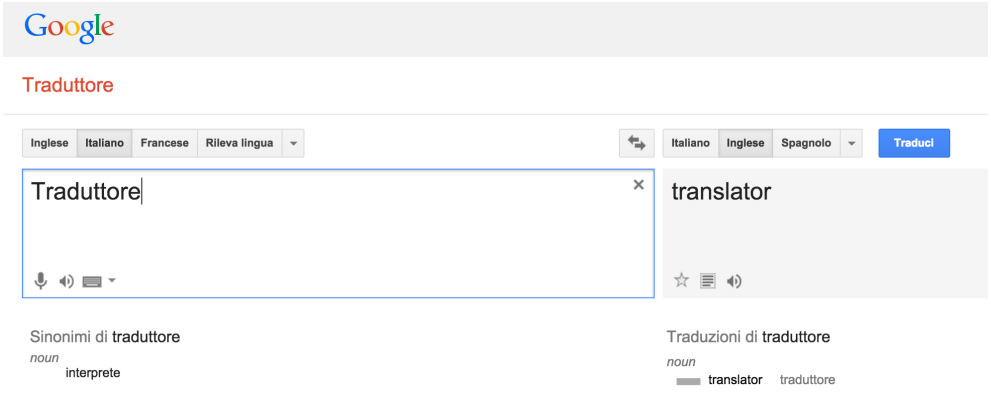


Figura 2.2 - Google Traslate permette di tradurre istantaneamente 57 lingue differenti, proponendo diverse possibili traduzioni e analizzando anche la struttura della frase e il suo significato.

Giungendo in epoca attuale, gli sviluppi che senza dubbio hanno avuto maggior risonanza mediatica e una più ampia popolarità tra gli utenti di tutto il mondo, sono quelli legati agli assistenti artificiali installati nei dispositivi mobile. In particolare Apple con *Siri* e Google con *Google Now*, sono andati ben oltre al semplice controllo vocale, già presente da diversi anni nei dispositivi, con il quale era possibile svolgere semplici compiti come avviare una chiamata o leggere un messaggio: hanno creato dei segretari digitali in grado di rispondere a domande complesse poste in forme differenti, di fare ricerche sul web, dotati persino di una personalità e di umorismo.

Prendendo come esempio Apple Siri, una delle maggiori innovazioni è che per funzionare vengono utilizzate tecnologie cloud: quando viene posta una domanda, per prima cosa il dispositivo analizza il testo per capire se si tratta di un compito eseguibile in locale, come ad esempio avviare una chiamata o fare partire una canzone, oppure se si tratta di compiti più complessi da gestire in remoto. In caso di richieste articolate, la frase, o meglio la scomposizione della frase pre-elaborata (che mira a mantenere intatto il significato della richiesta dell'utente raccogliendo le feature principali), viene inviata ai server Apple, elaborata ulteriormente e poi rispedita al dispositivo sotto forma di istruzioni. A questo punto il compito può essere svolto facilmente anche ricorrendo a numerosi servizi web.

Il cuore della tecnologia consiste nel fatto che la computazione necessaria a comprendere la richiesta e ad eseguire il compito non è tutta a carico del dispositivo singolo, ma viene svolta in remoto, con il vantaggio di velocizzare il processo che non grava più sulle risorse limitate del telefono o del tablet. Inoltre questo sistema permette di apprendere dalle richieste di milioni di utenti, sfruttando tecniche statistiche e di *machine learning*, migliorando con il tempo l'esperienza utente.

La progettazione del sistema si basa su tecnologie ed idee sviluppate inizialmente nell'ambito di un progetto militare internazionale partito nel 2003 e finanziato dalla *Defense Advanced Research Projects Agency* (DARPA) , del dipartimento americano della difesa e portato avanti dalla *SRI International* (una associazione di ricerca affiliata alla Stanford University), col fine di creare un assistente (o meglio, un *cognitive assistant*) in grado di apprendere dal comportamento degli utenti e da un enorme quantità di dati [21]. L'evoluzione di Siri rispetto ai sistemi precedenti di riconoscimento del linguaggio sta proprio nel fatto che il sistema non si basa unicamente sulla segmentazione della frase, ma tratta i concetti come entità vere e proprie, cercando di individuarne il contesto di appartenenza.

Per quanto riguarda l'ambito mobile in realtà lo scopo di queste applicazioni non è più solamente quello di facilitare alcuni compiti quando si guida o si è impossibilitati alla scrittura, ma di indurre gli utenti ad utilizzare gli assistenti anche in altre occasioni, per divertimento o semplicemente guidati dalla curiosità di conoscere la risposta ad una domanda complicata, scopi che rientrano più nella sfera dell'intrattenimento.

Consultando le statistiche sui primi anni di utilizzo di Apple Siri infatti, sembrerebbero proprio queste ultime le ragioni per cui l'assistente viene maggiormente utilizzato, insieme a semplici task come ad esempio l'invio di messaggi o le chiamate, funzioni già presenti nei suoi predecessori: un uso piuttosto limitato rispetto alle potenzialità previste inizialmente [1]. Inoltre ci si potrebbe domandare se un tale sistema, seppur avanzato, possa realmente funzionare anche in altri ambiti più complessi e magari meno legati al puro intrattenimento, come quello medico, assistenziale, per la gestione delle emergenze, e così via.

In tali casi la risposta deve essere istantanea anche se posta in condizioni sfavorevoli e la principale difficoltà è dettata dal fatto che il robot, oltre a

rispondere ad una domanda, deve eseguire un compito legato al movimento, come ad esempio spostarsi da una posizione ad un'altra, afferrare un oggetto o svolgere un'operazione manuale. Questo introduce tutta una serie di difficoltà ulteriori che, come vedremo ora, rende i sistemi esistenti piuttosto inadeguati alle reali necessità di questi ambiti, in cui il progresso delle tecnologie di riconoscimento del linguaggio naturale ha portato sino ad ora a risultati molto limitati.

2.2.3 L'uso dell'interfaccia uomo-macchina in medicina

Un campo in cui il progresso della robotica sarebbe auspicabile è quello ospedaliero, dove sono molto frequenti situazioni in cui sarebbe necessario l'intervento di un robot (o di un sistema di intelligenza artificiale) per svolgere alcuni compiti, come assistenza agli anziani, supporto nelle operazioni chirurgiche, gestione database e altro ancora.

Ad accomunare tutti questi casi è la difficoltà per l'operatore umano di accedere ad una tastiera: l'utilizzatore potrebbe essere un anziano o un malato non in grado di muoversi, o che comunque non ha mai utilizzato strumenti tecnologici; oppure un medico che durante un operazione difficilmente può digitare su una tastiera, considerate anche tutte le problematiche legate all'igiene. Appare chiaro quindi che l'avere un'interfaccia basata unicamente sul linguaggio naturale possa essere provvidenziale e straordinariamente efficace in questi casi.

I ricercatori stanno attualmente sperimentando diversi tipi di robot ospedalieri (detti *Hygeiorobot*) che comunicano con l'utente tramite *Spoken Dialogues Systems* (SDS) [22]. Prima di tutto questi sistemi utilizzano un riconoscitore audio (*speech recognizer*) per convertire le parole dell'utente in forma scritta. Poi analizzano la frase tramite un analizzatore linguistico, costruendo una rappresentazione logica della stessa, da utilizzare nell'elaborazione. Una volta recepito il task da eseguire, un gestore del dialogo interno decide se contattare altre applicazioni esterne o se il robot può eseguire il compito da solo. In tal caso viene inviato un messaggio di risposta all'utente dal controller del robot.

Su questo schema generale sono stati implementati diversi tipi di robot, utilizzati in campi differenti. Ad esempio *RHINO* è una guida per musei che riconosce semplici frasi ed è in grado di descrivere gli oggetti basandosi su *landmarks* presenti in ogni posizione [22]. Allo stesso modo è stato sviluppato *MAIA*, un robot in grado di trasportare oggetti da un luogo ad un altro in un ufficio, o *AESOP*, che è una linea di robot più complessi utilizzati in chirurgia e controllati dal medico tramite la voce [22].

Figura 2.3 - I robot ospedalieri permettono di operare in sicurezza, evitando molti problemi igienici e incrementando la precisione degli interventi. Questi robot possono essere controllati anche da remoto permettendo ad un chirurgo, specializzato in un determinato ambito, di operare persino in una località diversa da quella dove si trova in quel momento.

2.2.4 Il problema dell'ambiguità nella traduzione del linguaggio naturale in linguaggio formale e sue soluzioni

I problemi che sorgono negli ambiti appena descritti sono numerosi e costituiscono un esempio molto significativo in quanto forniscono un'idea generale sulle complessità da affrontare. Prima tra tutte c'è la difficoltà di tradurre la frase pronunciata dall'utente nella forma testuale, separando correttamente le parole (*word discovery*). Pur riuscendo a costruire un sistema resistente al rumore e ai disturbi audio, non sempre la segmentazione del discorso è univoca e potrebbero sorgere casi di ambiguità. Questi casi sono dovuti non solo alle tecnologie di elaborazione audio (che possono essere migliorate nel tempo) e al fatto che una stessa parola può essere pronunciata in molti modi differenti, ma anche ai problemi teorici visti in precedenza e ben più difficili da risolvere, legati al fatto che il linguaggio naturale non è completamente formalizzabile. Alle ambiguità acustiche dunque, si aggiungono quelle sintattiche.

Per risolvere questi problemi si cerca di equipaggiare i robot con un contesto, ossia una semantica che gli permetta di risolvere le ambiguità; questa semantica è fornita solitamente in modo astratto, tramite trascrizioni e *labeling* dei dati da parte dei programmatori che inseriscono in questo modo una conoscenza innata sugli oggetti del mondo fisico all'interno del robot, utilizzando ad esempio le ontologie concettuali, che sono informazioni strutturate sul mondo reale in una forma immediatamente comprensibile dal computer.

Su questo modello, per cercare di superare le limitazioni illustrate nel paragrafo precedente, è stato realizzato *AINI* che utilizza *Dragon Naturally Speaking* per processare e riconoscere il linguaggio naturale [23]; questo software è uno dei più utilizzati e più accurati nella NLP e possiede un'enorme collezione di vocaboli e la capacità di apprendere nuove parole. Il robot, per aumentare la sua conoscenza sul contesto, sfrutta anche un sistema che tramite riprese video cerca di riconoscere, dall'espressione dell'utente di fronte, il suo stato d'animo, così da regolare le reazioni. La conoscenza che il robot sviluppa nel tempo viene raccolta sotto forma di pattern salvati in un linguaggio di mockup (simile all'XML).

Altri studi mirano a risolvere il problema della traduzione dell'inglese in linguaggio formale, puntando a realizzare un controller (scritto in un linguaggio simile al LISP) in grado di effettuare un parsing semantico della frase, estraendo quindi informazioni sul contesto [24].

Un'altra soluzione degna di nota, proposta recentemente, è quella che mira a ridurre la complessità e l'ambiguità del linguaggio naturale direttamente all'origine, fornendo comandi precisi al robot tramite il *microblogging* [25]. Si è visto infatti che nei *tweet* (i messaggi scambiati su Twitter), che hanno una lunghezza massima limitata a 144 caratteri, si tende a ridurre notevolmente i costrutti e ad esporre i concetti in modo chiaro e non ambiguo. In questo modo, che per gli utenti comuni appare comunque piuttosto naturale, si semplifica il processo di traduzione del linguaggio, senza contare che i social network esistenti costituiscono dei dataset enormi su cui effettuare il training dei sistemi. Naturalmente la limitazione è che il sistema vale solo per la scrittura e non è applicabile, senza forzature, alla lingua parlata.

La possibilità di tradurre rapidamente il linguaggio naturale in linguaggio formale comprensibile agli agenti artificiali è di grande rilevanza anche nell'ambito della comunicazione tra robot stessi, in cui sarebbe utilizzato in modo bidirezionale. Si parla in questo caso di *Multi Agent System (MAS),* in cui i diversi attori comunicano direttamente, si scambiano informazioni per eseguire al meglio determinati compiti, senza l'intervento diretto dell'uomo [26]. Far comunicare tali sistemi in un linguaggio condiviso, specificando i protocolli di interazione e soprattutto rendendo esplicito il significato della comunicazione porterebbe tuttavia enormi benefici dal punto di vista del controllo della conformità e della sicurezza. Tale linguaggio sarebbe indipendente dall'implementazione e dai dettagli operazionali interni e permetterebbe una verifica efficace dei processi di interazione tra sistemi differenti.

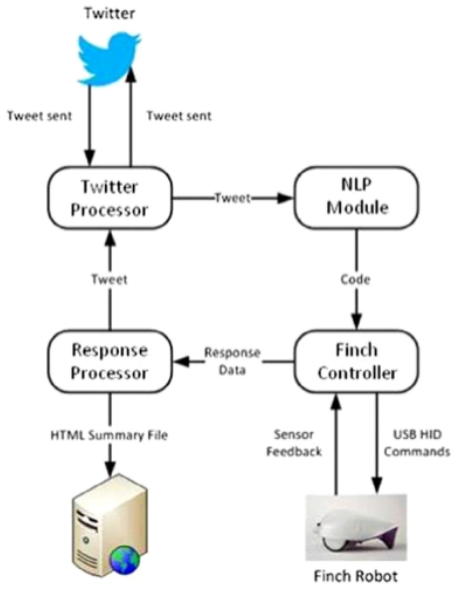


Figura 2.4 - Nell'immagine è mostrato lo schema di funzionamento di un automa che riceve comandi tramite Twitter, uno dei social network più diffusi, che permette agli utenti di scambiarsi dei "tweet", che sono brevi messaggi da 144 caratteri che in poco tempo possono fare il giro del mondo grazie al meccanismo delle condivisioni (o "retweeting"). La limitazione sulla lunghezza dei messaggi costringe gli utenti ad essere molto concisi ed utilizzare frasi dalla struttura semplificata, che possono essere facilmente analizzati da una macchina.

2.3 Primi passi verso il linguaggio naturale in robotica cognitiva

Le soluzioni esposte sembrano semplificare i problemi legati all'uso del linguaggio naturale nei sistemi artificiali, tuttavia non rappresentano risoluzioni definitive al problema generale.

Esse si basano infatti sull'ipotesi che in determinati ambiti non serva un'interazione approfondita tra l'uomo e la macchina, ma basti che l'automa comprenda poche frasi precise e sia esperto in quel compito specifico: in questo modo si semplifica molto la creazione del robot. Questa visione è tipica della corrente che considera il robot unicamente come uno strumento da utilizzare per determinati compiti e si discosta molto dall'idea che la robotica possa seguire la strada verso un *human-friendly computing*, in cui l'automa possa apparire, agli occhi dell'utilizzatore, come un umano artificiale [23].

2.3.1 L'importanza di un'interazione naturale uomo-macchina e i problemi attuali

L'intenzione di sviluppare un'interazione approfondita tra uomo e macchina, basata sulla comprensione del linguaggio naturale, non deve tuttavia apparire soltanto come il frutto di un romanticismo poco pragmatico, derivante dalle molteplici novelle di fantascienza che negli anni si sono susseguite.

Ci sono numerose motivazioni, tutte rigorosamente scientifiche, che spingono verso un approccio di questo genere.

Prima di tutto si possono trovare numerosi esempi in cui un'interazione poco naturale con la macchina possa risultare svantaggiosa per chi l'utilizza: una modalità limitata di chiedere le cose e poche decine di frasi a disposizione possono inibire la conversazione con la macchina già nel caso di utilizzatori malati e anziani che cercano assistenza o di utenti che non sono stati edotti, precedentemente, sulle modalità di utilizzo.

In situazioni di emergenza, come ad esempio in missioni di salvataggio tramite elicotteri o in ambienti difficilmente raggiungibili, il robot può trovarsi ad interagire con persone non esperte o con soggetti impauriti o feriti, che difficilmente sono in grado di utilizzare un linguaggio chiaro e conciso: in questi casi l'uomo non può adattarsi alle esigenze della macchina.

Allo stesso modo, in caso di macchine mobili, una persona potrebbe trovarsi a dover specificare la propria posizione o il percorso da seguire per raggiungere una meta. In questi casi è importante che il sistema linguistico sia collegato al sistema motorio e sia in grado di selezionare le coordinate spaziali corrette relative a quel determinato contesto [29].

La dimostrazione che tali sistemi non funzionano in situazioni sfavorevoli per il robot si sono avute in molti casi pratici quando, distaccandosi dalla simulazione in laboratorio e facendo operare il robot sul campo, non si sono ottenuti i risultati sperati. Ad esempio, durante il disastro nucleare di Fokushima, i robot giapponesi hanno fallito nella gran parte dei tentativi di operare sul campo, per motivi legati sia alle condizioni difficili in cui dovevano operare (alti livelli radioattivi, di umidità o calore), sia alla loro scarsa abilità di prendere decisioni, che non è stata all'altezza della situazione; anche il controllo remoto svolto dagli operatori sul campo è risultato piuttosto difficoltoso, mostrando quanto la tecnologia, nonostante i molteplici passi avanti, sia del tutto impreparata ad affrontare situazioni non previste [27].

Inoltre, anche dal punto di vista della progettazione, i metodi illustrati in precedenza presentano numerosi problemi, infatti l'implementazione è *hard-coded* e molto complessa, necessita di un livello molto alto di astrazione e richiede tecniche di *path planning* sviluppate ad hoc in base a compiti specifici. Le applicazioni di questo tipo sono basate su un linguaggio altamente logico di implementazione che richiede di progettare numerose

regole di azione e che impedisce dunque l'utilizzo di linguaggio naturale, se non nelle sue forme più basilari.

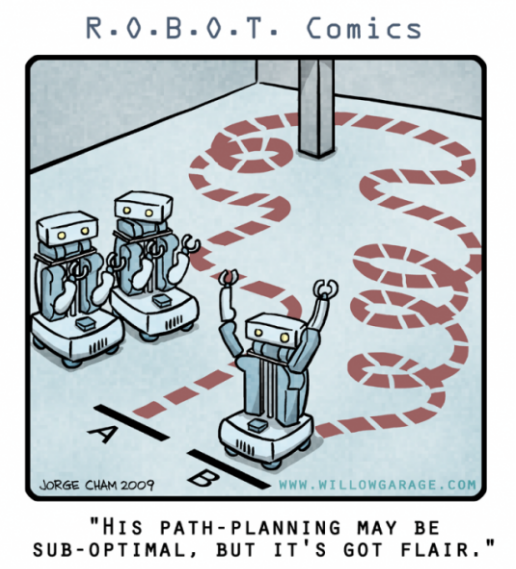


Figura 2.5 - "La sua pianificazione del percorso può essere non ottimale ma ha talento artistico."

2.3.2 Le esperienze sensoriali come fondamento del significato

Viste le ragioni pratiche che spingono ad evitare un approccio che consideri l'agente artificiale unicamente come uno strumento dedicato ad un determinato compito e incapace di utilizzare un linguaggio naturale complesso, riportiamo altre considerazioni a livello teorico che ulteriormente ci spronano a percorrere questa strada.

Per iniziare si può considerare un esempio semplice ma significativo: si provi ad immaginare una banana verde. Quasi certamente l'immagine affiorata nella mente rappresenta una banana poco matura, di un verde chiaro, come se ne trovano comunemente nei negozi di frutta e non una banana verde scuro, di una tonalità innaturale, di cui non abbiamo mai avuto esperienza. Questo accade perché per la comprensione del linguaggio, che sia scritto o parlato, ci basiamo moltissimo sulla nostra conoscenza pregressa [13].

Fondiamo le parole, gli aggettivi, i concetti sulle nostre esperienze sensoriali, sugli input che nel corso della nostra vita abbiamo raccolto ed immagazzinato nella memoria. In questo modo i concetti mentali incontrano il loro corrispondente nel mondo reale, perché ognuno di essi è collegato al ricordo di una determinata esperienza sensoriale.

E' questo probabilmente il modo che il cervello utilizza per bypassare il *Symbol Grounding Problem* [13] . La stessa cosa avviene se si pensa ad una

ragazza con i capelli rossi: ognuno ha in mente una tonalità differente di capelli, che può variare dall'arancione scuro al rosso acceso. Anche qui l'immagine mentale è legata all'esperienza tramite la quale è stata etichettata: si tratta dell'*object labeling* di cui abbiamo parlato all'inizio, che mette in relazione una parola ed un'esperienza sensoriale.

Si evince quindi quanto sia importante e allo stesso tempo difficile superare questo problema nei sistemi informatici e quanto sia limitata dal punto di vista funzionale la soluzione di inserire manualmente la conoscenza semantica nella macchina tramite labeling o tecniche simili, come quelle che abbiamo analizzato in precedenza. Proprio da qui nasce l'esigenza di tentare un approccio differente, che permetta di collegare una parola appresa dall'agente artificiale al suo significato fisico, di connettere un input sensoriale ad un concetto memorizzato nella macchina, senza la necessità di inserirlo manualmente sotto forma di regole logiche formali, che non risolvono in modo efficace il problema e conducono a quanto visto nell'esempio della Stanza Cinese di Searle [14].

Lo stesso discorso vale per i movimenti, che restano astratti per il robot e totalmente scollegati (*ungrounded*) rispetto ai comandi espressi nel linguaggio naturale e quindi, visto che il linguaggio è profondamente collegato all'organizzazione della mente, al sistema cognitivo generale dell'agente artificiale.

Capitolo 3

Scelta progettuale: perché utilizzare un approccio bioispirato?

"Un'idea un concetto un'idea
finché resta un'idea è soltanto un'astrazione
se potessi mangiare un'idea
avrei fatto la mia rivoluzione"

Giorgio Gaber da "Un'Idea"

3.1 Risoluzione di problemi complessi imitando la natura

Come visto nella sezione precedente, i primi approcci all'analisi del linguaggio, basati sull'inserimento di un gran numero di regole formali all'interno della macchina, non hanno portato i risultati sperati.

Questa situazione ricorda molto quanto accaduto, sempre nel campo della robotica, nell'ambito della meccanica, in cui si studiano i dispositivi, i materiali e le strutture tramite le quali costruire un corpo adeguato al robot.

Anche in questo caso, in particolare per quanto riguarda la progettazione di robot umanoidi, gli ingegneri si sono dovuti scontrare con enormi difficoltà tecniche, nella ricerca di parti meccaniche adeguate che rispondessero a stringenti requisiti per ciò che concerne dimensioni, peso, capacità di movimento. Basti pensare alla difficoltà di racchiudere il sistema di intelligenza artificiale, necessario per un robot ad operare, nello spazio limitato del suo corpo, o della sua testa, il che porta con sé tutta una serie di problematiche relative al surriscaldamento dei componenti.

Problemi simili sono quelli che riguardano l'alimentazione, perché la batteria necessaria per consentire al robot di avere un'autonomia ragionevole potrebbe essere troppo pesante da essere trasportata dall'automa. Un altro esempio di progettazione estremamente complessa riguarda la possibilità di inserire numerosi motori nello spazio minuscolo della mano del robot, per renderlo in grado di afferrare e manipolare oggetti, quesito molto studiato anche nel campo delle protesi meccaniche.

Per affrontare tutti questi problemi ingegneristici di difficile soluzione, una buona strada da percorrere è quella della robotica bioispirata, che fa parte della più ampia categoria del design bioispirato.

L'idea alla base di tale approccio è quella di studiare le strutture biologiche degli esseri viventi per cercare strade efficienti da seguire per risolvere il tipo di problemi esposti in precedenza: piuttosto che ricercare nuove soluzioni partendo da zero dunque, conviene ispirarsi a quelle già trovate dalla natura e, in particolare, dall'evoluzione, nei milioni di anni di lavoro ed ottimizzazione che ha avuto a disposizione [29].

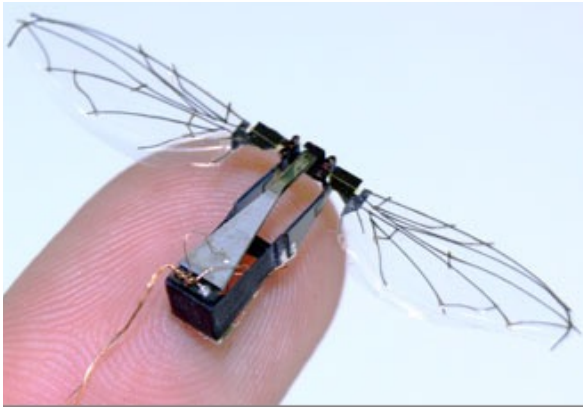


Figura 3.1 - L'approccio bioispirato riutilizza e si ispira alle soluzioni implementate dalla natura nel corso dell'evoluzione per risolvere problemi ingegneristici complessi.

Dal punto di vista computazionale infatti, l'evoluzione può essere considerata come un complesso e potente strumento per la risoluzione di problemi, dove il codice genetico rappresenta il linguaggio con il quale i problemi sono scritti e le soluzioni prodotte.

In realtà l'idea di ispirarsi alla natura per disegnare oggetti o risolvere problemi è tutt'altro che nuova e non è stato necessario scoprire i complessi meccanismi alla base del funzionamento del DNA per avere esempi concreti: basti pensare all'importanza che ha avuto lo studio del volo e delle ali degli uccelli nella creazione dei primi mezzi in grado di volare. Leonardo da Vinci, già nel '400, osservava a lungo la natura e i suoi elementi, cercando di ispirarsi ad essi nella costruzione delle sue macchine. Probabilmente, sin dalle sue origini l'uomo ha preso in prestito dalla natura soluzioni efficaci per quanto riguarda la difesa (punte e frecce sono ispirate alle corna e ai denti degli animali), ha utilizzato le pellicce degli animali per isolarsi dal freddo quando qualsiasi materiale in grado di fornire un simile riparo era ben lontano dall'essere inventato.

La robotica bioispirata dunque cerca di studiare e apprendere dai sistemi biologici, così da applicare le soluzioni implementate dalla natura nella

progettazione dei sistemi artificiali, imitando ad esempio il movimento di cani, insetti o persino dei gechi per permettere ai robot di spostarsi in modo efficiente su superfici scivolose o in ambienti difficili da raggiungere [29].

In realtà ispirarsi alla natura non coincide con copiare completamente i sistemi naturali (in tal caso si dovrebbe parlare di biomimetica), ma piuttosto imparare dalla natura, producendo meccanismi più semplici o addirittura più efficienti rispetto a quelli esistenti. Un esempio è lo studio del funzionamento dei neuroni nel cervello che ha portato allo sviluppo dell'idea di percettrone prima e delle reti neurali poi, che prendono spunto, ma sono ben diverse, da quelle realmente presenti nel cervello biologico.

Figura 3.2 - Nella figura è mostrato un robot in grado di arrampicarsi su numerosi tipi di superficie sfruttando gli stessi meccanismi che utilizzano i gechi in natura. Si tratta di un design profondamente bioispirato.

3.1.1 L'embodiment e lo sviluppo di capacità cognitive tramite l'interazione di mente, corpo ed ambiente

La disciplina della robotica bioispirata non esaurisce la sua utilità nel solo ambito del movimento e in quello dell'ottimizzazione delle parti meccaniche del robot: negli ultimi anni sta portando i ricercatori ad assumere un punto di vista nuovo per quanto riguarda lo sviluppo delle capacità cognitive dei robot, non più basato, come nei modelli costruiti nei decenni passati, sull'inserimento di basi di conoscenza contenenti assiomi e di regole formali da elaborare per

ottenerne di nuove, ma su uno sviluppo *bottom-up* dell'intelligenza a partire dalle esperienze sensoriali dell'ambiente circostante, che il robot sperimenta tramite un corpo.

Per chiarire questa idea bisogna introdurre il concetto di *embodiment* [30], che sarà poi ripreso successivamente quando parleremo dello sviluppo del linguaggio nei bambini. Questa idea si basa sull'ipotesi che l'intelligenza necessiti di un corpo e di un ambiente per emergere: la conoscenza non deve essere fornita direttamente al robot come un set di regole e assiomi astratti, ma deve essere sviluppata dalla macchina stessa interagendo con l'ambiente circostante tramite un corpo. Anche negli esseri umani le capacità cognitive si sviluppano, a partire dai primi anni di vita, grazie ai feedback ricevuti dall'ambiente: molto prima di imparare a manipolare gli oggetti, il bambino inizia a stringerli, a spostarli, a toccarli, in modo involontario. Lo stesso vale per gli altri sensi: il bambino inizia a vedere prima ancora di comprendere quello che sta vedendo, o a sentire quando ancora non ha sviluppato il linguaggio. L'atto di toccare, sentire, vedere, precede la capacità di farlo in modo volontario e cosciente, ma è grazie ad esso e ai feedback ricevuti, ossia agli stimoli sensoriali ricevuti dall'esterno, che il bambino impara ad interagire con il mondo, a comprendere quello che sta facendo sviluppando la propria mente fino a raggiungere una volontà cosciente.

Il tutto è possibile naturalmente se si ha a disposizione, oltre che una *mente* (ad esempio un'intelligenza artificiale), un *corpo*, composto da particolari materiali con una certa morfologia, e un *ambiente*.

Lo stesso vale anche in ambito biologico: nelle ultime ricerche di neurobiologia si tende in modo crescente a studiare come le funzioni celebrali siano *embedded* nel cervello, come il sistema fisico interagisca con il mondo reale esterno, piuttosto che considerarne unicamente il funzionamento astratto e teorico.

Questo porta a conseguenze molto profonde per quanto riguarda la relazione tra elaborazione dell'informazione da parte della mente e l'ambiente fisico: la *morfologia* e i *materiali* con i quali sono costituiti i corpi potrebbero addirittura assolvere ad alcune funzioni solitamente attribuite al cervello stesso, tanto che si parla di *morphological computation* [30].

La particolare forma del corpo dunque, durante il movimento e l'interazione con il mondo reale, potrebbe addirittura facilitare l'apprendimento e la percezione: essa aiuta a configurare la struttura informativa dei dati ricevuti dal cervello riducendo la complessità generale, processo che è difficile comprendere appieno se ricercato unicamente all'interno del cervello e che è impossibile da riprodurre nelle macchine basandosi unicamente sul software [31].

Riassumendo, sono due le idee principali che emergono dalla discussione precedente: la prima è che il robot deve avere un *ruolo attivo* nel processo di apprendimento e di sviluppo cognitivo, la seconda è che per farlo bisogna

fornire al robot un *corpo* ed un *ambiente* in cui operare, tramite il quale accumulare *esperienze* sul mondo.

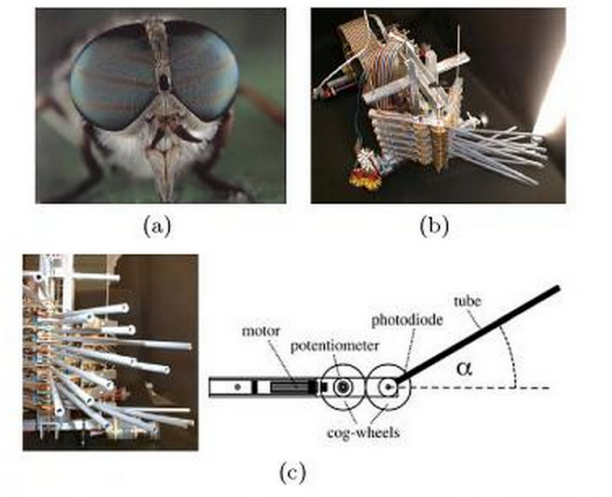


Figura 3.3 - La figura mostra un'esempio di computazione morfologica tramite l'utilizzo di sensori: la loro disposizione non omogenea permette di compensare il disturbo derivante dal movimento e la distorsione del parallasse.

a) Insetto presente in natura

b) Conformazione dell'insetto robot

c) Funzionamento dei sensori che si basano sulla conformazione dell'occhio dell'insetto reale

3.2 Le primitive motorie

Analizziamo ora un concetto molto studiato nella robotica bioispirata, che ci porterà ad introdurre, nella sezione seguente, l'idea di un linguaggio bioispirato: le primitive motorie.

Questa idea si basa sul reale funzionamento del cervello biologico per quanto riguarda l'apprendimento dei movimenti: le prime volte che si compie un movimento più o meno complesso, si deve pensare attentamente a quali muscoli muovere, si devono pesare gli impulsi inviati ad ogni estremità e bisogna valutare attentamente i feedback ricevuti dall'apparato sensoriale, così da capire se la traiettoria che stiamo seguendo è corretta o se va modificata. Tuttavia, compiendo lo stesso movimento più volte e in condizioni simili, non sarà più necessaria una concentrazione così elevata e nel tempo, continuando a ripeterlo, esso apparirà sempre più naturale, tanto che alla fine si potranno aggiungere altri movimenti, complicare l'operazione, senza troppa fatica. Questo è alla base di ogni tipo di apprendimento motorio: inizialmente il bambino gattona poi, nel tempo, percorre brevi tratti in piedi, fino a camminare perfettamente. Lo stesso accade se si inizia a suonare uno strumento: si parte da semplici movimenti che corrispondono a determinate note, poi si inizia a collegare le note tra di loro, a variare i tempi tra una e

l'altra, fino ad arrivare a comporre una melodia. Come vedremo in seguito, anche il linguaggio si basa su dinamiche simili.

Questo processo di apprendimento si può schematizzare perfettamente tramite le primitive motorie: esse possono essere considerate come l'unità di base del controllo volontario [32].

Ogni muscolo, infatti, viene attivato da un impulso nervoso proveniente da un motoneurone: per attivare più muscoli, occorrono altrettanti impulsi. In base alla frequenza dell'impulso nervoso il muscolo si comprime con maggiore o minore forza, fino ad arrivare alla compressione massima chiamata tetano.

Per compiere qualsiasi movimento, anche il più semplice, come ad esempio distendere un braccio, occorre coordinare numerosi muscoli e quindi trasmettergli molteplici impulsi. Ogni impulso è una primitiva motoria. Ripetendo più volte il movimento tuttavia, il sistema nervoso (non distinguiamo, per ora, tra quello centrale e periferico) è in grado di apprendere il movimento, formato dall'insieme delle primitive motorie utilizzate per quel compito, e registrarne l'andamento generale sotto forma di funzione di attivazione.

Successivamente, per compiere lo stesso movimento, non sarà più necessario attivare singolarmente ciascuna primitiva motoria, ma basterà ripetere la funzione generale che a questo punto è diventata, a sua volta, una nuova primitiva motoria.

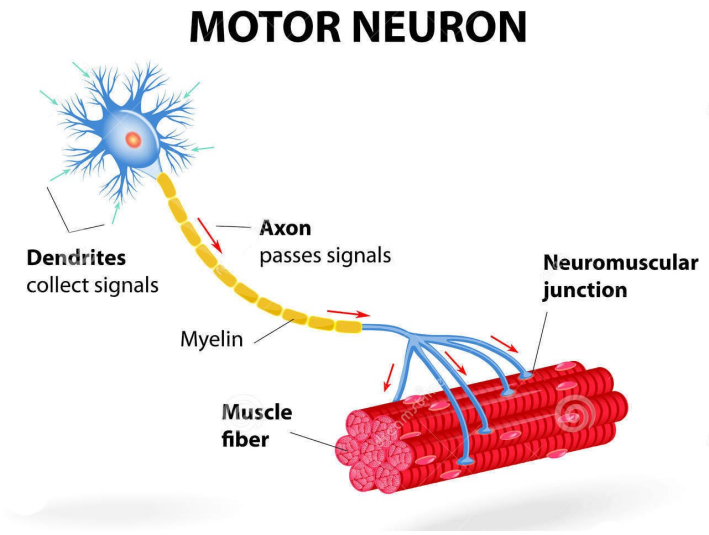


Figura 3.4 - La figura mostra il funzionamento di un moto-neurone generico: i dendriti ricevono in input i segnali dagli altri neuroni, che vengono elaborati e passati, attraverso l'assone del neurone, ai muscoli. L'assone è ricoperto di mielina, una particolare sostanza isolante che accelera la trasmissione e la protegge da disturbi. La fibra muscolare riceve l'impulso tramite le giunzioni neuro-muscolari e si contrae, finché l'impulso non termina.

La stessa cosa accade con l'operazione di afferrare una palla con la mano: dopo molte prove, i molteplici impulsi diretti a numerosi muscoli, che corrispondono ad altrettante primitive motorie, andranno a formare una nuova, unica primitiva motoria.

A questo punto unire due primitive motorie (come ad esempio quella di distendere il braccio e quella di afferrare la palla) sarà un compito piuttosto semplice, perché si tratta soltanto di elaborare una nuova primitiva motoria che unisca entrambe e non calcolare un impulso di attivazione per ogni singolo muscolo coinvolto nel movimento.

Questo approccio che modellizza il funzionamento del cervello biologico è stato molto studiato in robotica, perché rappresenta un modo molto efficace per gestire il movimento dei robot ispirandosi alla natura anziché sviluppando modelli complessi basati su centinaia di coordinate motorie: basta far memorizzare all'automa, di volta in volta, nuovi movimenti, che possono in seguito essere fusi in un unico nuovo movimento.

3.2.1 Dai riflessi involontari ai movimenti volontari

Le primitive motorie spesso vengono associate ai riflessi, tuttavia, se pur condividono meccanismi simili, sono concettualmente differenti: i secondi sono movimenti involontari che coinvolgono gruppi di nervi e che non possono in alcun modo essere attivati o repressi volontariamente.

L'esempio più intuitivo di riflesso è quello che si ha in seguito ad un pericolo: se entra qualcosa nell'occhio, la palpebra si abbasserà molto prima che il sistema centrale diventi cosciente di cosa sia accaduto e risponda con un impulso volontario; stessa cosa accade se si tocca qualcosa che scotta: il braccio si ritrarrà spontaneamente, coordinando diversi muscoli nell'operazione senza coinvolgere, perlomeno inizialmente, il sistema centrale.

In realtà, in base agli studi, i riflessi iniziano ad operare molto prima che esista un vero e proprio controllo volontario: già nell'utero i bambini cominciano a muovere le mani e i piedi, persino la bocca, azioni che sono frutto di riflessi involontari [33]. Stessa cosa accade subito dopo la nascita, quando i movimenti delle mani o della testa dei bambini che tendono ad esplorare il mondo sono perlopiù movimenti provenienti da riflessi involontari: il bambino non muove coscientemente le proprie articolazioni.

Negli ultimi anni si è visto che questi riflessi costituiscono un passo fondamentale verso l'apprendimento dei movimenti volontari: essi permettono al bambino di esplorare le proprie possibilità motorie, di ricevere dei feedback dall'ambiente attraverso i quali conoscere forme, materiali e soprattutto di mappare mentalmente ogni movimento del suo corpo. Tramite i riflessi il bambino, ricevendo sensazioni sia dall'interno del corpo (propriocezione), che dall'esterno (tatto, stimoli visivi ed uditivi), riesce a creare una relazione tra

azione del suo corpo e reazione dell'ambiente circostante a quel movimento (mapping senso-motorio) [32].

La creazione di questa mappa mentale lo porterà, nei mesi successivi, a sviluppare movimenti volontari che imitano i movimenti inizialmente eseguiti in modo istintivo. Allo stesso tempo, in contemporanea allo sviluppo mentale, avviene nel bambino uno sviluppo fisico: i nervi si ricoprono di mielina, una particolare sostanza isolante che oltre a rendere i collegamenti più rapidi, riduce il disturbo degli impulsi, diminuendo i riflessi involontari che scompaiono progressivamente riducendosi ai soli presenti anche negli adulti, i quali si attivano in risposta a situazioni di emergenza.

La riduzione progressiva dei riflessi involontari corrisponde ad uno sviluppo, altrettanto progressivo, della capacità di eseguire movimenti volontari: dunque i primi consentono ai secondi di emergere e perfezionarsi. In molti muscoli infatti, (soprattutto in giovane età, ma anche negli adulti), coesistono stimoli del sistema nervoso centrale e stimoli provenienti dai riflessi e proprio questa cooperazione porta al corretto movimento.

L'idea secondo la quale dai movimenti spontanei dei bambini si originano i movimenti volontari è molto importante sia dal punto di vista degli studi medici, che da quello della robotica bioispirata: tale argomentazione sarà trattata in modo più approfondito quando parleremo del *babbling*, il balbettio involontario dei bambini appena nati, da cui si origina, a poco a poco, la capacità di parlare.

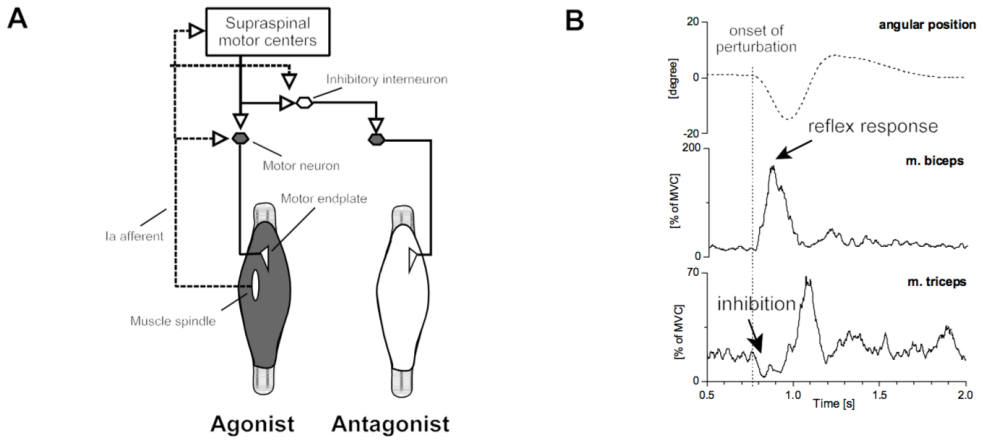


Figura 3.5 - Le figure mostrano il percorso di attivazione di due muscoli antagonisti nel momento in cui uno stimolo produce un riflesso involontario. Nella figura A viene mostrato il collegamento neurale tra i due muscoli: l'attivazione del muscolo agonista inibisce automaticamente l'antagonista. Ciò può essere visto anche nella figura B dal punto di vista dei segnali elettrici generati dai muscoli e analizzati attraverso l'elettromiografia.

3.2.2 Il Central Pattern Generator

Un altro concetto da tenere in considerazione consiste nel fatto che, anche in età adulta, non tutti gli stimoli motori provengono unicamente dal cervello. Anzi moltissimi movimenti, soprattutto quelli ripetuti che seguono un certo pattern (come ad esempio camminare), sono principalmente gestiti dal dal midollo spinale.

In questi casi si parla di *Central Pattern Generator* (CPG), ossia una rete di neuroni che da sola riesce ad originare e gestire azioni ritmiche, senza bisogno dell'analisi dei feedback sensoriali da parte del cervello, che invece si attiva solamente in risposta a situazioni insolite o a variazioni volontarie del ritmo [34].

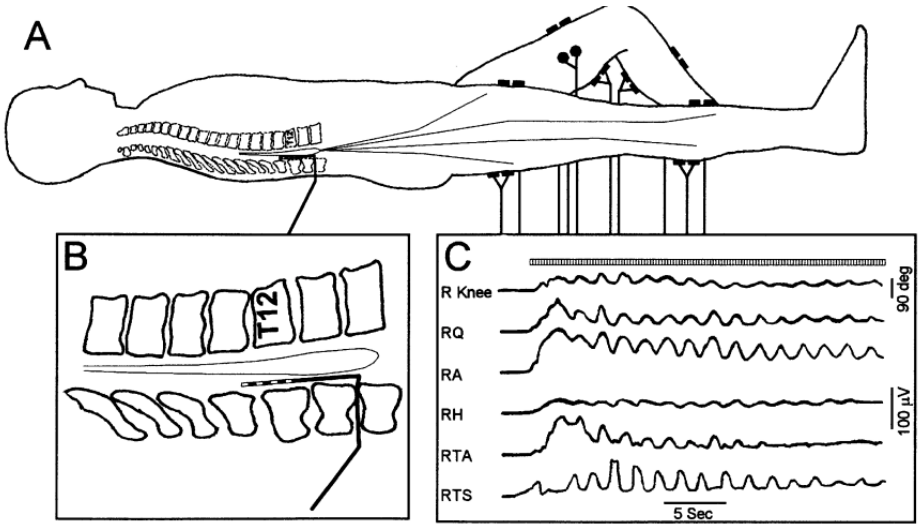


Figura 3.6 - Esperimento effettuato su un uomo adulto con lo scopo di valutare il funzionamento del CPG (A). Applicando uno stimolo nella colonna vertebrale (B), nel canale dietro la struttura posteriore lombare si ottiene, come effetto, un movimento ritmico regolare degli arti inferiori. Nella figura C vengono mostrati i diversi pattern di onde rilevate in diversi punti del corpo.

Questo concetto si ricollega al discorso sull'*embodiment*, perché una conseguenza positiva di considerare il corpo nel modello computazionale di un robot, è proprio la possibilità di distribuire la computazione in più centri, senza caricare un solo sistema centrale dell'onere della gestione di qualsiasi movimento, esattamente come viene fatto nei sistemi biologici.

3.3 Verso un modello bioispirato di linguaggio

Nel corso della sezione precedente abbiamo analizzato nei dettagli un particolare approccio ingegneristico definito bioispirato in quanto prende spunto da conoscenze sul funzionamento di sistemi biologici per risolvere diversi tipi di problemi in robotica e in altre discipline. Questo approccio è già impiegato da diversi anni nello studio del movimento dei robot, in particolare con la creazione di strutture che richiamano il corpo di animali reali in grado di sopravvivere in determinati ambienti.

La letteratura inoltre tratta ampiamente il concetto di primitive motorie, che sono già utilizzate in diversi progetti. Tuttavia lo stesso non si può dire per quanto concerne il funzionamento della mente del robot e del linguaggio, oggetto di questa tesi: soltanto negli ultimi anni, infatti, si è cominciato a spostare l'attenzione dalla progettazione di sistemi meccanici bioispirati, a modelli cognitivi bioispirati.

Sembra esserci un interessante parallelismo tra la robotica artificiale e la psicologia: gli psicologi alla fine del diciannovesimo secolo e inizio ventesimo erano principalmente interessati ai comportamenti motori e alla percezione dei pazienti studiati, mentre negli ultimi decenni del '900 si sono concentrati maggiormente sulle abilità cognitive e sul funzionamento della mente [32]. Allo stesso modo gli ingegneri impegnati nella costruzione di robot si sono concentrati inizialmente sul movimento e sullo sviluppo di sensori in grado di imitare in modo efficace i sensi umani; in seguito, solo negli ultimi anni, hanno cominciato ad interessarsi allo sviluppo di modelli mentali in grado di rendere i robot capaci di prendere decisioni o sviluppare motivazioni in modo autonomo.

3.3.1 Perchè utilizzare un linguaggio bioispirato

Per quanto riguarda la comprensione e l'attitudine al linguaggio, gli studi sono ancora più arretrati: come visto nelle sezioni precedenti, al di fuori di poche applicazioni perlopiù ludiche riguardanti la sfera dell'intrattenimento, sono scarse le applicazioni pratiche dell'utilizzo del linguaggio naturale nella robotica [35].

In precedenza abbiamo visto che i problemi principali sono essenzialmente due: il *Symbol Grounding Problem* e la traduzione del linguaggio naturale in linguaggio formale utilizzato dai robot e dai sistemi informatici in genere; entrambi conducono allo stesso problema, rappresentato dalla mancanza di una semantica legata al linguaggio e quindi di un contesto. I sistemi esistenti infatti non sono *grounded*, ossia non basano il loro significato su oggetti reali: le rappresentazioni semantiche inserite nei sistemi di intelligenza artificiale sono invece specificate manualmente, ed anche il training delle macchine è effettuato tramite registrazioni trascritte manualmente e label predefiniti.

Questo tipo di semantica astratta, che isola completamente le macchine dal mondo reale, assume significato soltanto se interpretata da un essere umano.

Infatti abbiamo visto che negli esseri umani il linguaggio, e quindi la sintassi e la semantica, emergono dall'osservazione del mondo tramite i sensi e associando prima le parole ad oggetti reali, inserendole in contesti, e poi mettendole in relazione tra di loro.

La conseguenza di questi problemi non ancora del tutto risolti è che nel mondo accademico il linguaggio è stato sempre studiato in modo isolato dagli altri sistemi del robot, tramite notazioni formali e simboliche mentre, negli esseri umani, come avremo modo di analizzare più approfonditamente nella prossima sezione, esso non evolve come sistema isolato, bensì perfettamente integrato con tutto il resto: dai movimenti, alle esperienze sensoriali, allo sviluppo cognitivo della mente [36].

Questo ci porta a dire che così come la mente, anche il linguaggio va studiato in una prospettiva *embodied* e, possibilmente, bioispirata, perché potrebbe essere una strada per superare, perlomeno dal punto di vista pratico, il *Symbol Grounding Problem*.

3.3.2 L'approccio grounded: fondare il linguaggio sulle esperienze senso-motorie

Quando un uomo legge una frase, per comprenderla utilizza la propria conoscenza, non solo per quanto riguarda la grammatica e la sintassi, ma riconosce il contesto e sfrutta le sue nozioni in merito a quel contesto per capirla fino in fondo [37]. In molti casi, in tutti se si escludono i concetti astratti, l'uomo si basa sulle conoscenze acquisite attraverso gli stimoli sensoriali, sulle esperienze senso-motorie o sulle rappresentazioni mentali delle sue azioni e dell'ambiente reale. Anche i concetti astratti sono basati su i concetti iniziali legati ad oggetti del mondo fisico e poi messi in relazione tra di loro e generalizzati in idee teoriche.

Per fornire al robot una tale capacità di comprensione bisognerebbe collegare in qualche modo il linguaggio naturale umano con le percezioni del robot: si dovrebbe quindi fondare la comprensione del linguaggio naturale umano da parte dell'automa non sulla semantica dell'uomo, ma sulle esperienze senso-motorie del robot stesso [38].

Solo legando al linguaggio naturale le percezioni che l'agente ottiene interagendo con l'ambiente circostante tramite un corpo si farebbe un'enorme passo avanti. Esattamente come avviene nei bambini quando per la prima volta fanno esperienza del mondo.

La conoscenza del robot non sarebbe più legata a regole astratte fornite dall'uomo, ma sarebbe sviluppata dall'agente stesso interagendo con il mondo e dunque sarebbe fondata sulle sue stesse esperienze, il che permetterebbe di superare anche il problema della traduzione del linguaggio naturale in linguaggio formale, perché non ci sarebbe più la necessità di trasformare la conoscenza dell'uomo in una comprensibile alla macchina, dato che è essa

stessa in grado di sviluppare un proprio bagaglio di conoscenze, una propria semantica.

Riassumendo, da questo approccio che chiameremo *grounded*, possono essere ottenuti due vantaggi principali [35]. Prima di tutto il problema dell'apprendimento del linguaggio sarebbe risolto senza utilizzare il labeling dei dati o altre tecniche affini di caricamento manuale della conoscenza, perché sarebbe svolto dal robot stesso, in grado a quel punto di estrarre indizi dal contesto delle frasi ascoltate e valutarli in relazione al bagaglio di conoscenza acquisita dall'apparato senso-motorio.

Inoltre, avendo un contesto sul quale basarsi e dunque un insieme di informazioni non linguistiche da sfruttare nel processo di comprensione, sarebbe risolto anche il problema dell'ambiguità, che come abbiamo detto nelle sezioni precedenti ha una doppia valenza: dal punto di vista uditivo la difficoltà di riconoscere e segmentare le frasi, dal punto di vista sintattico la difficoltà di comprendere l'organizzazione interna delle frasi che potrebbero essere interpretate in diversi modi.

E' evidente come un tale approccio al linguaggio risulti bioispirato: esso si basa infatti sul concetto di embodiment e segue un processo cognitivo simile a quello dei sistemi biologici. Allo stesso tempo, si può notare come un qualsiasi approccio bioispirato al linguaggio non possa fare a meno di essere *grounded*: tutti gli studi di neurobiologia si soffermano sul profondo legame tra linguaggio, esperienze sensoriali ed azioni.

Come già visto, anche analizzando il cervello da un punto di vista biologico, si vede che aree coinvolte nel movimento o nella percezione vengono attivate durante il task di comprensione del linguaggio perché permettono di assegnare un significato, fondato sull'esperienza, alle parole, ossia ai simboli astratti che costituiscono il linguaggio stesso [39].

Il processo di formare nuovi significati (riguardanti oggetti di cui non si ha esperienza diretta) a partire da parole base (di cui si ha esperienza diretta) si chiama *symbol grounding transfer*, ed è un metodo grazie al quale, ad esempio, unendo le parole "strisce" e "cavallo", si da vita al concetto di "zebra".

L'insieme di questi termini di base è detto *grounding kernel*, in cui ogni simbolo è collegato con un'esperienza senso-motoria, ed è la base dalla quale, tramite i processi di elaborazione linguistica visti in precedenza, si ottiene tutto il lessico e i rispettivi concetti.

Nel cervello biologico, per ottenere tali processi linguistici, non c'è bisogno di prevedere regole sintattiche esplicite, dato che essi sono conseguenza della capacità intrinseca della rete neurale di processare i simboli (*symbol composition capability*) grazie agli stimoli ricevuti: il linguaggio infatti può essere definito come una caratteristica emergente del cervello, che si sviluppa grazie all'interazione con l'ambiente circostante [39].

Per concludere questa sezione, in cui si è vista l'importanza di adottare una prospettiva bioispirata per il linguaggio, prima di andare a studiare come

muoversi dal punto di vista pratico, è bene fare alcune considerazioni finali sulle possibilità offerte a lungo termine da questo approccio.

Come abbiamo visto fino ad ora, studiare il linguaggio è fondamentale dato il suo contributo significativo all'organizzazione interna della mente: se c'è un link così profondo tra i due, vuol dire che adottare un approccio *grounded* per il linguaggio porta ad estendere una simile proprietà anche alle altre capacità cognitive dell'automa, coinvolgendo tutte le sotto-categorie dell'intelligenza artificiale.

Vedendo il tutto da una prospettiva più ampia, le implicazioni a lungo termine di un linguaggio grounded includono la possibilità di fornire agli automi la capacità di acquisire autonomamente conoscenza sul mondo, tramite l'apprendimento e l'elaborazione del linguaggio naturale ascoltato, integrando ad esso la conoscenza sulle azioni effettuate e gli stimoli sensoriali ricevuti. In ultima analisi dunque, riprendendo quanto detto nella prima sezione, un piccolo sviluppo nel campo del linguaggio naturale bioispirato corrisponderebbe, con ogni probabilità, ad un enorme passo avanti nello sviluppo delle capacità cognitive degli automi e nella loro capacità di prendere decisioni [36].

Capitolo 4

Dallo sviluppo del linguaggio nei bambini alla definizione di un modello

"Con molta attenzione AC [Automatic Calculator] organizzò il programma. La coscienza di AC inglobava tutto ciò che un tempo era stato un Universo, e rifletteva su quello che adesso era il Caos. Passo a passo, doveva venir fatto.
E AC disse: «SIA LA LUCE!»
E la luce fu…"

Asimov da "L'Ultima Domanda"

Abbiamo concluso la sezione precedente proponendo un approccio bioispirato al problema del linguaggio naturale nei robot, che nel futuro potrebbe essere in grado di superare, perlomeno dal punto di vista pratico, il *Symbol Grounding Problem* e portare a miglioramenti consistenti nell'intero modello dei sistemi cognitivi artificiali.

Naturalmente questa strada avrà bisogno di numerosi studi ed esperimenti prima di diventare percorribile e lo scopo di questo elaborato non è sicuramente quello di proporre un modello definitivo, dato che le sperimentazioni in questo campo sono solo agli esordi, ma di cercare un punto di partenza valido per approfondire la ricerca.

In primo luogo analizzeremo i concetti base che riguardano l'evoluzione del linguaggio nei bambini, dai primi mesi di vita fino ai primi anni. In questo modo si chiariranno i meccanismi più importanti grazie ai quali, da un punto di vista neurobiologico, ha origine la capacità di parlare e comprendere ciò che si ascolta negli esseri umani.

Tra questi sarà data particolare rilevanza al babbling, ossia il balbettio dei bambini nei primi mesi di vita, che funge da precursore alla parola. In seguito ci baseremo proprio su questo meccanismo per iniziare a costruire un modello di apprendimento del linguaggio da utilizzare in robotica. Infine prendendo spunto da esperimenti già svolti in questo campo si proseguirà nella ricerca approfondendone alcuni aspetti utili.

4.1 Lo sviluppo del linguaggio nei bambini

4.1.1 Aspetti multisensoriali e neurali

La maggior complessità dal punto di vista biologico legata alla parola è data dal fatto che la comprensione e l'utilizzo del linguaggio è un processo multi-sensoriale. Questo vuol dire che per studiarlo non possiamo prendere in considerazione un unico senso, ma è indispensabile coinvolgere anche gli apparati visivi, motori, propriocettivi.

Naturalmente per udire un suono generico si utilizzano unicamente le orecchie, ma per comprendere le parole, secondo molteplici studi, viene utilizzata anche la vista, così da captare i numerosi segnali provenienti dalle espressioni del viso, dai movimenti del corpo e della bocca del nostro interlocutore o per associare una forma ad una parola. In seguito, dopo aver appreso la lettura, la stessa parola scritta diventa un'etichetta per il suono: è dimostrato che associando parole scritte diversamente a suoni distinti questi saranno discriminati con maggior facilità, anche se molto simili[40].

Altri esperimenti hanno mostrato che, facendo vedere a dei bambini di pochi mesi due video registrati, ciascuno con il volto di una persona che parla e facendogli sentire l'audio di una sola delle due, i bambini saranno maggiormente attratti dal video di chi pronuncia effettivamente le parole. Non c'è dunque bisogno di saper leggere il linguaggio labiale o di comprendere effettivamente il significato delle parole per poter ottenere degli indizi visivi dal nostro interlocutore. Allo stesso modo, per produrre il linguaggio, è necessario articolare le parole, un processo che coinvolge, tra gli altri, anche l'apparato motorio e propriocettivo della bocca.

Dal punto di vista cerebrale, durante l'ascolto sono coinvolte, oltre alle aree specializzate nel linguaggio (ad esempio l'area di Broca), anche parti della corteccia, dell'area visiva e di quella motoria; quest'ultima è di ausilio per la comprensione delle parole legate al movimento [41]. Dunque il contesto del discorso viene estratto, oltre che dall'organizzazione e dal contenuto della frase, dalle circostanze fisiche nelle quali viene pronunciato.

Nei primi mesi di vita, quando ancora si è ben lontani dal saper pronunciare, o soltanto rilevare, le prime parole, si è invece già in grado di determinare i contrasti fonetici tra le parole. Questo fenomeno porta al *phonetic attunement* e cioè ad una maggiore sensibilizzazione dell'udito verso i contrasti e i toni specifici di un determinato linguaggio; già nel corso dei primi mesi quindi, si diventa esperti nel rilevare i fonemi specifici della propria lingua madre, mentre si perde a poco a poco la capacità di captare contrasti statisticamente meno frequenti [41]. Questa maggior sensibilità ai suoni tipici della propria lingua permetterà, a poco a poco, di cominciare a distinguere le prime parole.

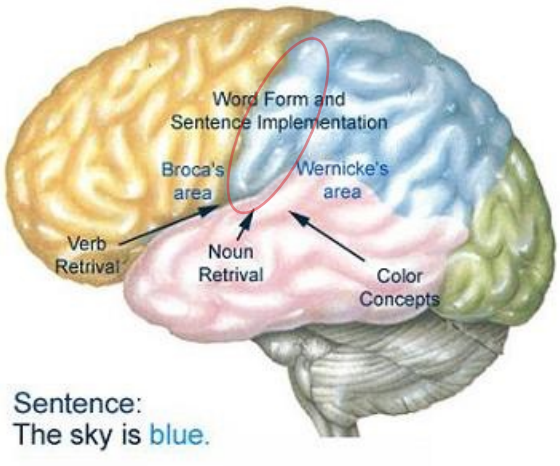


Figura 4.1 - Per elaborare il significato di una frase ("il cielo è blu") sono coinvolte numerose aree. Le aree di Broca e di Wernicke sono i centri del linguaggio, maggiormente coinvolti nell'elaborazione delle frasi, ma utilizzano diverse aree circostanti per recuperare i dati relativi al significato. In rosso è evidenziata la corteccia motoria primaria che viene spesso coinvolta nei processi di comprensione delle frasi, in particolare in quelle che contengono concetti riguardanti i movimenti.

Allo stesso tempo, in modo naturalmente inconscio, il bambino comincia a rilevare i *pattern* ripetuti del linguaggio tramite un approccio perlopiù statistico, ossia basato sulla frequenza degli stessi toni nelle frasi, indipendemente dallo sviluppo del lessico: in questo modo non c'è bisogno di segmentare sin dall'inizio la frase, ma basta ascoltare i suoni per poter cominciare a creare, a livello mentale, le prime categorie fonetiche, tramite le quali distinguere successivamente le parole.

Questo è un metodo molto efficiente dato che, come abbiamo visto, il compito di segmentare una frase è molto complesso non solo per le macchine, ma anche per i bambini, poiché il suono che ha una parola pronunciata singolarmente è molto differente dal suono che assume in una frase, quando il discorso fluisce velocemente e i confini tra i termini sono più sfumati.

Questo approccio statistico fa parte di un meccanismo più generale che si chiama *distributional learning* ed è all'origine dell'apprendimento del linguaggio: le informazioni relative alle distribuzioni delle frequenze dei toni vengono unite agli altri indizi visivi (ad esempio oggetti o il movimento delle labbra), contribuendo a creare un contesto al discorso.

Ad esempio, se una parola di cui inizialmente si apprende unicamente il tono viene pronunciata più volte in un dato contesto (Contesto A) e un'altra parola viene invece utilizzata più spesso in un secondo contesto (Contesto B), le due parole saranno memorizzate come differenti l'una dall'altra, al di là della conformazione sintattica delle frasi e dall'ortografia delle parole stesse, semplicemente perché associate a contesti differenti, in modo simile a quanto si fa con l'*object labeling* [41].

Grazie a questa abilità il bambino inizia non solo a creare diverse categorie fonetiche, ma riesce a rilevare dei *pattern* regolari e ripetuti in diverse parole, generalizzando la regola ed applicandola a diversi termini: essi infatti sono in grado di rilevare e tracciare le informazioni statistiche relative a sequenze regolari (ad esempio la probabilità che X predice Y). Questa capacità in realtà non viene sfruttata solamente nel linguaggio, ma anche in altre forme di apprendimento, come quello visivo. Secondo alcuni esperimenti infatti un bambino di sette mesi riesce a rilevare e generalizzare regole sia se queste si riferiscono a sequenze di suoni (ad esempio ABA e ABB), sia se si riferiscono a forme, come ad esempio sequenze di animali (cani e gatti) [9]. Questo dimostra, ancora una volta, che nell'apprendimento e nello sviluppo linguistico vengono utilizzate regole ed abilità non specificatamente legate al linguaggio stesso.

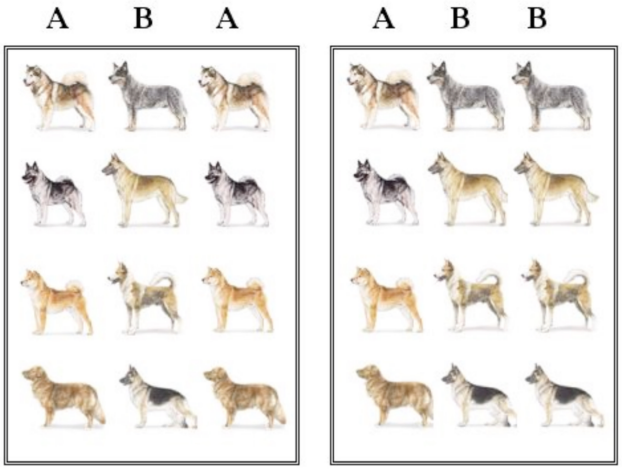


Figura 4.2 - Tabelle utilizzate nell'esperimento che valuta l'abilità dei bambini di generare regole indipendentemente dal linguaggio utilizzato e dai simboli che rappresentano ciascuna regola.

Un altro meccanismo che conferma l'importanza di stimoli multi-modali nell'apprendimento del linguaggio è quello legato all'utilizzo dei *neuroni specchio* [39]. Questo particolare tipo di neuroni rappresenta una classe che si attiva quando un uomo osserva un'azione compiuta da un altro suo simile, e sembra fondamentale per moltissimi processi mentali in quanto spinge il cervello a simulare l'esecuzione dell'azione osservata.

E' lo stesso meccanismo per cui, se si guarda una persona sbadigliare, si sentirà lo stimolo di farlo.

Tornando ai bambini, si è notato con diversi esperimenti che sentire qualcuno parlare e guardarlo in viso, stimola negli infanti il movimento di labbra e bocca, constatando una notevole risposta a livello neurale nei casi in

cui il movimento delle labbra dei bambini era simile a quello dell'adulto che pronunciava la vocale ascoltata [42].

Questa capacità non viene persa nell'età adulta, durante la quale i neuroni specchio restano attivi e anzi sono necessari per lo sviluppo di altre tecniche di simulazione come il *corollary discharge*, un segnale neurale generato dal sistema dei neuroni motori per distinguere le sensazioni interne alla mente e quelle causate invece da eventi e stimoli esterni [43]. Questo meccanismo è alla base della voce mentale che ci sembra di sentire quando pensiamo e che è centrale nel nostro pensiero cosciente: essa è prodotta dalla simulazione, da parte della nostra mente, del suono che avrebbero le parole pensate se pronunciate realmente dalla nostra bocca e dunque udite esternamente.

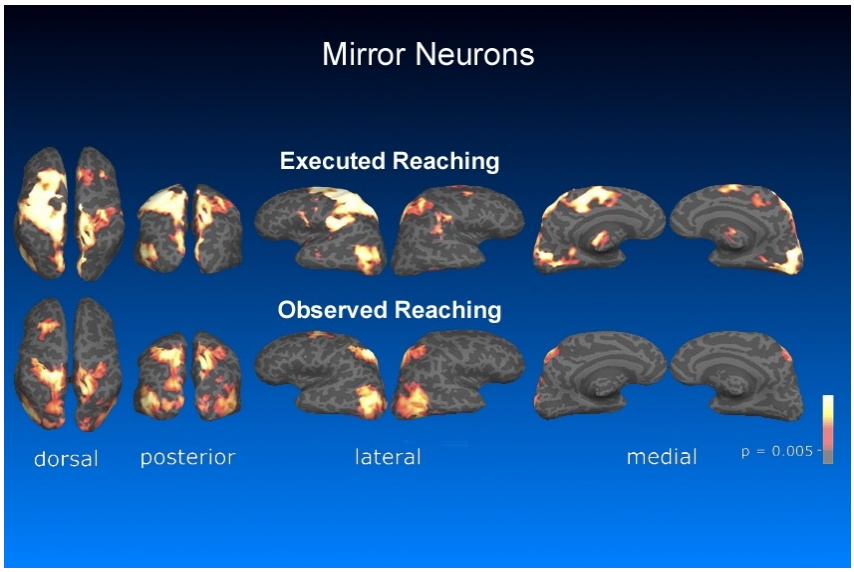


Figura 4.3 - Nell'immagine è mostrata l'attivazione a livello cerebrale dei neuroni specchio, confrontando le stesse aree impegnate prima nell'esecuzione di un compito e poi nell'osservazione dello stesso. Come si può vedere i neuroni specchio si attivano quando osservano un'azione compiuta da qualcun altro, in modo simile a come si attiverebbero se l'azione fosse effettivamente svolta dall'organismo di cui fanno parte.

4.1.2 Apprendimento linguistico e sistema senso-motorio

Ancora una volta si può notare quindi come la capacità di parlare sia profondamente integrata nella mente e collegata ai meccanismi motori e sensoriali. D'altra parte questo meccanismo dimostra anche che le sensazioni sono profondamente influenzate dall'elaborazione mentale degli input esterni, e così come le sensazioni conseguenti da un'azione sono anticipate mentalmente, allo stesso modo le parole attese in un determinato contesto possono influenzare la nostra percezione degli stimoli uditivi. Se in un determinato punto della frase ci aspettiamo di sentire una parola, ad un suono ambiguo attribuiremo automaticamente quella parola, senza analizzare nel

dettaglio il suono: è un altro metodo per ridurre la complessità nella segmentazione e nel riconoscimento vocale, basandosi sul contesto.

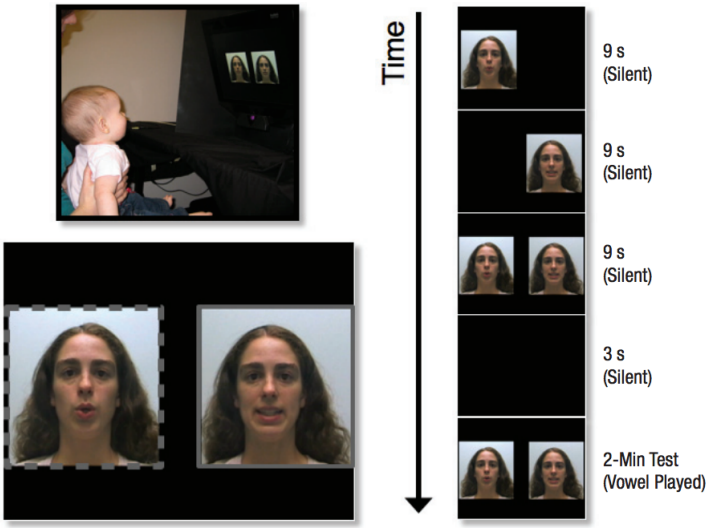


Figura 4.4 - Nell'esperimento vengono mostrate due registrazioni di una persona che parla e produce diverse vocali articolandole in modo evidente con la bocca. I bambini, sottoposti al video, tendono a mantenere maggiore attenzione verso il video quando sentono anche l'audio. Se entrambe le registrazioni vengono mostrate contemporaneamente, ma soltanto l'audio di una delle due può essere udito dal bambino, questo mostrerà maggior interesse per il video di cui sta sentendo effettivamente le parole.

Lo sviluppo della percezione nei riguardi di toni e contrasti tipici della propria lingua madre, che avviene al di sotto dei 5 mesi di età, apre dunque la strada alle fasi successive nell'apprendimento del linguaggio che sono il riconoscimento di vocali (tra i 6 e gli 8 mesi), delle consonanti (dagli 8 ai 12 mesi) e infine delle durate dei fonemi (intorno ai 18 mesi). Queste fasi si susseguono nel bambino molto velocemente, più di quanto gli input sonori appresi permetterebbero [44].

Questo *gap* tra conoscenze linguistiche acquisite e set di parole ascoltate, può essere superato proprio grazie all'associazione cross-modale, ossia all'utilizzo di diversi stimoli sensoriali nel processo di apprendimento. Ciò vorrebbe dire che se si riuscisse a replicare un tale sistema, sarebbe sufficiente un dataset molto ridotto per effettuare l'addestramento di un robot: questo sarebbe quindi capace di apprendere il linguaggio molto velocemente, integrando ai metodi statistici che valutano le frequenze dei toni metodi più robusti basati sulla relazione funzionale tra referenti nel mondo reale e tipologie fonetiche.

Fino ad ora sono stati analizzati numerosi meccanismi che vengono attuati dai bambini (e in parte anche dagli adulti), durante l'apprendimento del linguaggio.

Il primo concetto fondamentale che possiamo evincere da quanto detto, è che nell'apprendimento non viene assolutamente utilizzato un approccio simbolico, basato su un set di regole fisse, come avviene invece nei robot che abbiamo studiato in precedenza. L'abilità di parlare e comprendere il linguaggio viene sviluppata invece tramite l'esperienza, nel tempo, in modo incrementale ed interattivo [45].

Questa tipologia di apprendimento ricorda molto quella utilizzata nell'acquisizione di abilità motorie che, come abbiamo detto in precedenza, avviene utilizzando le primitive motorie. La similarità tra due approcci non deve stupire: pur trattandosi di due ambiti molto differenti, i meccanismi mentali utilizzati sono sempre gli stessi e c'è bisogno quindi, anche dal punto di vista pratico, di un collegamento profondo tra i due sistemi, dato che per parlare è necessario, oltre che formulare una frase mentalmente, articolarla tramite i muscoli della bocca.

Allo stesso modo le parole ascoltate possono stimolare l'apparato motorio, non solo per quanto riguarda la comprensione, ma anche per ciò che concerne l'azione: se udiamo il nostro interlocutore esclamare, di colpo, "Guarda!", l'azione di sollevare lo sguardo avviene in modo istantaneo, come in un riflesso motorio, ancora prima che la richiesta sia elaborata in modo cosciente ed approfondito dal sistema centrale. Questo avviene perché a quel suono, composto sia da una specifica sequenza di fonemi che da un determinato tono, viene associata una situazione di pericolo, così come il dolore istantaneo a un'articolazione attiva il riflesso di ritrazione dell'arto colpito.

L'idea di una vicinanza così accentuata tra le primitive motorie e il linguaggio sarà ripresa nella prossima sezione quando parleremo del babbling e sarà utilizzata come punto di partenza per studiare un meccanismo bioispirato che permetta di sviluppare il linguaggio in una macchina, partendo da pochi suoni di base.

4.1.3 Universalità dello sviluppo del linguaggio e indipendenza dal contesto

Tuttavia, prima di procedere oltre, conviene spendere alcune parole per definire un altro concetto basilare che ci servirà nell'analisi futura e cioè quello del linguaggio visto come caratteristica innata della mente. Quest'idea sembra andare contro quanto detto fino ad ora, perché abbiamo visto vari esempi che dimostrano come il linguaggio si sviluppi nel tempo grazie a determinati input e utilizzando diversi meccanismi cognitivi.

Ci sono tuttavia altri casi per i quali l'idea del linguaggio basato unicamente su elaborazione degli input ricevuti vacilla.

L'osservazione di partenza fatta da molti studiosi dello sviluppo del linguaggio è che, pur essendoci molte differenze tra un linguaggio e l'altro dal punto di vista dei fonemi e dei contrasti tra suoni, ci sono aspetti molto simili tra una lingua e l'altra, regole sintattiche sviluppatesi allo stesso modo in ceppi

linguistici diversi, che mettono in rilievo come alcuni concetti base dello sviluppo linguistico siano universali per la nostra specie [46].

Anche i significati espressi tramite le strutture sintattiche restano costanti, pur essendo le condizioni di vita molto differenti tra un continente e l'altro; persino tra popolazioni mai venute a contatto le une con le altre è possibile trovare una sintassi condivisa, metodi per farsi capire e tradurre ciò che si vuole esprimere, come se l'evoluzione della lingua avesse subìto un corso comune indipendentemente dai contesti [46].

Anche da un punto di vista neurobiologico è interessante notare come, sotto un'enormità di condizioni ambientali differenti, stili di vita ed esperienze diverse nei bambini, il linguaggio si sviluppi allo stesso modo e all'incirca nello stesso tempo (tra l'altro molto breve), come visto quando abbiamo parlato del *gap* tra input e incremento delle capacità linguistiche.

Ad esempio, dopo il primo anno di vita (in cui abbiamo visto essere presenti varie fasi), quasi tutti i bambini imparano a pronunciare molteplici parole singole, mentre nel secondo anno imparano ad utilizzare correttamente i verbi (più complessi perché richiedono di astrarre un'azione) ed apprendono le posizioni strutturalmente riservate ad ogni tipo di parola. Infine, solo nel quinto anno di vita imparano a gestire adeguatamente frasi multi-causali e particelle di vario tipo. Questo schema emerge in modo costante negli esperimenti effettuati, indipendentemente dal contesto geografico.

Una possibile spiegazione, inizialmente presa in considerazione dagli scienziati, era che le madri parlando con i bambini tendessero ad utilizzare un linguaggio semplice, privo di particelle complesse, guidando l'infante nell'apprendimento iniziale di parole e formule semplici. In realtà, dopo diversi esperimenti si è visto non solo che il linguaggio utilizzato dai genitori, seppur semplificato, conteneva costrutti difficili, effettivamente appresi dal bambino solo intorno ai cinque anni, ma che l'evoluzione del linguaggio da parte del bambino non ne era stato assolutamente danneggiato ed anzi aveva seguito il corso standard anche in specifici casi in cui simili accorgimenti non fossero stati adottati [46].

4.1.4 Apprendimento del linguaggio in caso di deficit fisici

A questo punto è interessante valutare sino a che punto lo sviluppo del linguaggio sia indipendente dal contesto, considerando ad esempio casi di bambini con gravi deficit fisici, in cui gli input linguistici siano stati scarsi o addirittura assenti. Questa curiosità non è in realtà figlia di un'epoca moderna in cui gli studi di psicologia e neurobiologia sono avanzati, ma è già sorta diverse volte nei secoli passati.

Si narra addirittura che un faraone egizio, chiedendosi quale lingua fosse apparsa prima nell'uomo, mise in atto un vero e proprio esperimento in cui due bambini furono isolati, sin dalla nascita, e confinati in luoghi chiusi privi di qualsiasi parola, con l'obbligo che nessuno, pur portando loro da mangiare

ed accudendoli, parlasse ad essi in alcuno modo. A quanto pare, pur senza alcuno stimolo, i due bambini svilupparono comunque una sorta di linguaggio.

Naturalmente ci sono forti dubbi sull'attendibilità scientifica di un tale esperimento, per non parlare della pratica crudele ed inaccettabile dal punto di vista etico; tuttavia la conclusione è estremamente interessante e sembrerebbe confermata, in parte, da studi effettuati in epoca moderna su bambini non udenti [46].

Questa grave disabilità impedisce ai bambini di ricevere qualsiasi stimolo uditivo, ma non intacca gli altri stimoli, come quello visivo. Come abbiamo già detto in precedenza, nell'apprendimento del linguaggio non è coinvolto un solo senso, ma diversi stimoli cooperano insieme per sviluppare l'abilità di parlare e comprendere la lingua. Questo è dimostrato dal fatto che non solo i bambini con disabilità uditive riescono a comprendere molte parole leggendo semplicemente i movimenti della bocca del loro interlocutore, ma sono perfettamente in grado di comunicare tra di loro tramite il linguaggio dei segni, che è strutturato in modo molto simile al linguaggio naturale. I gesti non sono casuali, ma sono composti da un set di primitive gestuali (simili ai suoni di base del linguaggio), organizzate in parole, che a loro volta possono dare vita a frasi, seguendo una precisa sintassi. E' interessante notare ancora una volta come l'idea di primitive gestuali richiami alla mente il concetto di primitive motorie che, come detto in precedenza sono profondamente legate al linguaggio e che, in casi come questi, sono in grado addirittura di sostituire la lingua parlata.

Figura 4.5 - Ci sono differenti tipologie di linguaggio dei segni. Nell'immagine è riportato il linguaggio dei segni che rappresenta l'alfabeto, con il quale si possono comporre parole e frasi. Molte altre tipologie di segni invece sono utilizzate per esprimere direttamente concetti o azioni, e possono essere eseguiti in rapida successione per ottenere frasi complete di sostantivi, aggettivi e verbi.

Ancora più sorprendente è il fatto che, in molti casi in cui al bambino non è stato insegnato il linguaggio dei segni e non è stato quindi mai sottoposto a stimoli di questo genere a causa di pregiudizi nell'ambiente familiare, esso sviluppa autonomamente un linguaggio [46].

Anche in condizioni isolate, a partire dal primo anno di vita, il bambino è in grado di farsi comprendere dai familiari con un insieme di gesti da lui inventati, chiari dal punto di vista del significato grazie alla loro iconicità. Inoltre, analizzando il loro linguaggio, sviluppatosi autonomamente, è possibile rintracciare segni di sintassi e un'organizzazione simile a quella che hanno le frasi nel linguaggio naturale (ad esempio è chiara la distinzione tra "Il pollo mangia" e "Mangia il pollo").

Un caso simile a quello dei bambini con deficit uditivi, è quello in cui gli infanti hanno problemi di vista o sono completamente non vedenti. In una tale situazione ci si potrebbe aspettare di assistere ad un ritardo nell'apprendimento del linguaggio o perlomeno ad una distorsione dato che, come abbiamo visto in precedenza, diverse tecniche di sviluppo del linguaggio coinvolgono la vista.

Sorprendentemente, di rado si assiste ad un tale ritardo, infatti il senso della vista viene sostituito perfettamente, dal punto di vista degli input che concorrono a sviluppare il linguaggio, dal senso del tatto. Questo senso permette ai bambini di ottenere abbastanza stimoli dal mondo esterno da riuscire ad utilizzare, in linea con i tempi di sviluppo normali, il linguaggio naturale.

Questo vale anche per quei casi che sembrerebbero legati in modo indissolubile al senso della vista: abbiamo detto che se sentiamo esclamare dal nostro interlocutore imperativi come "Guarda!" di riflesso alziamo lo sguardo, ancora prima di percepire coscientemente il comando. Nella stessa situazione i bambini non vedenti non alzano lo sguardo (anzi la testa rimane ferma), ma alzano istintivamente le mani, per mettersi in "ascolto" di un input tattile.

Dunque non c'è alcun deficit nell'apprendimento della lingua, piuttosto il significato legato ad alcuni termini viene riadattato, seguendo l'esperienza, ad una particolare condizione: ancora una volta è l'esperienza a fungere da fondamento per il linguaggio.

Questo dimostra nuovamente come il linguaggio naturale sia profondamente legato all'organizzazione mentale tipica della specie umana e, dunque, comune ad ogni suo componente e anche che il processo di apprendimento del linguaggio avviene dall'interno all'esterno, ossia è originato da una naturale predisposizione biologica, che si manifesta insieme allo sviluppo della mente e che matura grazie agli input ricevuti dall'esterno, di qualunque tipo essi siano. Questo modello *inside-out* spiega il *gap* tra input ricevuti dall'esterno e velocità di apprendimento e chiarisce anche perché l'apprendimento del linguaggio segue lo stesso corso anche in condizioni enormemente differenti [46].

Diverso è il caso di infanti con deficit a livello mentale: se il sistema cognitivo non matura correttamente o non si forma adeguatamente, l'abilità di parlare e di comprendere il linguaggio non si svilupperà adeguatamente, indipendentemente dal numero di input ricevuti.

E' evidente dunque, come un problema nel normale sviluppo biologico della mente colpisca il linguaggio in modo più grave rispetto ai deficit che riguardano gli stimoli sensoriali. Naturalmente in casi in cui gli stimoli esterni fossero completamente impediti, come si è studiato in casi di prigionia infantile dovuta a genitori con problemi mentali, o in casi di bambini con deficit gravissimi in diversi sensi (ad esempio sia visivi che uditivi), il linguaggio non potrà svilupparsi in modo adeguato e allo stesso modo lo sviluppo della mente risulterà ritardato e parzialmente impedito [46].

Riassumendo, si è visto come il linguaggio sia una proprietà emergente del cervello, innata dal punto di vista biologico, che si sviluppa nel tempo grazie a numerosi meccanismi, ma che segue un corso comune profondamente legato all'organizzazione e allo stato di maturazione della mente. Come detto disabilità che affliggono la mente danneggiano la capacità di parlare molto più profondamente di deficit nella possibilità di ricevere input, causati da isolamento o da carenze dell'apparato sensoriale. E' chiaro quindi come un sistema biologico in grado di far emergere il linguaggio sia condizione necessaria per l'apprendimento e sia il punto di partenza se si vuole trattare, anche nell'ambito della robotica, il linguaggio naturale.

Segue di conseguenza la necessità di un approccio bioispirato nella costruzione di tale robot che sfrutti il concetto di *embodiment* fornendo all'automa un corpo oltre che una mente, così da renderlo in grado di ottenere stimoli esterni attraverso diversi canali e di rielaborarli mediante diversi sistemi (sensoriali, motorio, cognitivo) tutti profondamente correlati.

Sarà proprio questo il punto di partenza per le sezioni successive, in cui si studierà la possibilità di un modello che permetta di procedere alla realizzazione di un simile approccio.

4.2 Babbling e vocalizzazioni infantili

Nelle sezioni precedenti siamo giunti ad una conclusione di fondamentale importanza: nella ricerca di una strada che porti, nel tempo più breve possibile, ad un utilizzo efficace del linguaggio naturale nei robot, bisogna utilizzare un approccio non simbolico, in contrasto con quanto visto sino ad ora dalla maggior parte delle applicazioni commerciali.

La soluzione che ci è sembrata più adatta è quella di utilizzare gli input sensoriali del robot, dotato naturalmente di un corpo adeguato, come fondamento alla base del significato delle parole e dei concetti che utilizza.

Per avere una conferma anche dal punto di vista biologico, nella sezione precedente si è analizzato lo sviluppo del linguaggio nei bambini, da cui si è

concluso che questa abilità ha una duplice natura: da un lato è una capacità innata del cervello biologico, legata quindi profondamente alla sua struttura, dall'altro ha bisogno di un sistema, un corpo, adatto per svilupparsi correttamente ed emergere tramite numerosi meccanismi di apprendimento.

A questo punto, per iniziare a studiare un possibile modello bioispirato di apprendimento ed utilizzo del linguaggio, bisogna analizzare approfonditamente il meccanismo di base tramite il quale le parole affiorano durante i primi mesi di vita: il babbling [47].

Questo termine in italiano può essere tradotto con "gorgogliare", ed indica i versi fatti dai neonati quando ancora non sono in grado di pronunciare parole complete, che vengono ripetuti in continuazione; un esempio è il suono "baba" o "dada". Se all'inizio sono stati considerati semplicemente come dei versi involontari frutto di un tratto vocale e di una mente poco sviluppati, successivamente a diversi studi si è capito che il babbling riveste un ruolo chiave per il corretto sviluppo della capacità di parlare e in particolare nell'esplorazione delle proprie capacità vocali da parte del bambino.

In realtà i movimenti involontari della bocca del neonato iniziano già prima del parto, così come i quelli degli arti, e sono frutto di riflessi involontari. Durante i primi due mesi di vita, il bambino, oltre a muovere la bocca, emette diversi suoni che sono chiamati *protophones* [47]. Questi sono considerati precursori del linguaggio tipicamente umani perché molto diversi sia dai segnali vocali fissi, come ridere o piangere, che dai suoni vegetativi come starnutire o tossire, i quali sono presenti in molte specie animali.

I *protophones* sono anche definiti *quasivowel*, visto che posseggono già qualche caratteristica delle vocali ed evolvono durante i mesi attraversando varie fasi, finché l'infante non raggiunge, intorno agli 8-12 mesi, il *canonical stage*, in cui inizia il vero e proprio babbling [48].

Così come i riflessi motori, anche i primi suoni emessi dai bambini sono involontari, derivando da particolari movimenti della bocca e delle corde vocali; questa vicinanza non deve stupire dato che, come abbiamo visto durante tutta la nostra trattazione, linguaggio e movimento condividono moltissimi tratti comuni. Tra i meccanismi che si attivano in questi primi mesi rientra anche quello dei neuroni specchio che, come già visto per il movimento, spinge il bambino ad imitare ciò che riceve dagli input visivi ed uditivi.

Dunque, esattamente come avviene per i riflessi motori, a poco a poco i *protophones* smettono di essere suoni involontari e diventano sempre più volontari, ossia dei suoni prodotti intenzionalmente col fine di comunicare [49].

Questa volontà, che ricorda il concetto di intenzionalità espresso da Searle [14], deriva dallo sviluppo cognitivo della mente del bambino che inizia ad assumere proprietà causali, cioè il cervello riesce a rilevare un nesso causale tra due eventi separati. Questo porta quindi all'esigenza di comunicare qualcosa, di attirare l'attenzione per soddisfare i propri bisogni, manifestando i

primi accenni di coscienza. Il bambino infatti, pur non potendo comunicare esattamente una richiesta, può comunicare l'intento di farla [50].

L'altro aspetto da considerare è la volontà di imitare ciò che si vede nel mondo circostante, in particolare ciò che si vede fare agli altri esseri umani, come ad esempio parlare o emettere suoni, il che è già un valido indicatore di intelligenza (non a caso Turing ha scelto l'*Imitation Game* come gioco per valutare l'intelligenza in una macchina [2]). Così come avviene per i movimenti, in queste prime fasi nel cervello del bambino si crea un *mapping* tra movimenti del proprio tratto vocale (che all'inizio avvengono spontaneamente) e suoni risultanti, così da acquisire poco a poco l'abilità di replicare un movimento quando vuole imitare un determinato suono. Questo *mapping* diventa sempre più profondo e dettagliato grazie all'esperienza, fino a portare il bambino alla parola, momento nel quale scompaiono quasi completamente le vocalizzazioni involontarie per lasciare spazio alle parole e dunque ai movimenti volontari del tratto vocale in risposta a determinate esigenze o volontà.

E' interessante vedere che, secondo diversi studi, il babbling è talmente importante che il ritardo o l'assenza di questa fase determina disordini a livello del linguaggio, tanto che molto spesso è utilizzato come strumento di diagnosi nei casi di bambini affetti da malattie o disabilità [47].

A questo punto appare chiaro quanto il babbling e le vocalizzazioni infantili siano importanti per lo sviluppo del linguaggio, ed è proprio per questo profondo legame che abbiamo scelto di partire da questo punto per studiare il nostro modello, anche perché occorre iniziare a considerare le basi biologiche del linguaggio naturale prima di procedere oltre.

Un'altra motivazione che ci spinge a partire dal babbling è la sua profonda somiglianza con il concetto di primitive motorie, dato che le vocalizzazioni infantili appaiono, a questo punto, come l'elemento più prossimo alle primitive del linguaggio che si possa ottenere.

Come detto in precedenza, le primitive motorie sono state già studiate approfonditamente e c'è molta letteratura in merito, al contrario degli studi sullo sviluppo del linguaggio bioispirato che sono invece ai loro esordi. Proprio per questo motivo, trattando le primitive del linguaggio in modo affine a quelle motorie, si potrebbe partire dai metodi e dagli studi fatti su quest'ultime, riadattandoli per il nuovo ambito del linguaggio. Questo punto di vista è corretto anche dal punto di vista biologico ed evolutivo poiché, come abbiamo visto, alcuni dei meccanismi mentali di apprendimento del linguaggio sono gli stessi utilizzati nell'apprendimento del movimento ed entrambi i sistemi comunicano in modo diretto, sostenendosi a vicenda.

Costruire un sistema che funzioni unicamente per il linguaggio e sia separato da tutto il resto non solo non porterebbe ad un modello bioispirato, ma potrebbe generare limiti infrastrutturali difficili da superare, come nel caso dell'integrazione di linguaggio e movimento, senza contare che non permetterebbe di superare in alcun modo il *Symbol Grounding Problem*.

4.3 Human Speechrome Project: la raccolta di dati empirici

Solitamente, prima di poter implementare un modello ispirato alla natura, è necessario studiare approfonditamente il fenomeno, riprodurlo in condizioni controllate e sotto particolari vincoli, per vedere empiricamente come le leggi e i meccanismi da modellizzare agiscano e cooperino.

Nel caso dello studio del linguaggio, se da un lato sono molti gli esperimenti compiuti su soggetti adulti, nel caso di bambini si hanno ben pochi esempi di osservazioni approfondite. Soprattutto nel caso di bambini neonati (8-12 mesi) ottenere osservazioni realistiche in ambienti controllati, come può essere lo studio di un medico, è sempre stato un obiettivo difficile da raggiungere [51].

Per lo studio delle malattie psicologiche e di quelle legate allo sviluppo mentale che affliggono i bambini in tenera età, uno dei metodi più utilizzati è l'intervista ai genitori, in cui si chiede di specificare dettagliatamente le osservazioni fatte durante il tempo passato con l'infante. Spesso viene richiesto alla famiglia di tenere un diario in cui appuntare i progressi, i casi degni di nota o quelli preoccupanti.

La stessa cosa viene fatta per monitorare l'apprendimento del linguaggio: sono i genitori a segnalare eventuali problemi o anomalie, stando a lungo contatto con il bambino nel suo ambiente naturale.

In letteratura si trovano alcuni esempi di osservazioni fatte in laboratorio in cui si sono studiate le varie fasi del linguaggio, ma nessuna di esse permette di chiarire in modo dettagliato come i rapporti con i genitori e con la famiglia permettano al bambino di sviluppare il linguaggio, né di avere la minima idea sul contesto familiare in cui i nuovi termini si sviluppano.

Secondo diversi studiosi, tra i quali Deb Roy, ricercatore presso il MIT, è impossibile ottenere risultati validi senza studiare lo sviluppo del linguaggio in un ambiente naturale, analizzando come le relazioni sociali e il contesto influiscano su di esso. E' seguendo questa idea che è nato lo *Human Speechrome Project*, un progetto portato avanti dal MIT e guidato proprio da Deb Roy che ha permesso di ottenere il più ampio database riguardante la nascita del linguaggio mai ottenuto: per tre anni, grazie ad un sofisticatissimo sistema di computer, telecamere e microfoni, è stato registrato ogni momento della vita di un bambino (il figlio del ricercatore), a partire dalla nascita [51].

Così come l'*Human Genome Project* ha consentito di ottenere la trascrizione dell'intero genoma umano ai fini di studiare lo sviluppo della specie, malattie e meccanismi fisiologici, questo progetto ha l'obiettivo di ottenere la trascrizione dell'intero processo di apprendimento della lingua, partendo dai primi suoni fino ad arrivare a frasi di senso compiuto, per chiarire i meccanismi con i quali il linguaggio viene acquisito ed elaborato.

Per ottenere un simile traguardo numerose telecamere e microfoni sono stati installati nella casa di Deb Roy e hanno permesso di ottenere, nei tre anni

in cui sono stati attivi, circa 200 gigabyte di riprese al giorno. Ogni giorno le registrazioni venivano trasferite nei supercomputer di un laboratorio dedicato dove venivano elaborate e trascritte, estrapolando dati statistici e informazioni di vario tipo.

Il sistema sviluppato appositamente per questo progetto e chiamato, non a caso, *Total Recall*, è all'avanguardia per la capacità di elaborazione di grandi quantità di dati e l'estrazione di feature e statistiche dettagliate. Oltre alla complessità computazionale, un'ulteriore sfida è stato far sì che un metodo di studio così invasivo per l'intimità dei soggetti coinvolti fosse realizzabile nel totale rispetto della privacy e della vita privata.

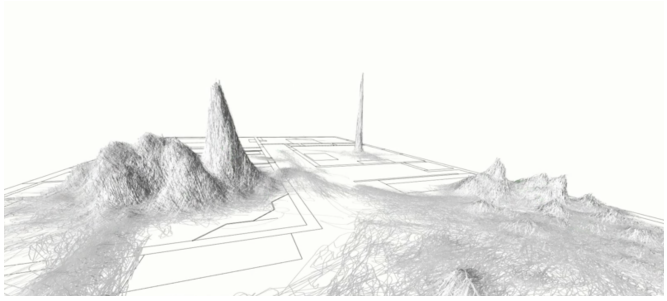


Figura 4.6 - Nell'immagine è mostrata la mappa dei luoghi in cui è stata pronunciata la parola water dal bambino. Sono presenti dei picchi in cucina e nel bagno, dove è presente grande quantità d'acqua.

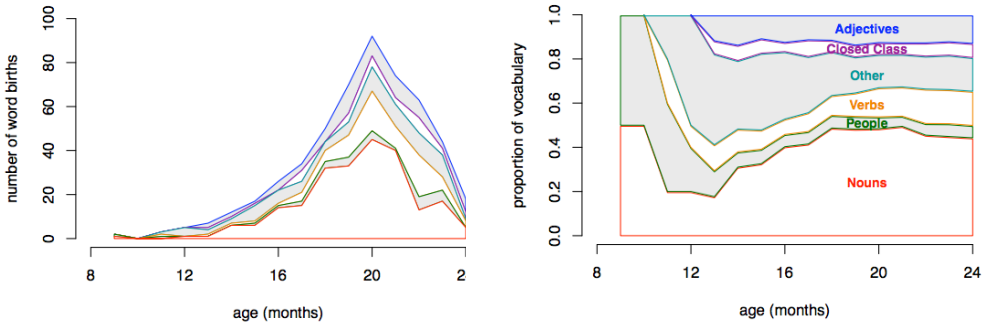
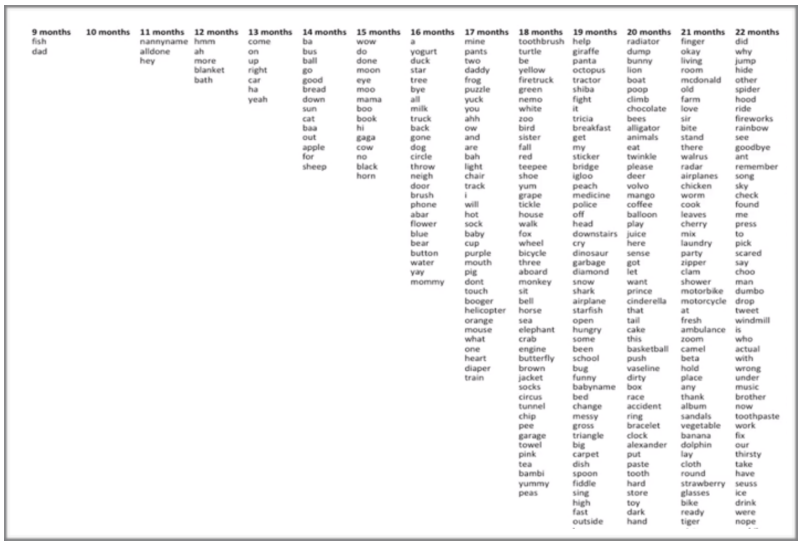


Figura 4.7 - Tipologie di parole apprese. Inizialmente si imparano parole relative ad oggetti, o persone. Sono successivamente si sviluppano i verbi (che implicano di immaginare e gestire un movimento) e in fine gli aggettivi, che specificano meglio gli attributi degli oggetti.



9 months	10 months	11 months	12 months	13 months	14 months	15 months	16 months	17 months	18 months	19 months	20 months	21 months	22 months
fish		nannyname	hmm	come	ba	wow	a	mine	toothbrush	help	radiator	finger	did
dad		alldone	ah	on	bus	do	yogurt	pants	turtle	giraffe	dump	okay	why
		hey	more	up	ball	done	duck	two	be	panta	bunny	living	jump
			blanket	right	go	moon	star	daddy	yellow	octopus	lion	room	hide
			bath	car	good	eye	tree	frog	firetruck	tractor	boat	mcdonald	other
				ha	bread	moo	bye	puzzle	green	shiba	poop	old	spider
				yeah	down	mama	all	yuck	nemo	fight	climb	farm	hood
					sun	boo	milk	you	white	it	chocolate	love	ride
					cat	book	truck	ahh	zoo	tricia	bees	sir	fireworks
					baa	hi	back	ow	bird	breakfast	alligator	bite	rainbow
					out	gaga	gone	and	sister	get	animals	stand	see
					apple	cow	dog	are	fall	my	eat	there	goodbye
					for	no	circle	bah	red	sticker	twinkle	walrus	ant
					sheep	black	throw	light	teepee	bridge	please	radar	remember
						horn	neigh	chair	shoe	igloo	deer	airplanes	song
							door	track	yum	peach	volvo	chicken	sky
							brush	i	grape	medicine	mango	worm	check
							phone	will	tickle	police	coffee	cook	found
							abar	hot	house	off	balloon	leaves	me
							flower	sock	walk	head	play	cherry	press
							blue	baby	fox	downstairs	juice	mix	to
							bear	cup	wheel	cry	here	laundry	pick
							button	purple	bicycle	dinosaur	sense	party	scared
							water	mouth	three	garbage	got	zipper	say
							yay	pig	aboard	diamond	let	clam	choo
							mommy	dont	monkey	snow	want	shower	man
								touch	sit	shark	prince	motorbike	dumbo
								booger	bell	airplane	cinderella	motorcycle	drop
								helicopter	horse	starfish	that	at	tweet
								orange	sea	open	tail	fresh	windmill
								mouse	elephant	hungry	cake	ambulance	is
								what	crab	some	this	zoom	who
								one	engine	been	basketball	camel	actual
								heart	butterfly	school	push	beta	with
								diaper	brown	bug	vaseline	hold	wrong
								train	jacket	funny	dirty	place	under
									socks	babyname	box	any	music
									circus	bed	race	thank	brother
									tunnel	change	accident	album	now
									chip	messy	ring	sandals	toothpaste
									pee	gross	bracelet	vegetable	work
									garage	triangle	clock	banana	fix
									towel	big	alexander	dolphin	our
									pink	carpet	put	lay	thirsty
									tea	dish	paste	cloth	take
									bambi	spoon	tooth	round	have
									yummy	fiddle	hard	strawberry	seuss
									peas	sing	store	glasses	ice
										high	toy	bike	drink
										fast	dark	ready	were
										outside	hand	tiger	nope

Figura 4.8 - Schema delle parole apprese dal bambino divise mese per mese. Si parte da parole semplici di frequenze utilizzo e si procede verso parole più complesse. Il picco massimo di apprendimento è tra i 19 e i 22 mesi.

A questo fine sono stati numerosi i meccanismi messi in atto, come ad esempio spegnere il sistema automaticamente nell'ora in cui il bambino andava a dormire, o la possibilità, tramite pannelli installati in ogni stanza, di disattivare la ripresa momentaneamente. Inoltre, appositi pulsanti di sicurezza permettevano, in caso di necessità, di cancellare l'intera ora di ripresa precedente, prima che fosse elaborata e trasmessa al centro di controllo e raccolta.

Il progetto, pur avendo terminato di raccogliere il materiale, è ancora in corso per quanto riguarda l'elaborazione e la correlazione dei dati. Sono già numerosi i risultati ottenuti, che hanno permesso, data l'estensione temporale, di ottenere informazioni mai estratte prima empiricamente, come ad esempio l'esatta sequenza delle parole apprese in ogni mese, i luoghi della casa in cui una parola è stata pronunciata più volte dal bambino, o la relazione tra comportamento dei genitori e le parole apprese [52].

Benché questo progetto non abbia ancora prodotto un vero e proprio modello che spieghi lo sviluppo del linguaggio, ci è stato molto utile come punto di partenza per iniziare la nostra analisi, in particolare per lo studio dello sviluppo del babbling a partire dai primi suoni. Tra il materiale reso disponibile online, infatti, sono presenti dei file audio che contengono una stessa parola pronunciata dal bambino decine di volte in momenti diversi e in mesi diversi, a partire dall'imitazione tramite babbling, fino ad arrivare a pronunciare la parola completa in modo del tutto volontario.

A titolo di esempio riportiamo la trascrizione dello sviluppo progressivo della parola *"water"* che abbiamo ricavato da una registrazione tratta dal materiale fornito online dallo *Speechrome Project*:

- *gaga*
- *gata*
- *wata*
- *wate*
- *water*

Ogni singola associazione consonante-vocale (*"ga", "ta", "wa"*) rappresenta quanto più vicino ad una primitiva linguistica sia possibile estrarre. Il bambino, in presenza di un certo contesto (il bagno o la cucina nel caso dell'acqua) e in risposta alla stessa parola pronunciata da un adulto, prova ad imitare lo stesso suono, inizialmente cercando nel set delle uniche primitive linguistiche che possiede quelle più vicine a quanto ascoltato, in questo caso ripetendo la particella *"ga"*.

Nel corso dei mesi, grazie sia ai movimenti involontari della bocca, che allo sviluppo del tratto vocale e della muscolatura facciale, il bambino impara a riprodurre suoni più complessi, come *"ta"*, o *"wa"*, sostituendoli progressivamente alle particelle iniziali *"ga"*, ottenendo un risultato più somigliante al suono che vuole imitare. Infine, utilizzando i vari meccanismi cognitivi analizzati in precedenza, riesce a riprodurre la parola finale *"water"*.

L'analisi di questo processo, effettuato per diverse parole, ha ispirato e spinto il nostro lavoro di creazione di un modello: a partire da una parola pronunciata da un adulto vogliamo che il sistema tenti di imitarne il suono, sfruttando un dataset di particelle di base (le stesse a disposizione dei bambini) e basandosi su una architettura bioispirata che sfrutti gli stessi meccanismi di apprendimento utilizzati dagli infanti e in particolare il LPM, così da valutarne l'attendibilità.

4.4 Il dataset iniziale di primitive linguistiche

Il punto di partenza per giungere ad un modello valido è dunque quello di ottenere un buon dataset di primitive linguistiche da utilizzare nelle fasi successive dello sviluppo del linguaggio. Questo dataset è presente anche nel sistema cognitivo dei bambini, sotto forma di associazione tra un suono semplice di base e un determinato movimento della bocca e del diaframma [53]. Questa connessione si ottiene nel momento in cui, in seguito ad un suono prodotto involontariamente, il bambino, ascoltandosi, comprende il nesso causa-effetto che ha provocato quel risultato.
Ognuna di queste associazioni costituisce una primitiva linguistica e, una volta registrata, può essere usata volontariamente dal bambino per riprodurre dei suoni (o delle parole) che ascolta da un adulto, o per comunicare un bisogno.

Per capire come generare il dataset iniziale abbiamo analizzato approfonditamente le diverse tipologie di suoni prodotti dai bambini in mesi differenti del loro sviluppo, partendo da numerosi video e registrazioni di babbling. Ognuno di questi video riportava l'età dei bambini, cosicché fosse semplice identificare il preciso stadio in cui si trovavano.

Riportiamo in seguito le trascrizioni estratte dai video, suddividendole in cinque stadi differenti [47]:

Stadio 1: **Cooing (1-4 mesi)**
eg: ooooooo, aaaaaaah

Stadio 2: **Consonant-Vowel (CV) or Vowel-Consonant (VC) sound combinations (4-6 mesi)**
eg: maaaa, uuuum, baaaa

Stadio 3: **Reduplicated Babbling (6-10 mesi)**
eg: babababa, gagagaga, dadadada

Stadio 4: **Nonreduplicated Babbling (6-10 mesi)**
eg: bama, gagamee

Stadio 5: **Quasiwords (10-12 mesi)**
eg: watee

Il primo stadio è composto da suoni quasi totalmente involontari, generati da riflessi che portano a comprimere il diaframma e sono difficili da trascrivere ed utilizzare: rappresentano dunque un livello di linguaggio inadeguato ai nostri scopi perché riguardano un livello troppo basso di cognizione.

Il secondo e terzo livello invece sono le fasi più interessanti per l'estrazione di un dataset di suoni elementari, perché sono prodotti da meccanismi che prevedono, oltre alla compressione del diaframma, il movimento della bocca e della lingua in diverse posizioni diverse. Questi suoni, adeguatamente trascritti e segmentati, costituiranno le nostre primitive linguistiche e saranno utilizzati in tutte le elaborazioni successive.

Nel modello che abbiamo sviluppato non sarà trattata la parte relativa allo sviluppo biologico del tratto vocale del bambino, che inseriremo tra gli sviluppi futuri, e la parte relativa alla nascita delle primitive che compongono il dataset perché sono considerate già presenti nel bambino. Questo tuttavia non rappresenta un grosso limite al modello perché le primitive linguistiche base sono a tutti gli effetti innate, dato che si sviluppano per motivi fisiologici (come i riflessi involontari) e sono uguali in tutti i bambini indipendentemente dalla lingua o dal contesto.

Nello stadio quattro e cinque si assiste infine all'utilizzo delle primitive linguistiche iniziali per comporre suoni più complessi ed imitare alcune parole. Essendo il dataset disponibile molto limitato, non si riescono a comporre ancora parole complete, se non termini molto semplici (come "mama" o "dad"), composti dall'associazione di due o più primitive linguistiche. Soltanto nell'ultima fase, grazie all'arricchimento del dataset in seguito allo sviluppo dei muscoli facciali e dei movimenti volontari, i bambini riescono a pronunciare parole con suoni più complessi, ancora non perfettamente formate ma abbastanza chiare da essere comprese.

Capitolo 5

Analisi ed elaborazione dei segnali audio in Praat e Matlab

"Tutto ciò che vediamo o a cui rassomigliamo è soltanto un sogno dentro un sogno?"

Edgar Allan Poe da "A dream within a dream"

Per ottenere la prima versione del dataset di primitive linguistiche, in attesa che sia reso disponibile l'enorme database ricavato dallo *Human Speechrome Project*, ed essendo molto difficile ottenere delle registrazioni dal vivo, abbiamo scelto di creare un sistema in grado di estrarre i suoni elementari da video e file audio. Parte del materiale audio utilizzabile è rappresentato da alcuni esempi estratti dall'HSP, mentre un'altra parte può essere estratta da video presenti sul web, che trattano specificamente le varie fasi dello sviluppo linguistico e mostrano diversi bambini dai 6 ai 12 mesi nei numerosi stadi del babbling. Per estrarre le primitive abbiamo utilizzato unicamente video in cui fosse presente l'indicazione esatta relativa all'età del bambino, così da selezionare soltanto le registrazioni degli stadi corretti.

Per perseguire questa scelta abbiamo dovuto approfondire diversi aspetti legati all'elaborazione dei segnali audio, processi necessari prima di tutto ad ottenere un dataset di primitive linguistiche adeguato e in seguito a costruire un modello in grado di utilizzarle nell'apprendimento linguistico. L'ambiente di sviluppo scelto è Matlab (Matrix Laboratory), che uno dei software più utilizzati nell'ambito della costruzione di modelli perché permette di concentrarsi unicamente sui dettagli matematici e logici del sistema, astraendo dai dettagli implementativi che riguardano sistemi specifici. Inoltre molti metodi matematici che abbiamo utilizzato per elaborare i segnali sono già compresi nelle librerie software di Matlab e piuttosto efficienti in termini di velocità e utilizzo memoria.

L'unico caso in cui abbiamo utilizzato un software differente (Praat) è la segmentazione di file audio poiché abbiamo scelto di seguire un approccio semi-manuale dato che gli algoritmi automatici esistenti, che fanno uso di classificatori, risultano inefficienti per una tipologia audio così specifica come il babbling [54].

5.1 Segmentazione dei file audio con Praat

Il primo problema che il modello deve considerare è quello relativo al rumore e alla scarsa qualità dei file audio che, se non appositamente elaborati, sono inutilizzabili ai fini delle analisi necessarie ai vari processi. Bisogna tener conto di questi fattori anche perché, pur avendo in futuro la possibilità di registrare i suoni dal vivo e di integrare tutto il sistema in un robot umanoide, la qualità dei suoni sarà sempre soggetta a rumore legato al contesto o a scarsa qualità degli apparati di registrazione.

Il rumore, unito alla fisiologica ambiguità dei suoni del babbling che non sono parole complete ma sequenze di consonante-vocale, rende la fase di segmentazione piuttosto complicata, anche perché i vari metodi sviluppati per il riconoscimento del parlato lavorano con parole complete, ma non sono in grado di isolare e trascrivere correttamente il babbling. La segmentazione del suono rappresenta dunque la prima grande sfida da affrontare se si vuole sviluppare un modello valido, anche perché un dataset poco valido renderebbe inefficaci tutte le elaborazioni successive.

Data l'ambiguità dei suoni e l'assenza di sistemi esistenti validi che siano in grado di segmentare adeguatamente la particolare tipologia di file audio che utilizziamo, abbiamo scelto di estrarre i suoni in modo semi-manuale utilizzando Praat, un software di elaborazione dei suoni creato da due ricercatori dell'Università di Amsterdam e molto utilizzato nell'ambito della ricerca.

In questo caso, benché più lenta di una completamente automatica, permette di selezionare i suoni di base in modo più accurato e finalizzato a nostri scopi.

Per raggiungere una precisione maggiore abbiamo deciso di utilizzare anche le rappresentazioni visive dell'onda sonora (in diverse forme) e non basarci unicamente sull'ascolto delle tracce audio, perché in questo modo risulta molto più semplice identificare i punti di transizione tra un suono e l'altro. Così è possibile, inoltre, effettuare delle query di ricerca per ottenere statistiche e per evidenziare i punti notevoli nel grafico dell'onda (es: picchi, zero-crossing, punti di minimo). Infine, una volta elaborato l'algoritmo, è possibile scrivere degli script per automatizzare i processi e i passi che si vogliono seguire [54].

Un'onda audio si può rappresentare in diversi modi: il più comune è quello di utilizzare un grafico cartesiano con il tempo sull'asse delle ascisse e lo spostamento delle particelle nello spazio su quello delle ordinate [55]. In questo modo appare subito chiaro il significato fisico del suono: un'oscillazione nello spazio (sempre in un mezzo) compiuta da atomi e molecole in seguito ad una variazione di pressione.

Tuttavia, data la natura delle onde sonore soprattutto in caso di forte rumore, è molto complicato, se non impossibile, segmentare l'audio manualmente basandosi unicamente sulla loro rappresentazione nel tempo. In particolare è impossibile conoscere con precisione, unicamente da un analisi visiva dell'onda, il punto dove termina il suono prodotto dal bambino ed inizia il rumore di fondo.

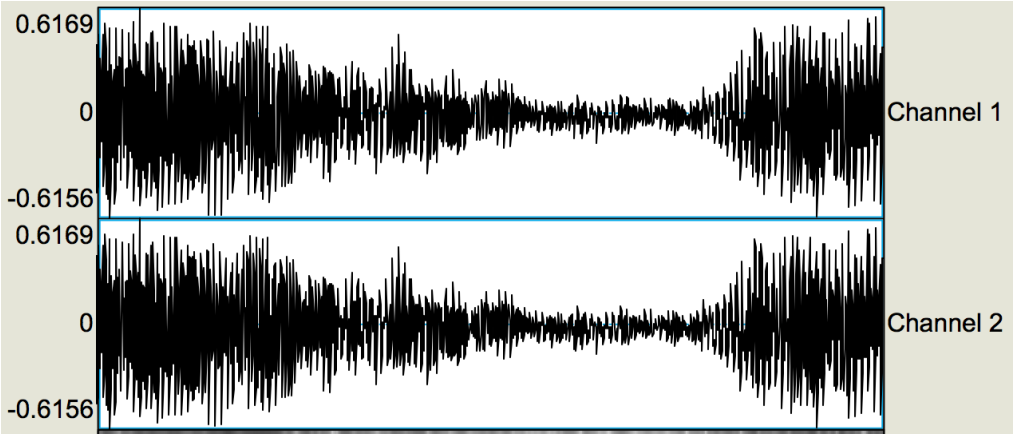


Figura 5.1 - Il grafico l'onda sonora di un frammento di babbling nel tempo. E' difficile discriminare con esattezza il punto in cui finisce un suono ed inizia il successivo a causa del rumore di fondo.

Un'altra rappresentazione del suono che utilizzeremo nel nostro modello è lo spettrogramma: esso rappresenta graficamente l'intensità sotto forma di frequenze (sulle ascisse) in funzione del tempo (sulle ordinate) [54]. L'orecchio umano riesce a percepire le frequenze dai 20Hz ai 20KHz, e spesso questo metodo di visualizzazione viene utilizzato nei sistemi di riconoscimento vocale, dato che ogni parola possiede uno spettro caratteristico.

Tratteremo in seguito nel dettaglio il metodo con il quale si ottiene lo spettrogramma a partire da un suono tramite la Trasformata di Fourier (in particolare useremo la Trasformata Veloce di Fourier).

In questa fase non sarà tuttavia possibile utilizzare lo spettrogramma del segnale dato che anche in questo caso, sempre a causa del rumore e per le caratteristiche intrinseche a questo tipo di rappresentazione, è difficile sfruttarla per la segmentazione semi-manuale.

L'ultima tipologia di rappresentazione riguarda invece l'intensità, che in acustica definisce il volume del suono, distinguendo suoni deboli dai forti [55]. L'andamento dell'intensità viene mostrato in evidenza nella Figura 5.2 ed è espresso in Decibel (dB), dunque in forma logaritmica.

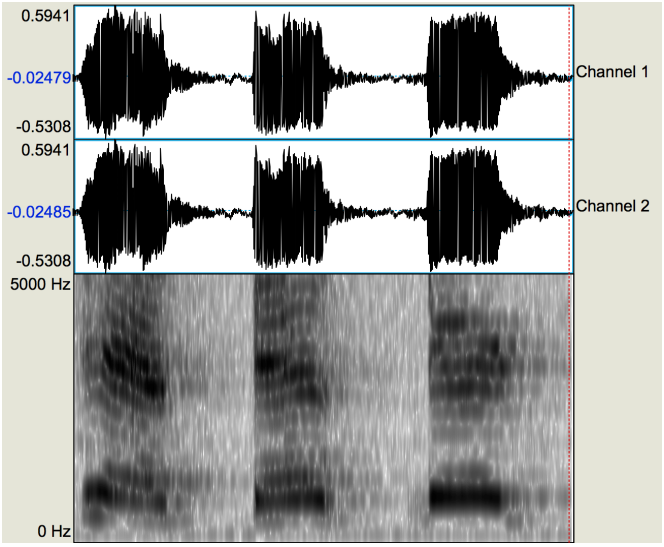


Figura 5.2 - Il grafico superiore mostra l'onda sonora di un frammento di babbling nel tempo, mentre quello inferiore mostra lo spettrogramma con le frequenze in base al tempo.

Come si può vedere nell'immagine, questa visualizzazione grafica permette di individuare gli istanti temporali in cui cessa completamente la voce del bambino e resta unicamente il rumore di fondo evidenziando, tramite query, i punti di minimo relativo. Utilizzando l'intensità, ed automatizzando il processo di query tramite script Praat, è possibile ottenere un sistema semi-automatico di segmentazione.

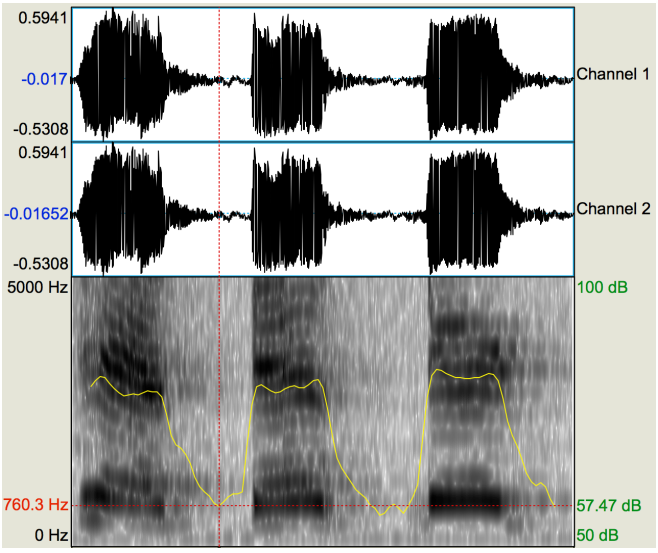


Figura 5.3 - Il grafico superiore mostra l'onda sonora di un frammento di babbling nel tempo, mentre quello inferiore mostra lo spettrogramma con le frequenze in base al tempo a cui è stato sovrapposto l'andamento dell'intensità sonora (in giallo). Le linee tratteggiate rosse mostrano un punto di minimo relativo, che sarà poi utilizzato per effettuare la segmentazione.

Nonostante la possibilità di realizzare uno script, il sistema viene definito semi-automatico perché Praat non permette la stessa interoperabilità con le altre fasi del modello che permetterebbe un ambiente di sviluppo come Matlab; inoltre è necessario monitorare manualmente la composizione e la qualità dei suoni processati, dato che anche utilizzando l'andamento dell'intensità, in caso di contesti molto rumorosi, è difficile ricavare una threshold valida per separare il suono dal silenzio.

5.2 Normalizzazione dei segnali audio

Dopo aver effettuato la segmentazione con uno dei metodi descritti in precedenza, si avrà a disposizione un dataset più o meno ampio in base alle esigenze, che contiene numerose occorrenze dei suoni base del babbling, ossia delle primitive linguistiche. Queste rappresentano i suoni elementari (non ulteriormente scomponibili) che il bambino è in grado di produrre in modo volontario per cercare di imitare le parole degli adulti o i suoni che ascolta nell'ambiente circostante.

Ciascuna primitiva linguistica è salvata in un file a parte, ognuno dei quali contiene un'associazione consonante-vocale o una *quasivowel*. Nel sistema vengono utilizzati unicamente i file nel formato *.wav*, (WAVE), una codifica digitale del suono molto diffusa che permette di immagazzinare i dati senza alcun tipo di compressione, semplicemente digitalizzando la forma dell'onda. I vantaggi di questa codifica sono l'alta qualità del segnale e la semplicità di riproduzione ed elaborazione, caratteristiche assolutamente necessarie per i nostri scopi. Il fatto che il file prodotto sia di elevate dimensioni (non essendo compresso) non porta grossi svantaggi nel nostro sistema, dato che, operando con primitive linguistiche, i file prodotti hanno comunque una durata molto breve e conseguentemente una dimensione piuttosto ridotta.

Una volta ottenuto il dataset iniziale il sistema, nel modulo dedicato alla lettura dei file, carica in memoria ciascun file salvando, in una struttura apposita, il segnale e la frequenza del segnale.

5.2.1 Normalizzazione dei canali

In Matlab ogni segnale audio è rappresentato o da un vettore colonna o da una matrice, a seconda che sia monofonico o stereofonico. In quest'ultimo caso il segnale è rappresentato da una matrice di due colonne, ciascuna delle quali si riferisce ad un diverso canale (destro o sinistro), che possono avere anche una frequenza diversa.

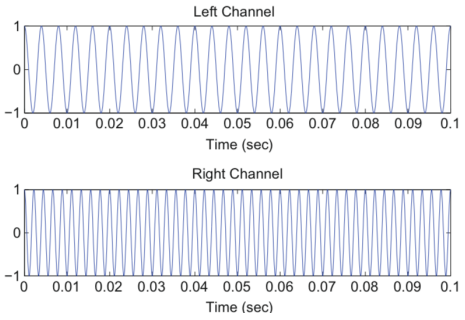


Figura 5.4 - Le due forme d'onda che compongono un segnale stereofonico. Il canale sinistro e quello destro hanno una frequenza differente.

Il primo passo compiuto dal sistema è assicurarsi che ogni file caricato abbia un solo canale (sia dunque monofonico), dato che analizzare ed elaborare segnali polifonici risulta molto più complesso. Nel caso in cui il segnale non fosse monofonico si esegue la conversione del segnale unendo i due canali [55].

```
x = sum(x,2); %where x = dataSet(ii).sig
```
(5.1)

5.2.2 Normalizzazione delle frequenze

Estratti i segnali e le frequenze dei file, convertiti e salvati in una struttura apposita, si passa nel modulo di normalizzazione dove i diversi segnali vengono resi omogenei: infatti, per procedere nelle analisi approfondite dei suoni e in particolare nell'estrazione delle feature e nel processo di concatenamento, è necessario che i suoni abbiano caratteristiche simili per quanto riguarda frequenza, ampiezza, e pitch, così da poter essere facilmente confrontati.

Questo passo è necessario e di enorme importanza sia nel caso in cui il dataset venga estratto da registrazioni audio di tipi e qualità differenti, sia nel caso in cui tutte le registrazioni vengano fatte con lo stesso strumento e dal vivo: anche in questa situazione infatti ci possono essere differenze di ampiezza, pitch e rumore, dovuti a disturbi o a modifiche del contesto di acquisizione nel tempo.

Nel processo di registrazione di un segnale audio la frequenza di campionamento (*sampling*) è uno dei parametri più importanti, perché determina il numero di campionamenti che vengono effettuati, dal meccanismo di registrazione, sul segnale reale (che è continuo). Questi, riprodotti nel tempo, daranno vita al segnale digitale, che invece è discreto.

Se la frequenza di sampling è troppo bassa il segnale digitale risultante sarà di scarsa qualità e sarà soggetto al fenomeno dell'*aliasing*, (o distorsione da

campionamento lento) che porta due segnali diversi nel segnale analogico, a diventare indistinguibili nel segnale digitale: si avrà una distorsione del segnale generale e, in questo caso, della registrazione audio [55].

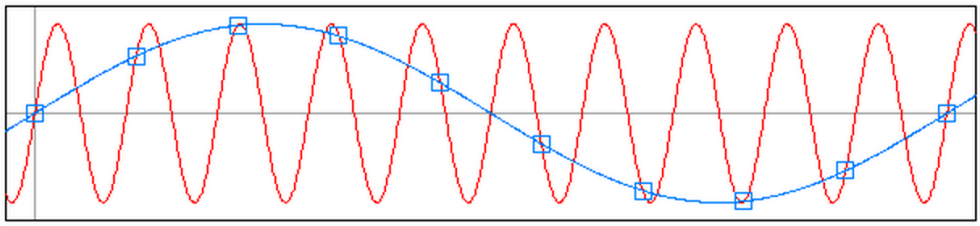


Figura 5.5 - Il segnale rosso (analogico) viene campionato (ogni quadrato è un campione del segnale) per essere trasformato in digitale, ma essendo la frequenza troppo bassa il segnale digitale avrà una forma differente e risulterà dunque distorto.

Per evitare questo fenomeno e ridurre al minimo la distorsione del segnale, la frequenza di sampling deve essere maggiore o uguale ad almeno due volte la frequenza massima del segnale originale iniziale, un *lower-bound* della frequenza di sampling che è conosciuto come frequenza di Nyquist.

I concetti esposti sopra vanno tenuti presenti sia durante la fase di registrazione del segnale (che nel caso del nostro progetto è assunta già eseguita), che nel caso di un *re-sampling*, ossia di un nuovo campionamento effettuato sul segnale digitale, che questa volta ha l'obiettivo specifico di modificare la frequenza. Essendoci la possibilità che i segnali utilizzati per il dataset abbiano una frequenza differente (dovuta a diversi metodi di registrazione), nel primo step, all'interno del modulo di normalizzazione, viene eseguito un *re-sampling* per portare tutti i segnali alla stessa frequenza, in particolare a 44100 Hz. Questa frequenza è ben più alta di quella udibile dall'orecchio umano (che va dai 20Hz ai 20kHz), ma coincide con lo standard utilizzato nelle registrazioni di cd musicali, perché permette di evitare distorsioni e di ottenere una migliore qualità del suono.

Per ottenere questa conversione si utilizza la funzione *resample(x, P, Q)*, che effettua il *re-sampling* del segnale x ad una frequenza P/Q volte l'originale, utilizzando un'implementazione polifase, dove P/Q rappresenta il rapporto razionalizzato tra la nuova frequenza desiderata e quella iniziale del file originale:

```
[P,Q] = rat(newFs/dataSet(ii).freq);                    (5.2)
dataSet(ii).sig = resample(dataSet(ii).sig ,P,Q);
```

La funzione *resample(x, P, Q)* applica inoltre un filtro passa-basso che funge da *anti-aliasing*, così da ridurre al minimo la distorsione.

5.2.3 Normalizzazione dell'ampiezza

Le registrazioni audio dalle quali abbiamo estratto le primitive linguistiche potrebbero essere state effettuate utilizzando diversi strumenti di registrazione, con caratteristiche e parametri differenti che portano a tonalità ed ampiezze non omogenee. Inoltre, anche se i suoni fossero registrati direttamente tramite un automa, ci potrebbero essere variazioni e distorsioni dovute a cambiamenti di posizione dell'apparato audio rispetto alla fonte del suono: tali modifiche agiscono soprattutto sull'ampiezza del segnale audio che dipende quindi fortemente dalla distanza. Prima di operare sulla tonalità del suono che è piuttosto complessa da modificare e per far sì che le elaborazioni successive siano più efficienti, è bene normalizzare l'ampiezza rendendola omogenea tra un suono e l'altro.

Il primo passo da compiere è quello di normalizzare i picchi di ampiezza facendo in modo che siano compresi nel range tra -1 ed 1. Se così non fosse si rischierebbe di ottenere suoni particolarmente striduli e dal volume elevato, in particolare durante le elaborazioni e le interpolazioni delle fasi successive. Per prima cosa, nel modulo di normalizzazione dell'ampiezza, viene sottratta al segnale la media dell'ampiezza, così che la media totale sia zero. Questo procedimento serve a traslare il segnale e a centrarlo.

```
dataSet(ii).sig = (dataSet(ii).sig -                    (5.3)
mean(dataSet(ii).sig));
```

A questo punto si deve normalizzare l'errore quadratico medio di tutti i segnali audio. Ciò è reso necessario dal fatto che segnali con diverso errore quadratico medio suoneranno in modo differente, anche se uguali. Inoltre, senza compiere questa operazione, si rischierebbe di comprimere l'andamento delle onde sonore, in particolare quelle che hanno picchi di ampiezza molto elevati.

Il modulo ricava l'errore quadratico medio da ogni segnale, calcola l'errore massimo e rapporta tutti gli altri a questo. Infine tutti i segnali audio vengono scalati del minimo errore quadratico medio risultante, così da non distorcere il suono ed eguagliando tutti i valori [55].

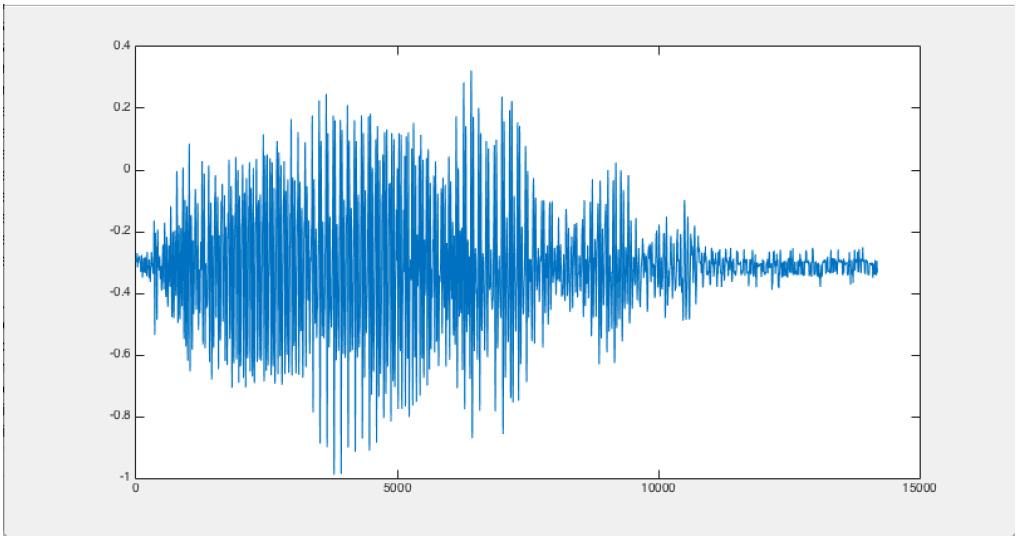


Figura 5.6 - Il grafico mostra un segnale audio preso dal dataset contenente le primitive linguistiche iniziali non normalizzate.

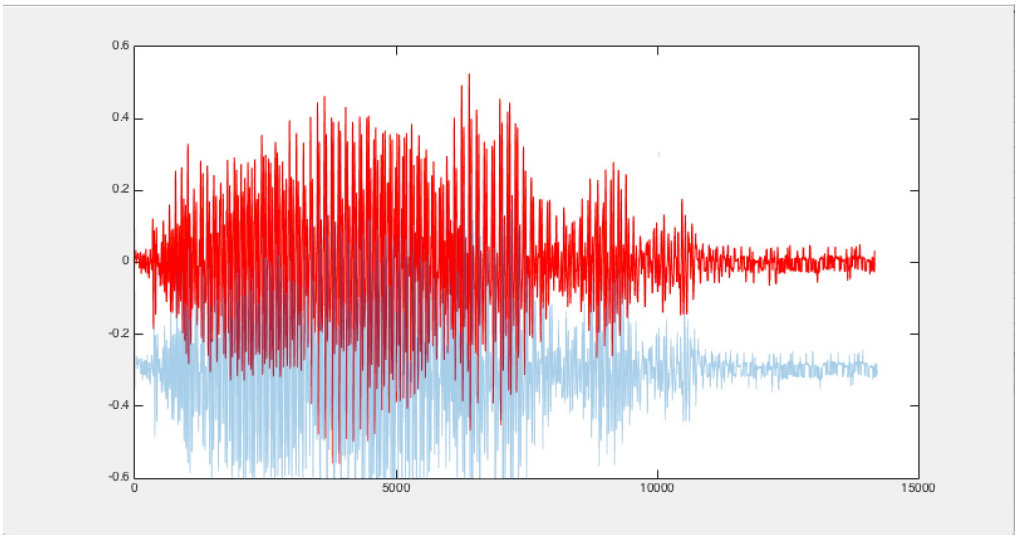


Figura 5.7 - Il grafico mostra lo stesso segnale audio della figura 5.6 dopo la fase di normalizzazione dell'ampiezza (in rosso), sovrapposto al grafico dell'onda non normalizzata (in blu).

5.2.4 Normalizzazione del pitch

Oltre all'ampiezza dell'onda che determina l'intensità di un suono, un concetto molto importante, basilare in ambito musicale, è il pitch, ossia l'altezza [56]. Il pitch corrisponde al set di frequenze da cui è composto il suono e da questo parametro dipende, ad esempio, la differenza di tono che ha una donna rispetto ad un uomo, pur cantando la stessa melodia. Se l'uomo cercherà di aumentare la sua altezza vocale, produrrà un suono simile a quello della donna, seppur con un timbro differente.

Il range musicale è diviso in diverse ottave, ciascuna delle quali è composta da 12 semitoni, definiti come metà step. Ogni semitono corrisponde ad una specifica nota musicale, riproducibile non solo con la voce ma con qualsiasi strumento musicale, che vibri a quella precisa frequenza. La differenza che c'è nella voce di persone diverse o tra strumenti differenti che suonano la stessa nota, è invece definita dal timbro, ossia dalle armoniche che compongono l'onda [56].

Il timbro ci permette di distinguere strumenti diversi che suonano la stessa nota perché ogni nota, oltre ad essere formata dal set di frequenze fondamentali (pitch), è composta da un set di armoniche (timbro) che hanno una frequenza che è un numero intero di volte la frequenza fondamentale (ad esempio 2x440Hz = 880Hz, 3x440Hz=1320Hz,…, corrispondono alle frequenze delle armoniche di una stessa nota suonata da strumenti differenti).

Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz
C1	32.7	C2	65.4	C3	130.8	C4	261.6	C5	523.3	C6	1046.5	C7	2093.0
C#1	34.6	C#2	69.3	C#3	138.6	C#4	277.2	C#5	554.4	C#6	1108.7	C#7	2217.5
D1	36.7	D2	73.4	D3	146.8	D4	293.7	D5	587.3	D6	1174.7	D7	2349.3
D#1	38.9	D#2	77.8	D#3	155.6	D#4	311.1	D#5	622.3	D#6	1244.5	D#7	2489.0
E1	41.2	E2	82.4	E3	164.8	E4	329.6	E5	659.3	E6	1318.5	E7	2637.0
F1	43.7	F2	87.3	F3	174.6	F4	349.2	F5	698.5	F6	1396.9	F7	2793.8
F#1	46.2	F#2	92.5	F#3	185.0	F#4	370.0	F#5	740.0	F#6	1480.0	F#7	2960.0
G1	49.0	G2	98.0	G3	196.0	G4	392.0	G5	784.0	G6	1568.0	G7	3136.0
G#1	51.9	G#2	103.8	G#3	207.7	G#4	415.3	G#5	830.6	G#6	1661.2	G#7	3322.4
A1	55.0	A2	110.0	A3	220.0	A4	440.0	A5	880.0	A6	1760.0	A7	3520.0
A#1	58.3	A#2	116.5	A#3	233.1	A#4	466.2	A#5	932.3	A#6	1864.7	A#7	3729.3
B1	61.7	B2	123.5	B3	246.9	B4	493.9	B5	987.8	B6	1975.5	B7	3951.1

Tabella 5.1 - Ogni colonna rappresenta un'ottava, divisa a sua volta in 12 semitoni, ciascuno dei quali è una nota differente.

L'ultimo step del modulo di normalizzazione prevede dunque, per rendere omogenee le primitive linguistiche che compongono il dataset del sistema, di uniformare i pitch dei vari file, mentre il timbro non è preso in considerazione dato che i bambini, avendo un tratto vocale non del tutto sviluppato, hanno un

timbro vocale piuttosto standard. La normalizzazione del pitch è invece necessaria perché si parte dal presupposto che le registrazioni audio, se pur dello stesso soggetto, non siano effettuate tutte allo stessa altezza, dato che un bambino non è in grado di controllare l'altezza della sua voce come potrebbe farlo un adulto dopo numerosi esercizi di canto.

La tecnica che permette di modificare il pitch di una registrazione audio è chiamata *pitch shifting*, ed è comunemente utilizzata in ambito musicale per modificare il suono di strumenti e voci dei cantanti, solitamente con meccanismi hardware. Nel nostro modello abbiamo scelto di portare tutti i suoni del dataset ad una frequenza di pitch di 440Hz (A4), che è definita dalla letteratura musicale come la frequenza fondamentale [56].

Per fare questo bisogna considerare la relazione che vige tra la frequenza fondamentale e i semitoni, espressa dalla formula

$$p = 69 + 12 \times \log\left(\frac{f}{440}\right) \tag{5.4}$$

dove p è il numero di semitono e f la sua frequenza in Herz.

L'algoritmo che viene utilizzato nel modulo di normalizzazione, per effettuare il pitch shift, prevede tre fasi:

1. si determina il pitch della primitiva linguistica
2. si calcola il numero di step (semitoni) necessari nel processo di shifting per ottenere il pitch voluto (440Hz)
3. si effettua lo shifting

Determinazione del pitch

Determinare il pitch di un suono è un'operazione molto importante non solo in ambito musicale, ma anche in tutti i processi implicati nella sintesi di parole e linguaggio, e nei sistemi di conversione *text-to-speech*, per ottenere una lingua parlata più realistica e comprensibile. Rilevare e modificare il pitch permette infatti di gestire correttamente la pronuncia delle parole da parte dei sistemi automatici e il tono generale da dare ad una parola in una determinata frase in base al contesto.

Questa ottimizzazione dei toni si rivela di centrale importanza soprattutto nei sistemi di ultima generazione che tendono ad essere estremamente realistici, come ad esempio nel sistema di sintesi vocale Vocaloid, un software inizialmente sviluppato dalla Yamaha Corporation, tutt'ora molto diffuso in Giappone, che permette di far cantare un brano ad una voce sintetizzata, semplicemente immettendo il testo e la melodia [57].

La voce virtuale viene poi adattata variando il pitch e fondendo le sillabe tramite cross-fading, una tecnica utilizzata anche nel nostro sistema nel modulo di concatenamento, di cui parleremo nel dettaglio in seguito.

Tra i vari metodi che permettono di stimare il pitch di un suono abbiamo scelto di utilizzare un algoritmo che fa uso del *Cepstrum*, molto utilizzato nella teoria dei segnali per eseguire vari compiti [56]. Questo metodo esegue la trasformata di Fourier applicata allo spettro in decibel di un segnale, in pratica esegue una doppia trasformata di Fourier, lo spettro dello spettro, da cui deriva il nome "Cepstrum" che è la parola "Spettro" con le lettere scambiate di posto.

L'algoritmo prevede le seguenti fasi:
segnale audio → trasformata di Fourier (FT) → logaritmo → fase istantanea → trasformata di Fourier (FT) → Cepstrum

Nel nostro caso l'algoritmo funziona perché se il logaritmo dello spettro del segnale audio contiene numerose armoniche regolarmente distanziate l'una dall'altra, la trasformata di Fourier mostrerà un picco in corrispondenza di questa distanza, che è esattamente la frequenza fondamentale del segnale, ossia il suo pitch.

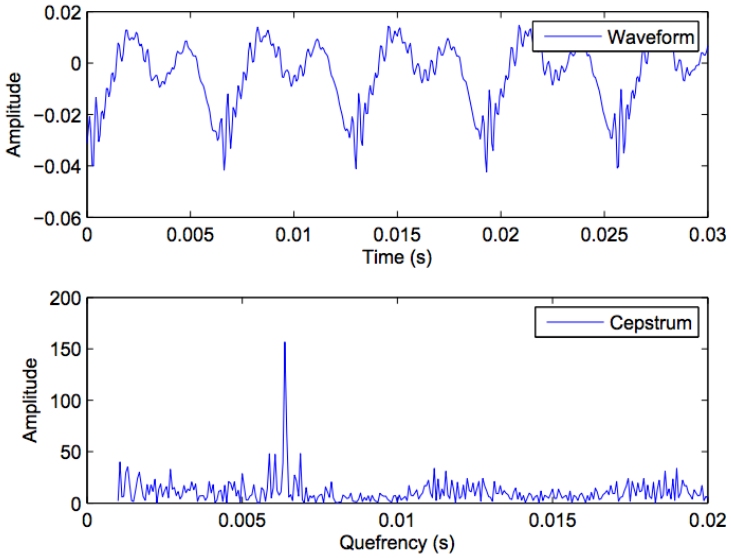


Figura 5.8 - L'asse delle ordinate del cepstrum ha come unità di misura la quefrency (che è la parola "frequency" con le lettere invertite di posto), e il picco nel cepstrum corrisponde alla periodicità nello spettro che indica esattamente la frequenza fondamentale (il pitch) che stiamo cercando, in questo caso intorno ai 156 Hz.

Calcolo degli step

Una volta identificato il pitch del file iniziale è necessario effettuare l'operazione di *pitch shifting* per assegnargli il valore desiderato, che nel nostro caso è esattamente la frequenza fondamentale 440Hz. Sono numerosi gli algoritmi che permettono di fare lo shifting, ma solitamente operano la modifica una volta specificato il numero di step (ossia di semitoni) di cui ci si vuole traslare, indipendentemente dal pitch iniziale di partenza: il pitch finale è determinato dal pitch iniziale variato (aumentato o diminuito) di s semitoni.

$$p_{final} = p_{initial} + s \tag{5.5}$$

Nel nostro caso invece si vuole effettuare lo shift di un numero di semitoni dipendente dal pitch iniziale, dunque bisogna trovare l'esatto numero di step (non necessariamente intero), che permettono di raggiungere un pitch di 440Hz.

Per farlo abbiamo sfruttato la seguente equazione che mette in relazione le frequenze di pitch con il numero di step *s*, dipendenza non lineare bensì esponenziale, che ci permette di calcolare lo step necessario specifico per ogni file all'interno del dataset.

$$f_{final} = 2^{(\frac{s}{12})} \times f_{initial} \tag{5.6}$$

Shifting

Il metodo più intuitivo per effettuare uno shift del pitch è variare la frequenza del segmento audio: ad esempio se provassimo a riprodurre un file al doppio della velocità, noteremmo un pitch molto elevato e una deformazione della voce, dato che anche la frequenza, insieme alla velocità, raddoppia. La difficoltà di effettuare lo shifting nasce proprio da questo fatto: si vuole raddoppiare la frequenza di riproduzione, ma lasciare in alterata la lunghezza del file audio, che corrisponde a variare il tono della voce senza velocizzarne e dunque deformarne la riproduzione.

Per raggiungere questo obiettivo si comprimerà o espanderà il segnale di un fattore di scala tale da modificare il pitch nel modo voluto (rappresentato dallo step calcolato prima); una volta fatto, si procederà ad effettuare un *re-sampling* per ripristinare la durata iniziale del file, mantenendo la variazione nel pitch.

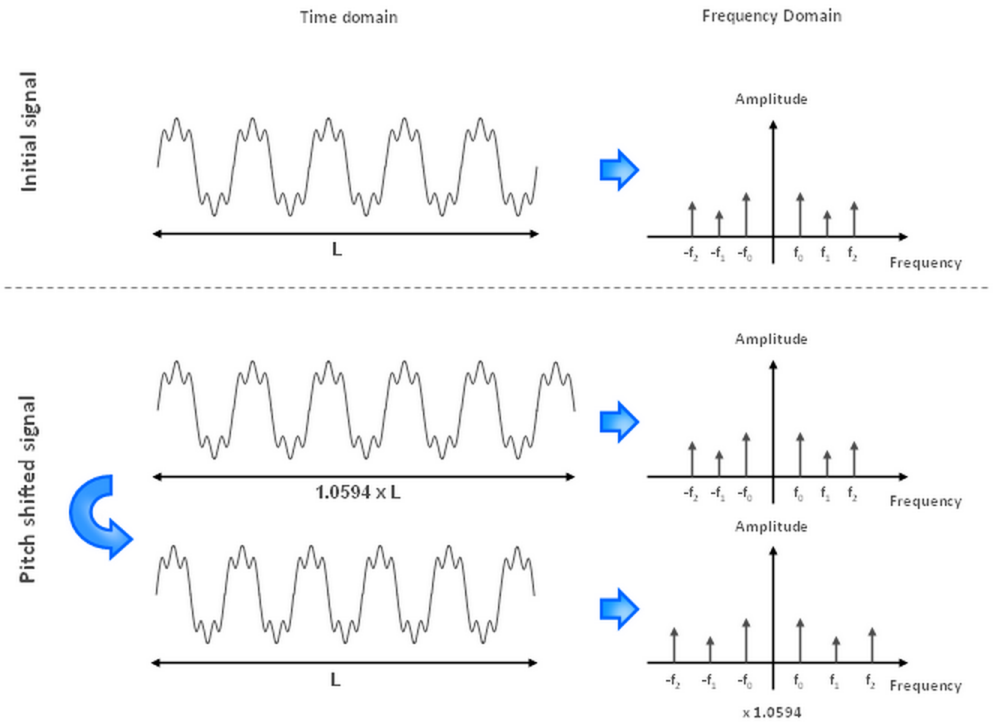


Figura 5.9 - Inizialmente il segnale viene allungato esattamente del fattore di scala (in questo caso 1.0594), lasciando le frequenza inalterate. Poi sarà effettuato un re-sampling per tornare alla lunghezza originale del file e aggiustare le frequenze.

Le fasi necessarie al processo di *shifting* sono tre:

1. *Creazione dei frame e sovrapposizione*
2. *Phase Vocoder*
3. *Resampling*

Per compiere il processo di shifting sarà eseguita, prima di tutto, un'operazione di framing (o windowing) sul segnale audio, che corrisponde ad una segmentazione del segnale originale in più parti di uguale durata. Questa tecnica, che per lo shifting del pitch è basilare, è impiegata anche in moltissime altre elaborazioni audio, perché permette di effettuare uno *short term audio processing*, ossia un procedimento che prevede di concentrarsi su un frammento dell'audio originale alla volta, riducendo l'allocazione della memoria da parte dell'algoritmo e ottimizzando i risultati.

Il segnale audio infatti non è stazionario per sua stessa natura e varia nel tempo molto rapidamente, come ad esempio nel caso in cui, durante la registrazione di una conversazione si avvertisse uno sparo. Se le analisi fossero eseguite sul frammento totale, le statistiche ricavate non sarebbero molto significative perché dominate dai campionamenti effettuati nell'istante dello sparo.

Adottando invece la tecnica di framing si procederebbe a calcolare statistiche separate per ogni frammento dell'audio originale, per poi unire tutto insieme una volta terminato il processo (ad esempio vedremo un'applicazione di questa tecnica nel paragrafo che tratterà della *features extraction*).

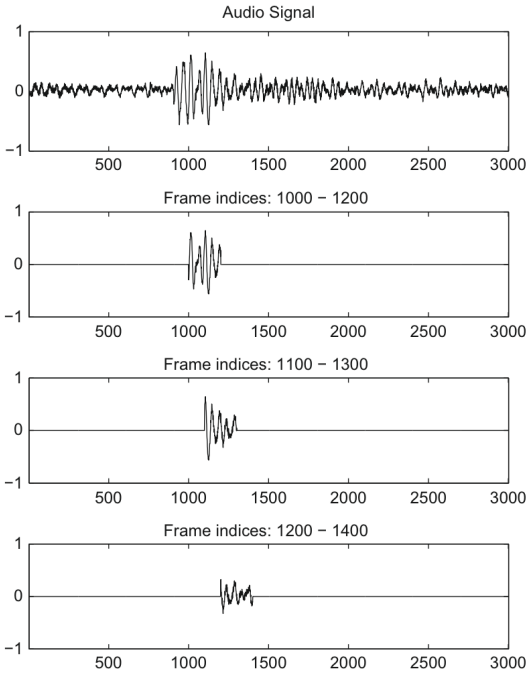


Figura 5.10 - Il segnale audio originale (nel primo rettangolo) viene suddiviso in diversi frame di uguale durata parzialmente sovrapposti l'uno all'altro.

I parametri da considerare quando si effettua un'operazione di framing su un segnale lungo *N* sample sono: la lunghezza della finestra di volta in volta valutata *Wl*, l'*hop-size Ws*, che determina il grado di sovrapposizione (*overlapping*) parziale tra una finestra e l'altra e la funzione utilizzata per ottenere le finestre *w(n)*.

Il numero totale di frame *K* sarà determinato dalla seguente formula:

$$K = \left[\frac{N - W_L}{W_S} \right] + 1 \tag{5.7}$$

mentre una possibile funzione per ottenere le finestre (in questo caso finestre rettangolari) potrebbe essere:

$$w(n) = \begin{cases} 1, & 0 \le n \le W_L - 1, \\ 0, & elsewhere. \end{cases} \tag{5.8}$$

Nel caso del pitch shifting, il processo di framing viene utilizzato anche per ottenere l'allungamento o la compressione del file audio (in base alle esigenze), semplicemente variando la distanza tra un frame o l'altro, ossia aumentando o diminuendo la regione di overlapping (tramite un hop-size specifico).

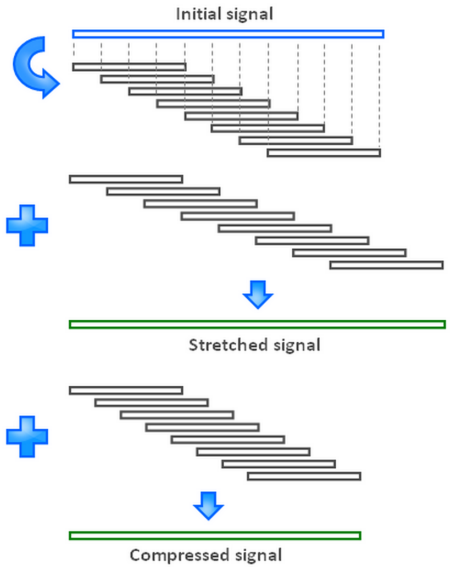


Figura 5.11 - Il segnale iniziale viene suddiviso in numerosi frame parzialmente sovrapposti, come spiegato in precedenza. In seguito si aumenta o diminuisce la porzione di sovrapposizione dei frame, così da estendere o comprimere l'intero segnale.

Questo procedimento, seppur molto efficace, potrebbe tuttavia creare discontinuità nel nuovo segnale audio generato, tali da essere avvertite dall'orecchio umano; è per questo che nel passo successivo il modello prevede l'utilizzo di un *phase vocoder* per ri-sintetizzare il segnale eliminando tali discontinuità.

Un phase vocoder è uno specifico tipo di *vocoder* (ossia di *voice encoder*, codificatore della voce umana) che permette di modificare un suono digitale trasportandolo nel dominio delle frequenze, analizzandone il segnale, elaborandolo ed infine sintetizzandolo nuovamente, riportandolo nel dominio del tempo [58]. Questo sistema fa uso della Trasformata di Fourier (in particolare della Trasformata Veloce di Fourier) per convertire il suono dal

dominio del tempo a quello delle frequenze, permettendo così di modificare l'ampiezza o la fase di frequenze specifiche (ripristinando, nel caso dello *shifting*, la coerenza delle fasi nei vari segmenti), prima di ri-sintetizzarlo nuovamente riportandolo nel dominio del tempo.

Infine, nell'ultimo step del modulo di normalizzazione del pitch, viene effettuato un nuovo *re-sampling* del segnale, così da riportare il segmento audio alla lunghezza originale, variando la frequenza di campionamento e dunque il suo pitch. Per effettuare il processo di campionamento in questo caso, poiché potrebbe accadere che il fattore di scala non sia intero, abbiamo preferito utilizzare una funzione di interpolazione lineare che, oltre a modificare la frequenza, funge da filtro passa-basso e rimuove il possibile *aliasing* generato nella trasformazione.

```
%Resample with linear interpolation                    (5.9)
outputTime = interp1((0:
(length(outputTimeStretched)-1)),outputTimeStretched,
(0:2^(step/12):(length(outputTimeStretched)-1)),'linear');
```

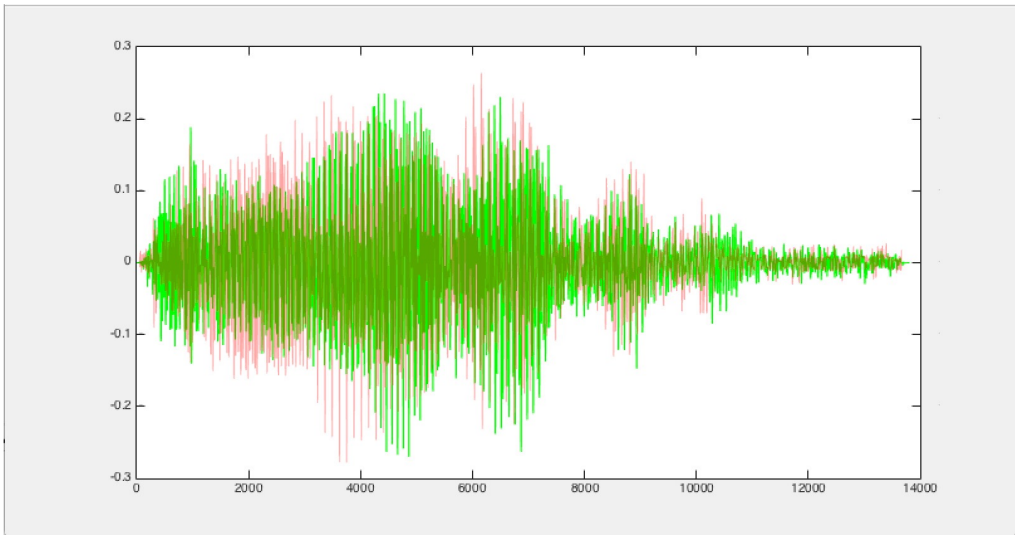


Figura 5.12 - Grafico che mostra l'andamento dell'onda sonora dopo la fase di normalizzazione del pitch. Il nuovo segnale (in verde) è confrontato con il vecchio segnale non normalizzato (in rosso) già mostrato nell'immagine 5.7

Capitolo 6

Architettura del sistema

*"Ogni tecnologia sufficientemente avanzata è indistinguibile dalla **magia**."*

Arthur Clarke

Lo scopo di questo capitolo sarà la costruzione di un modello di apprendimento del linguaggio che sia bioispirato e che costituisca un punto di partenza per la costruzione di un futuro sistema che integri, alle abilità linguistiche, quelle motorie e cognitive.

Come abbiamo già visto, sono diversi gli aspetti che portano l'essere umano a sviluppare correttamente la capacità di parlare e comprendere il linguaggio. Prima di tutto c'è lo sviluppo di abilità innate che avviene nei primi mesi, in cui i movimenti e i suoni sono involontari: questi movimenti sono profondamente legati alla crescita e alle modificazioni fisiologiche del tratto vocale che nei neonati è profondamente differente da quello degli adulti, in particolare per quanto riguarda la sua curvatura che segue la faringe, la quale si distende ed attenua in seguito alla crescita.

Associati a questi fattori biologici ci sono poi i vari meccanismi di apprendimento discussi in precedenza, che comprendono i sensi dell'udito e della vista, con i quali i bambini imitano movimenti e suoni percepiti nell'ambiente circostante.

Un sistema in grado di simulare l'apprendimento del linguaggio nel suo complesso dovrà dunque tenere conto di tutti questi fattori ed essere quindi integrato con quello cognitivo e motorio. Naturalmente un sistema del genere necessita di studi approfonditi nel corso di diversi anni e il modello che vogliamo proporre vuole fungere soltanto da punto di partenza per quanto appena descritto.

Prenderemo perciò in considerazione unicamente la parte relativa ai meccanismi di apprendimento che sfruttano il babbling e gli stimoli uditivi per passare dalle primitive linguistiche involontarie iniziali, alla pronuncia volontaria di parole che imitano quelle ascoltate.

I processi di analisi ed elaborazione dei suoni su cui ci siamo già concentrati saranno ora utilizzati per costruire un modello bioispirato che esegue operazioni assimilabili a quelle necessarie ad un bambino, nella fase di canonical babbling, per esplorare le proprie possibilità vocali e imitare i suoni che sente, per poi memorizzarli e riutilizzarli in modo volontario.

Per fare ciò, abbiamo sfruttato la vicinanza tra il babbling e le primitive motorie, introducendo il concetto di primitive del linguaggio, che corrispondono alle particelle utilizzate dai bambini nel babbling. Utilizzeremo gli stessi meccanismi già largamente impiegati e studiati nell'ambito dell'apprendimento dei movimenti, per lo sviluppo delle capacità linguistiche.

Una volta costruito questo modello in grado di simulare l'apprendimento linguistico di un bambino, studieremo come integrarlo a IDRA, un sistema cognitivo bioispirato già utilizzato in passato per lo sviluppo dei movimenti. In questo modo non solo si verificherà la possibilità di utilizzare, anche per il linguaggio, gli stessi meccanismi sfruttati per le primitive motorie, ma si valuterà la possibilità di integrare il nostro sistema linguistico ad un complesso sistema che imita il funzionamento del cervello umano per quanto riguarda l'apprendimento e nel quale sono già state studiate ed implementate le abilità relative al movimento.

6.1 Scelte implementative e struttura generale del sistema

Il modello costruito è suddiviso in due sottosistemi che corrispondono ai due momenti principali del percorso del bambino verso lo sviluppo del linguaggio: lo stadio involontario e lo stadio volontario.

Il primo sottosistema si occupa di quello che negli esseri umani è svolto dal normale sviluppo fisiologico del corpo a partire dai primi istanti di vita e rappresenta la componente innata del sistema. In questa fase vengono generate le primitive linguistiche grazie ai riflessi involontari, che spingono il bambino ad emettere suoni di vario tipo, i quali poi vengono memorizzati inconsciamente tramite una mappa mentale che collega azioni muscolari (respiro, movimento di corde vocali e bocca) ed effetto nel mondo fisico (suono prodotto). Queste primitive saranno poi riprodotte in vari modi e secondo diverse sequenze dettate principalmente dal caso e a loro volta mappate nel cervello, finché non inizierà a svilupparsi nel bambino la componente intenzionale, che lo porterà a poco a poco a controllare volontariamente i suoni prodotti.

Il sistema involontario dunque si occupa prima di tutto di generare le corrette primitive linguistiche prendendo in ingresso le tracce audio segmentate e normalizzandole per ottenere un dataset di suoni omogenei. In seguito, a partire dalle primitive iniziali, vengono generate tutte le concatenazioni possibili tra queste primitive, che rappresentano in ambito

biologico le effettive possibilità fisiche del bambino, esplorate tramite i suoni involontari prodotti nei primi mesi di vita. Infine sarà inizializzata la tabella corrispondente alla rete neurale biologica che contiene la mappa dei gruppi di suoni esplorati (ossia gli stati iniziali), necessaria all'apprendimento successivo.

Il secondo sottosistema rappresenta invece lo stadio più avanzato dello sviluppo del linguaggio, in cui avviene effettivamente il babbling, l'apprendimento di nuove parole e soprattutto l'imitazione volontaria da parte del bambino di una parola udita nell'ambiente circostante. In questa fase le abilità linguistiche vengono consolidate e il bambino impara sempre meglio a concatenare le primitive linguistiche in modo da imitare parole complesse in tempi sempre minori.

Il sistema volontario riceve quindi in ingresso un suono che rappresenta la parola da imitare e la struttura cognitiva inizializzata dal sistema precedente contenente gli stati, ossia i gruppi di suoni già esplorati. Il suono in ingresso sarà nuovamente elaborato in modo simile a quanto accade nel sistema involontario, così da renderlo omogeneo con il dataset di primitive e confrontato con ciascuno stato, così da identificare a quale gruppo di suoni appartenga. Una volta determinato lo stato attuale il sistema produce diversi suoni che rappresentano i numerosi tentativi svolti dal bambino nel tempo, col fine di imitare una determinata parola. I suoni prodotti vengono confrontati con quello da imitare e, quando si ottiene un suono abbastanza simile, il sistema si arresta e memorizza sia la nuova parola appresa che i tentativi fatti, il che ha un corrispettivo biologico nell'attivazione a livello neurale che avviene nel momento in cui il bambino pronuncia una parola simile a quella ascoltata [42].

Successivamente, di fronte ad una parola simile, il sistema non avrà più bisogno di molti tentativi per imitare la parola, ma saprà già quale suono produrre; nel caso in cui la parola fosse invece molto differente o appartenesse ad uno stato diverso, sarà attuato il meccanismo precedente fino a produrre una parola simile e memorizzarla, così da estendere le proprie abilità linguistiche nel tempo.

6.2 Il Sottosistema Involontario: predisposizione dell'ambiente di apprendimento

Il sistema involontario serve all'inizializzazione generale e ad eseguire una serie di operazioni che in natura sono svolte dal normale sviluppo fisiologico e che dunque sono considerate innate. Nel modello costruito non si è presa in considerazione la struttura fisica reale, né le varie fasi di sviluppo del corpo del bambino, perché non rilevanti ai fini di questo lavoro: sia il dataset iniziale che la mappa mentale si sviluppano in modo completamente involontario e non sono interessanti ai fini dell'analisi dell'apprendimento del linguaggio.

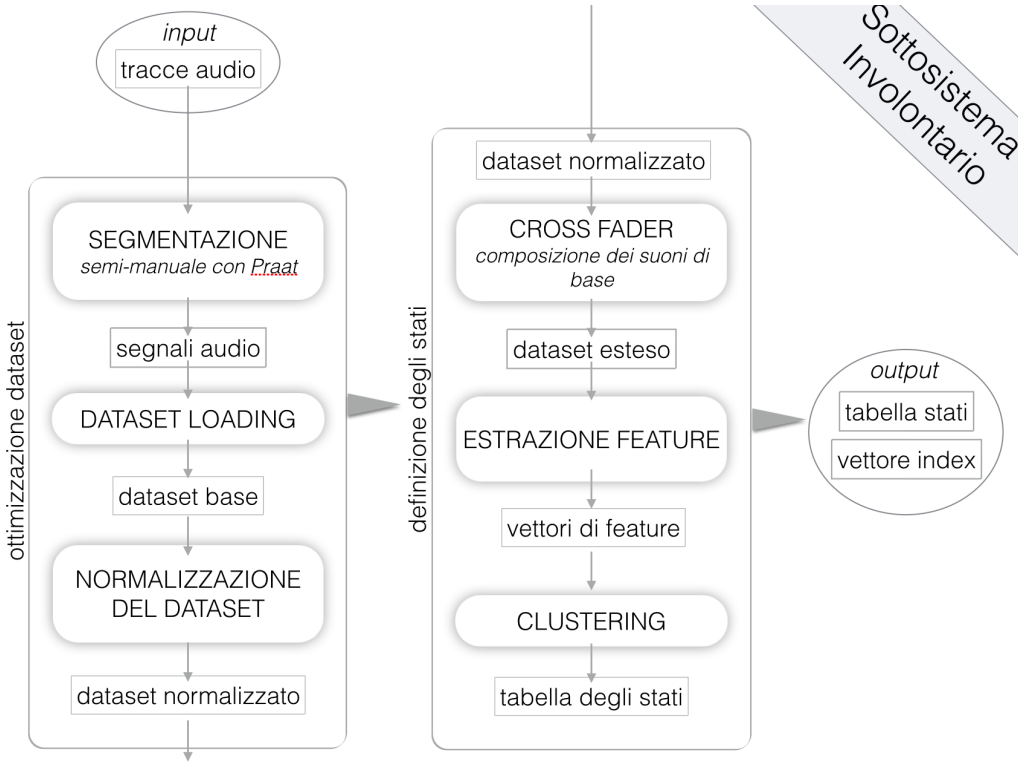


Figura 6.1 - Schema dell'architettura del sottosistema involontario. Il modulo di ottimizzazione prende in input i file audio e carica i segnali così da comporre il dataset iniziale, per poi eseguire la normalizzazione. Il secondo modulo invece prende in input il dataset di primitive linguistiche normalizzate, crea il dataset esteso, estrae le caratteristiche e ricava gli stati tramite clustering, restituendo in output la tabella stato-suono inizializzata e il vettore di indice.

6.2.1 Creazione dataset e Normalizzazione

Il primo problema da affrontare è quello di ottenere un dataset iniziale di primitive linguistiche. Questa parte è di fondamentale importanza perché i suoni selezionati in questo stadio saranno utilizzati in tutte le altre fasi ed elaborati numerose volte: per garantire dei risultati rilevanti è necessaria dunque una buona qualità delle primitive linguistiche di base.

Per eseguire questo compito utilizzeremo i metodi già studiati nel capitolo sull'analisi dei segnali con Matlab e Praat. Il sistema infatti riceve in ingresso i file audio (in formato WAVE), già segmentati in modo semi-manuale, che entrano nel modulo dedicato al caricamento dei segnali; questo li predispone in un'unica struttura contenente ciascun segnale e la rispettiva frequenza.

Una volta predisposta la struttura contenente il dataset iniziale completo, inizia il processo di normalizzazione che comprende numerose fasi, con lo scopo di rendere tutti i segnali audio omogenei. Vengono dunque normalizzati i canali, le frequenze, le ampiezze e infine il pitch, riducendo anche il rumore di fondo, così da ottenere delle tracce più chiare possibili.

Questo processo è indispensabile sia se si estrae il dataset da registrazioni di bambini diversi, sia se si registra dal vivo in una singola occasione: bastano infatti piccole differenze nel tono vocale (che sono fisiologiche sia nei bambini che negli adulti) o nell'ampiezza (nel caso di soggetti in movimento) per rendere le fasi successive difficili da eseguire; la stessa cosa vale per il rumore di fondo, dovuto ad un contesto non ottimale o ad una scarsa qualità dei mezzi di registrazione. La normalizzazione del dataset è considerata quindi anche in vista di possibili sviluppi futuri in cui si testi il sistema in un ambiente fisico e dunque non ideale e con un robot reale.

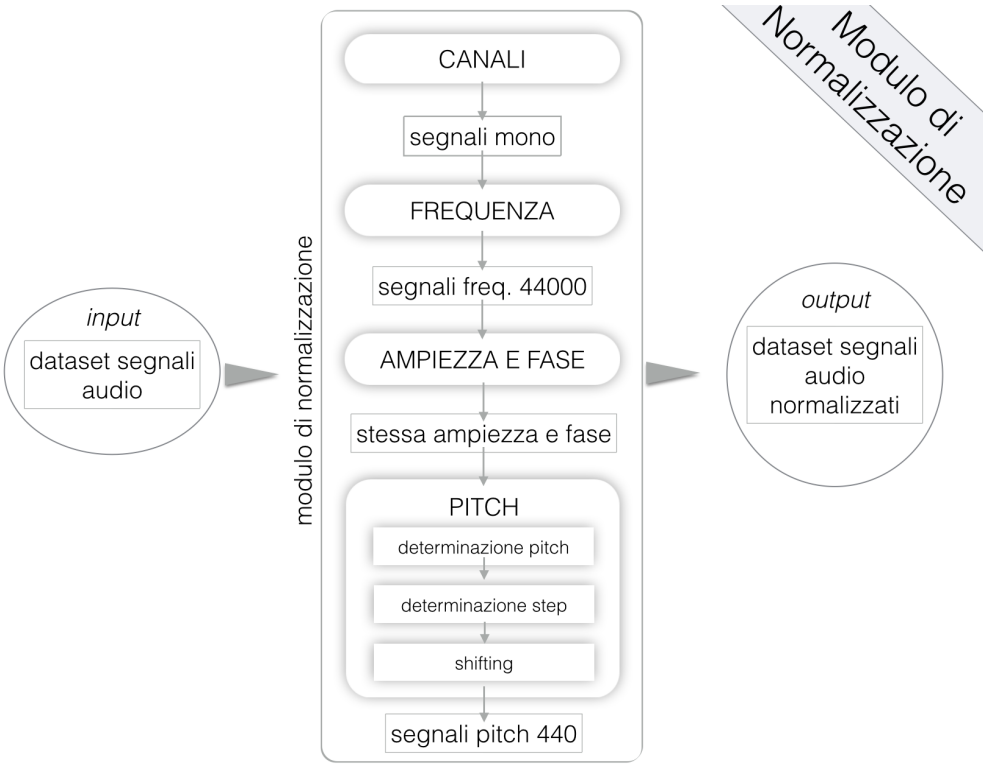


Figura 6.2 - Schema dell'architettura del modulo di normalizzazione che prende in input il dataset di primitive linguistiche e le rende omogenee normalizzando numerosi parametri. Il modulo restituisce in output un dataset di primitive normalizzato.

6.2.2 Concatenamento tramite Cross-Fading

I segnali audio adeguatamente normalizzati andranno a costituire il dataset iniziale che contiene le primitive linguistiche di base. Queste primitive rappresentano i suoni involontari inizialmente prodotti dai bambini in seguito a riflessi involontari della muscolatura e poi appresi, a poco a poco, ed utilizzati nel processo di imitazione per comporre suoni più complessi.

La composizione di più primitive linguistiche avviene nel bambino in modo inizialmente casuale e viene definita esplorazione vocale; soltanto dopo alcuni

tentativi, in seguito al processo di apprendimento, l'imitazione avviene in modo più mirato. Dunque, per fornire al sistema un dataset di primitive linguistiche composte, è stato predisposto un modulo a parte che esegue il concatenamento dei suoni iniziali secondo tutte le combinazioni possibili, che corrispondono a tutti i possibili suoni prodotti dal bambino. Anche questo processo è considerato innato perché deriva dalle possibilità fisiche e non da un processo di apprendimento; questo dataset esteso sarà poi utilizzato, durante la fase di apprendimento, per l'esplorazione vocale.

Il modulo di concatenamento dunque ha un ruolo fondamentale nel nostro modello perché permette di concatenare le primitive linguistiche elementari e generare nuove primitive, con un processo simile a quello che avviene per le primitive motorie nell'ambito dell'apprendimento motorio.

Come detto in precedenza il bambino, durante la fase di babbling, crea una mappa mentale tra suono prodotto e movimento necessario a farlo. I suoni elementari appresi (es: "ba", "ga") vengono riprodotti in rapida sequenza durante la fase di consolidamento, in cui il bambino passa progressivamente ad una gestione volontaria dei suoni prodotti. Questi suoni consolidati vengono poi riprodotti volontariamente in tutti i casi in cui il bambino prova ad imitare una parola ascoltata nell'ambiente circostante, concatenandoli in varie sequenze fino a giungere alla parola esatta o ad un suono molto simile, composto con le primitive motorie a disposizione.

Il concatenamento non viene effettuato semplicemente fondendo i due segnali, ossia unendoli, uno dopo l'altro, in un nuovo segnale perché, benché siano stati normalizzati, potrebbero comunque essere presenti piccole variazioni nell'ampiezza o nella fase che porterebbero ad un segnale disturbato o a discontinuità sonore.

Per evitare questi inconvenienti abbiamo utilizzato un algoritmo di *cross-fading*, che permette di ottenere una transizione pulita e lineare da un suono all'altro, anche in presenza di discontinuità [55].

Questo meccanismo viene molto utilizzato anche in ambiente musicale e prevede di applicare due fader ai suoni, esattamente nel punto di connessione. Ogni fader sfuma il suono su cui è applicato in modo lineare: il primo riduce progressivamente il primo segnale audio, il secondo agisce in modo esattamente inverso, aumentando progressivamente l'intensità del secondo segnale. In questo modo, sovrapponendo i due segnali nel punto di transizione dall'uno all'altro, si otterrà un output costante e le due onde saranno unite in modo progressivo e privo di discontinuità, con il risultato di avere i suoni perfettamente mixati.

Per eseguire questo procedimento l'algoritmo applica ai due suoni una *cross-fader mask*, formata da due funzioni rampa simmetriche, come mostrato in Figura 6.3.

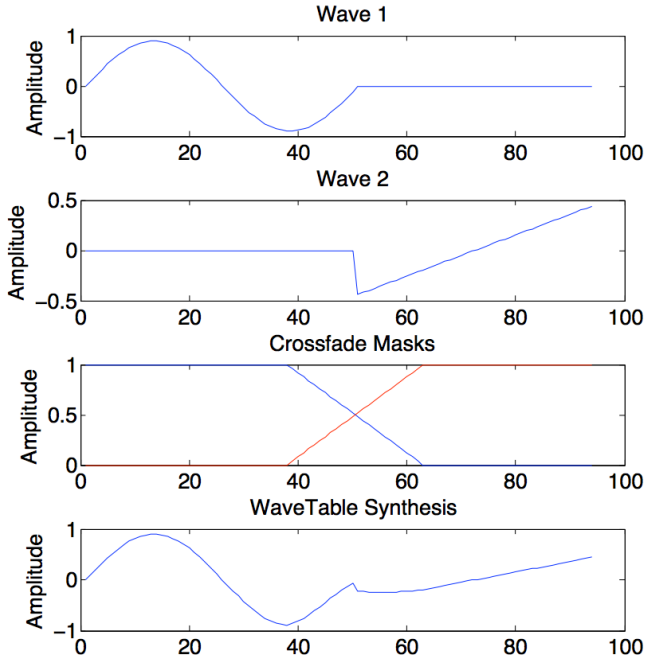


Figura 6.3 - L'onda 1 e l'onda 2 vengono unite grazie alla maschera formata da due funzioni rampa, applicata nel punto di passaggio, riducendo al minimo le possibili discontinuità tra i due segnali.

6.2.3 Estrazione delle feature dai suoni in modo automatico

Nel modulo di normalizzazione il dataset di primitive linguistiche viene sottoposto a varie elaborazioni col fine di ottenere un insieme di file sonori con caratteristiche omogenee, così da poter essere concatenati insieme nel modulo di *cross-fading* ed ottenere un dataset esteso.

Per poter ottenere gli stati mentali iniziali e sottoporre i suoni al sistema di apprendimento volontario è tuttavia necessaria un'ulteriore elaborazione: la *features extraction* [55].

Questo task è molto importante in tutti i processi di *pattern recognition* che riguardano i suoni, perché permette di ricavare, da un segnale complesso come può essere un segnale audio, un insieme di caratteristiche semplici in grado di descriverlo.

In questo modo, nei processi di *matching* e di *machine learning* non è necessario gestire tutta la complessità relativa al file originale, ma si può lavorare unicamente utilizzando le feature estratte, che rappresentano il solo contenuto informativo rilevante per le operazioni da compiere.

In caso di dati voluminosi l'estrazione delle feature può essere visto anche come un processo di riduzione dei dati, perché bastano poche feature per descrivere i dati di partenza e poter effettuare confronti validi.

Il sistema, nel modulo adibito all'estrazione delle feature, prevede due passaggi:

1. *Mid-Term Windowing*
2. *Short Term Processing*

Mid-Term Windowing

Nel primo passaggio il segnale audio è suddiviso in diversi segmenti di media durata (*windows*), ciascuno dei quali viene poi sottoposto, nel secondo passaggio, ad un processo di S*hort Term Features Processing,* in cui viene ricavato il vettore di caratteristiche corrispondente; questo vettore è utilizzato per computare un set di statistiche che descrive il segmento *mid-term* iniziale.

Data la tipologia dei segnali audio e la loro brevità (i segnali presenti nel dataset esteso sono infatti composti da due sole primitive linguistiche) si è scelto di estrarre un'unica finestra di mid-term da ogni file audio, ossia sottoporre l'intero segnale all'estrazione delle feature, senza dividerlo nuovamente in parti. La suddivisione in diverse finestre mid-term è comunque implementata nel sistema (anche se non utilizzata in questa fase) perché si potrà rivelare molto utile in caso di apprendimento di parole complesse e dunque più lunghe.

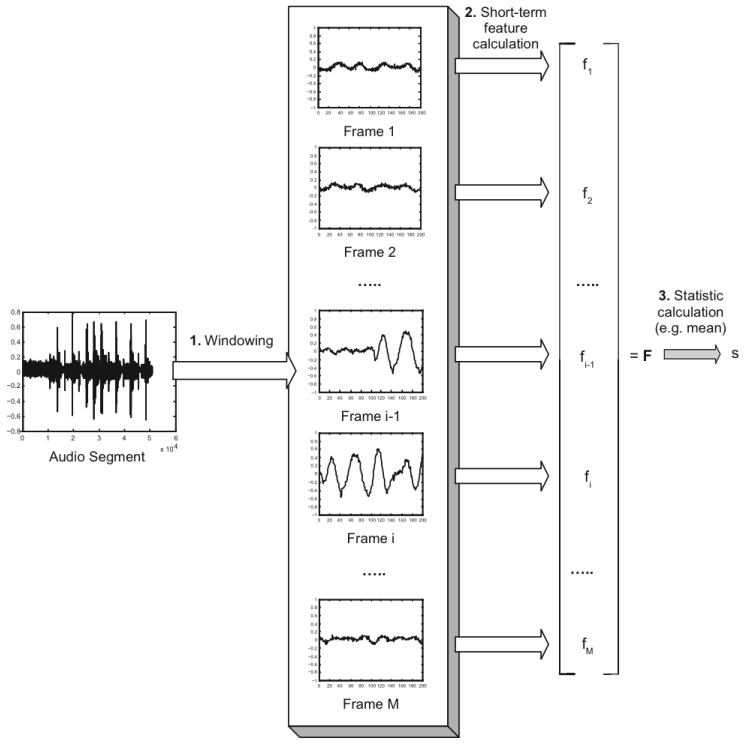


Figura 6.4 - Lo schema mostra l'estrazione delle feature sulla base di un processo mid-term. Ogni segmento mid-term è processato e vengono estratte le sue caratteristiche tramite un algoritmo di short-term features extraction.

Short-Term Processing

Lo *Short Term Processing*, che viene applicato ad ogni segmento iniziale, segue lo stesso meccanismo di *framing* descritto nel paragrafo riguardante il pitch shifting: ogni singolo segmento mid-term (che nel nostro caso è dunque l'intero segnale audio) viene suddiviso in numerosi segmenti più piccoli sovrapposti. Ognuno di questi frame viene poi processato per generare un set di feature corrispondenti che lo descrivono e dalle quali saranno estratte le statistiche finali, poi utilizzate nei moduli seguenti dedicati all'apprendimento.

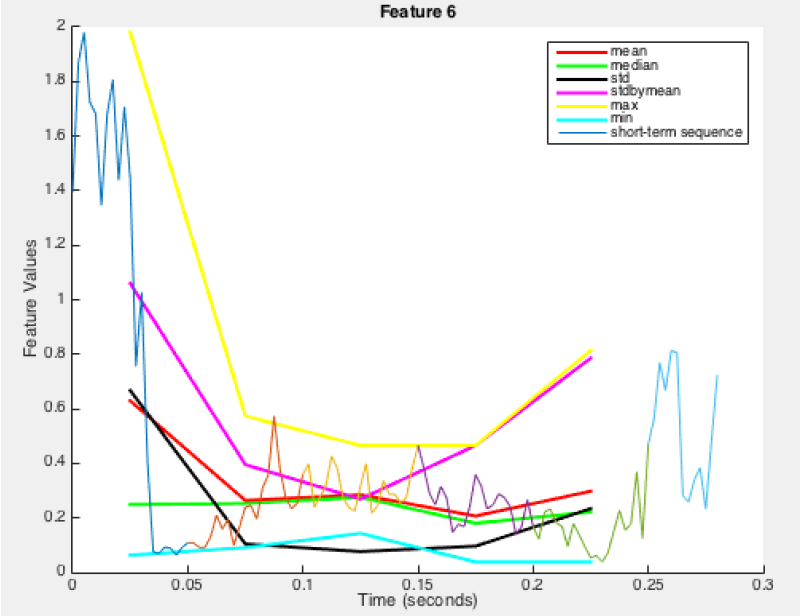


Figura 6.5 - Nell'immagine vengono mostrate in forma grafica le sei feature estratte dalla primitiva linguistica "ga".

Il sistema è in grado di estrarre numerose tipologie di feature differenti da un singolo segmento *short-term*, da utilizzare in base alle caratteristiche dei suoni che devono essere elaborati: ogni specifica feature ha infatti un significato fisico diverso e descrive un determinato aspetto dell'onda di partenza.

Le feature estratte si dividono in due categorie distinte:

- *feature che utilizzano il dominio del tempo (es: energy, zero-crossing, entropy)*
- *feature che utilizzano il dominio delle frequenza (es: spectral centroid, spectral entropy, spectral flux, spectral rolloff, cepstrum, chroma vector)*

Mentre la prima tipologia valuta direttamente la forma dell'onda audio e la descrive identificando i punti notevoli, la seconda trasforma l'onda in frequenze ed utilizza lo spettro per ricavare caratteristiche utili a descrivere l'onda di partenza, ottenendo dunque maggiori dettagli.

Infine, utilizzando i vettori di feature estratti per ciascun segmento di short-term (di lunghezza variabile in base al numero di caratteristiche che si è scelto di estrarre), vengono calcolate le statistiche generali che descrivono l'intero segmento mid-term e dunque l'intero segnale audio.

Anche in questo caso il sistema è in grado di estrarre numerose tipologie di statistiche (ad esempio media, deviazione standard, massimo, minimo) da utilizzarsi a seconda delle caratteristiche del suono che si sta processando. In base al numero di statistiche scelte, si otterrà il vettore finale che descrive l'intero segmento audio. Per esempio, se si sceglie di estrarre 34 feature differenti da ciascun segmento di short-term e 2 statistiche per ciascun segmento di mid-term, ogni segmento audio sarà descritto da un vettore di 68 valori.

6.2.4 Clustering e identificazione degli stati del sistema

Il dataset iniziale, composto dalle primitive linguistiche di base non ulteriormente scomponibili, è stato inizialmente processato in vari step che lo hanno normalizzato per rendere i suoni omogenei tra di loro e poi utilizzato dal modulo di *cross-fading* per ottenere un nuovo dataset esteso, contenente tutte le possibili combinazioni dei suoni di base. Infine sono state estratte le feature per ogni suono presente in questo nuovo dataset, così da ridurre la complessità computazionale necessaria agli stadi successivi di apprendimento.

A questo punto, prima di poter passare al sistema volontario, è necessario che il sistema estragga gli stati iniziali di partenza, che corrispondono ai diversi gruppi di suoni involontari simili, identificati inconsciamente dal bambino. Gli stati così identificati serviranno ad inizializzare la tabella che corrisponde alla mappa mentale sviluppata dal bambino, utilizzata poi nel processo di apprendimento per collegare un suono ascoltato ad una particolare azione fisica, che nel nostro caso corrisponde ad una diversa parola pronunciata.

Per identificare gli stati di partenza è necessario processare il dataset (espresso sottoforma di feature) con un algoritmo di *clustering* che sia in grado, a partire dai dati iniziali, di raggrupparli in base alla similarità. Ogni diverso gruppo dovrà quindi contenere suoni con caratteristiche molto simili, che corrispondono ai diversi tentativi di imitazioni.

Poiché il raggruppamento deve essere effettuato su dati descritti da numerose feature e dunque da molte dimensioni (ogni feature corrisponde ad una dimensione differente), si è scelto di utilizzare *K-Means*, un'algoritmo di clustering partizionale molto utilizzato e piuttosto veloce [59]. Inoltre questo algoritmo possiede un'implementazione molto efficiente in Matlab, che permette di ottenere buone soluzioni in tempi brevi e utilizza un metodo euristico per trovare sottogruppi non ottimi, ma che minimizzano la distanza (ossia la somiglianza) tra gli elementi in ognuno di essi. Solitamente K-Means

viene utilizzato in casi come il nostro in cui gli oggetti, dunque i suoni, sono rappresentati come vettori e le cui distanze possano essere calcolate in uno spazio vettoriale a più dimensioni.

L'algoritmo segue un processo iterativo e prende in input i dati iniziali non partizionati e il numero di cluster (nel nostro caso di stati) che si vogliono ottenere. K-Means non è in grado di determinare automaticamente qual è il numero perfetto di cluster da utilizzare per ottenere il partizionamento migliore ed è necessario eseguire diverse volte l'algoritmo e confrontare i risultati ottenuti scegliendo il migliore. Per ottenere una stima del numero di cluster più efficiente, si possono utilizzare degli algoritmi euristici (ad esempio Silhouette) che misurano la somiglianza degli oggetti in ciascun gruppo e restituiscono un parametro di qualità del clustering [59].

In K-Means ogni cluster è identificato da un centroide, ossia un vettore con caratteristiche più simili possibile a ciascun elemento contenuto nel suo gruppo di appartenenza. Una volta determinato il numero di cluster K-Means inizia assegnando un valore casuale a ciascun centroide, dopodiché effettua il primo partizionamento calcolando la distanza di ciascun elemento da ogni centroide ed assegnando alla stessa partizione gli oggetti più vicini al centroide che la descrive. Si possono utilizzare diverse metriche per calcolare la distanza, come ad esempio la distanza euclidea, in base alle quali cambiano sia le prestazioni dell'algoritmo che la qualità.

A questo punto, ottenuti i diversi cluster, l'algoritmo calcola il valore medio degli elementi di ogni partizione e lo imposta come nuovo centroide del gruppo. Il centroide così determinato serve a calcolare nuovamente le distanze tra tutti gli elementi e i diversi centroidi, ripetendo la fase precedente e calcolando nuovi centroidi. Queste due fasi vengono iterate finché l'algoritmo non converge ad una soluzione.

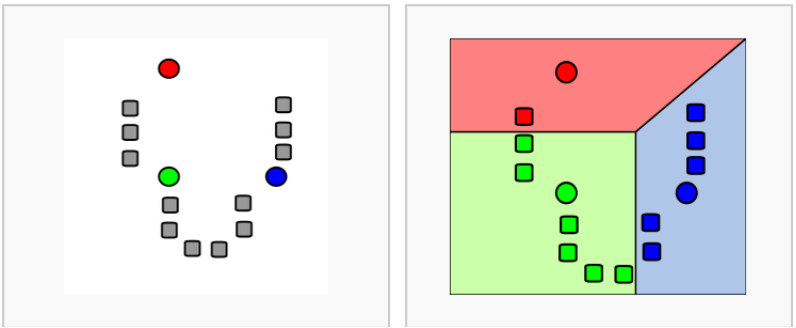


Figura 6.6 - Nella figura a sinistra sono mostrati gli elementi inseriti in K-Means (in questo caso utilizzato su elementi di due sole dimensioni) e i centroidi iniziali, calcolati in modo casuale. Nella figura di destra viene mostrato il primo partizionamento, eseguito associando nella stessa partizione elementi vicini al centroide caratteristico.

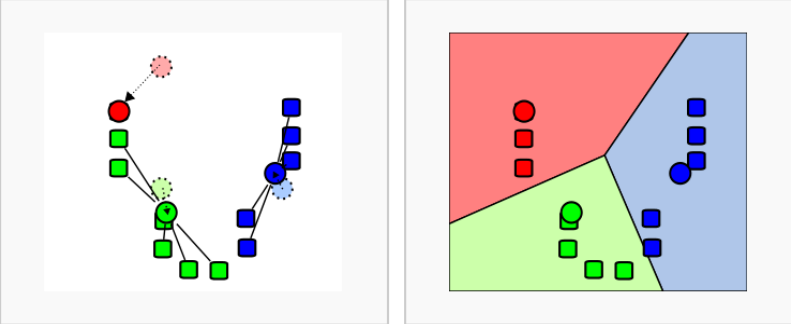


Figura 6.7 - Nella figura a sinistra viene calcolato, partendo dagli elementi di ciascuna partizione, il nuovo centroide caratteristico. Nella figura di destra viene mostrato il nuovo partizionamento effettuato a partire dai nuovi centroidi. Queste ultime due fasi si ripetono finché l'algoritmo non converge.

Alla fine del processo il modulo di clustering restituirà un'insieme di centroidi, che corrispondono all'insieme di stati cercati, ognuno descritto da un vettore di feature.

Gli stati ottenuti saranno invece inseriti nella tabella stato-suono (composta da tante righe quanti gli stati), inizialmente vuota, che sarà inizializzata e passata al sistema volontario di apprendimento del linguaggio, insieme ad vettore, anch'esso già inizializzato ma vuoto, che conterrà i diversi tentativi svolti e le parole apprese.

6.3 Il Sottosistema Volontario: imitazione ed apprendimento di una parola

Grazie ai moduli che compongono il sottosistema involontario, partendo da file audio o da registrazione è possibile generare un set di primitive linguistiche di base appositamente normalizzate per essere confrontabili l'una con le altre. Queste primitive possono essere fuse tra loro tramite concatenamento nel modulo di cross-fading, ottenendo un dataset più ampio rappresentante tutte le possibilità vocali del bambini. In seguito vengono estratte delle feature dalle primitive che permettono di descriverle in modo semplificato, così da poter effettuare le elaborazioni successive. Infine le feature vengono sottoposte a clustering e vengono identificati gli stati iniziali di partenza da utilizzare nel processo di apprendimento per tentare di imitare una parola ascoltata, esattamente come farebbe un bambino nella fase di canonical babbling.

Nei capitoli precedenti siamo giunti alla conclusione che l'integrazione dell'apparato responsabile del linguaggio con i sistemi relativi all'apprendimento cognitivo e motorio sarebbe la soluzione ideale per tentare di superare il Symbol Grounding Problem e creare un sistema in grado di

estrapolare la conoscenza necessaria a supportare i propri concetti mentali direttamente dagli stimoli sensoriali ricevuti.

Seguendo questo filo conduttore abbiamo scelto di ispirarci, nell'implementazione del sistema volontario responsabile dei processi di apprendimento, al sistema IDRA, un software profondamente bioispirato in grado di generare autonomamente nuovi obiettivi a partire da stimoli sensoriali e da esperienze passate, sfruttando una rete neurale di moduli ispirati alle diverse componenti del cervello.

Fino ad ora questo software era stato utilizzato per generare obiettivi riguardanti il movimento: al suo interno è presente infatti un sistema che simula l'apparato motorio e, utilizzando il concetto di primitive motorie, è in grado di decidere quale sia il movimento migliore da eseguire in una data situazione, perseguendo determinati obiettivi.

Il sottosistema volontario da noi realizzato implementa quindi un meccanismo simile a quello sfruttato da IDRA per l'apprendimento dei movimenti, ma facendo uso delle primitive linguistiche, così che sia semplice poi integrare insieme i due sistemi. Questo permette non solo di verificare l'ipotesi che si possano utilizzare gli stessi meccanismi mentali sia per i movimenti che per il linguaggio, ma costituisce un primo passo verso l'implementazione di un modello cognitivo in grado di gestire, tramite gli stessi processi, diversi aspetti della mente umana, come quello linguistico, motorio e sensoriale.

6.3.1 Apprendimento tramite le primitive linguistiche

Il sottosistema volontario, dedicato all'apprendimento e all'imitazione, è composto da due moduli: il primo serve a ricevere e a preparare il suono (ossia la parola) che deve essere imitato, mentre il secondo esegue effettivamente il processo di babbling e impara dall'esperienza.

La parola da imitare può essere registrata direttamente con Matlab (utilizzando i *toolbox* preposti) oppure estratta da un file audio, tramite un processo di segmentazione analogo a quello già utilizzato per selezionare le primitive. Inserito nel sistema, il segnale audio dovrà essere sottoposto ad una normalizzazione identica a quella a cui sono state sottoposte le primitive, col fine di avere un suono il più chiaro possibile e con una tonalità molto simile a quella dei segnali contenuti nel dataset esteso. Questo è necessario perché l'apparato di registrazione ed elaborazione audio di un computer o di un robot non è assolutamente paragonabile, in termini di prestazioni e qualità, a quello biologico, benché sia di un bambino. Questa fase di acquisizione e normalizzazione è svolta dal primo modulo, che è incaricato di passare al secondo la parola che bisogna tentare di riprodurre.

Il secondo modulo, in cui viene appresa la parola e riprodotta tramite babbling, prende dunque in ingresso sia la parola da riprodurre che gli output del sottosistema involontario: la tabella stato-suono, il vettore di indice e il dataset esteso di tutte le combinazioni di suoni effettivamente realizzabili.

A questo punto, per permettere le successive elaborazioni, dalla parola in ingresso (da ora in poi definita parola target) viene estratto il vettore di feature e confrontato con i vettori di feature dei centroidi rappresentanti gli stati. Per ogni stato viene calcolata la distanza tra il centroide stesso e la parola target, con una metrica analoga a quella utilizzata dall'algoritmo K-Means che ha creato gli stati. Lo stato che ha la distanza minore con la parola target diventa quello corrente e rimarrà attivo per tutto il processo di apprendimento della parola, finché non si ottiene un buon risultato e viene eseguito il babbling.

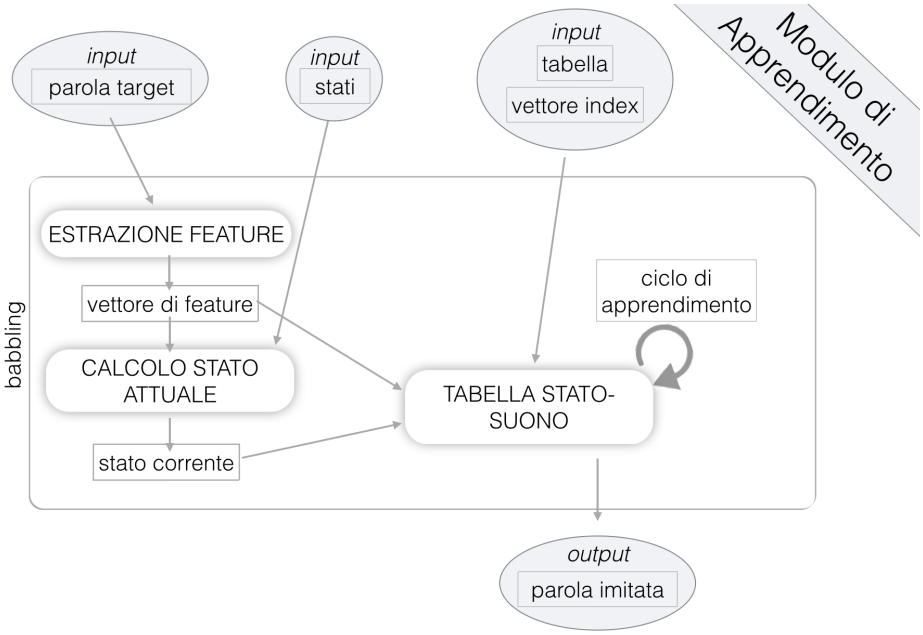


Figura 6.11 - Schema dell'architettura del sottosistema volontario. Il modulo find babble prende in input il file audio che andrà a costituire la parola target che si deve provare ad imitare e la normalizza, passandola al modulo babbling che esegue l'apprendimento. Il sottosistema involontario genera il dataset esteso di suoni normalizzati, la tabella stato-suono inizializzata e il vettore di indice e lo passa al modulo di babbling.

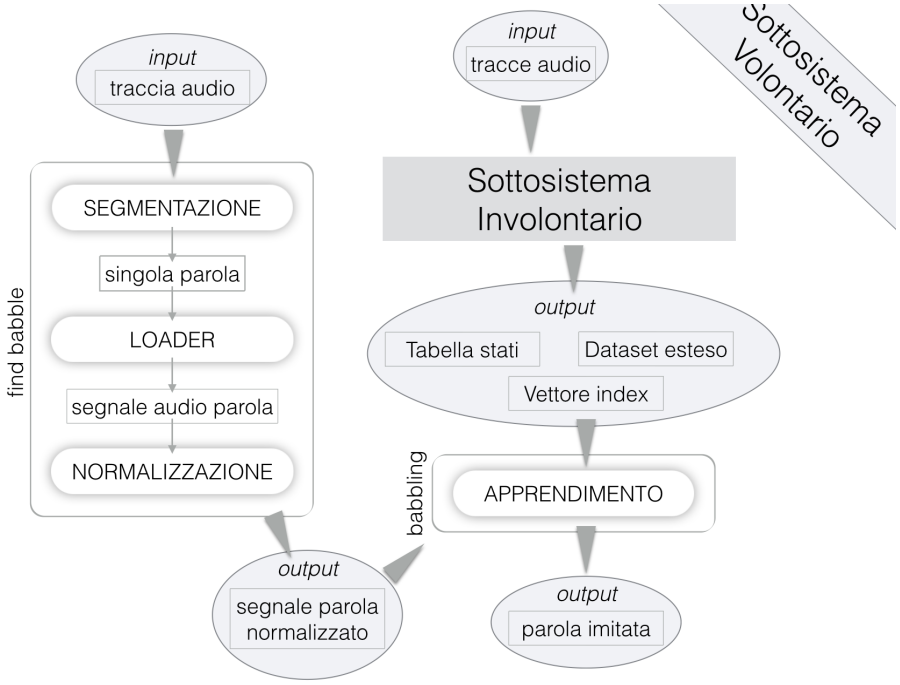


Figura 6.12 - Schema dell'architettura del modulo di babbling che esegue l'apprendimento. La parola passata in input dal modulo find babbling viene processata per estrarne le feature e calcolare lo stato corrente. La tabella stato-suono, proveniente dal sottosistema involontario, viene riempita attraverso i cicli di apprendimento finché non viene generata la parola imitata.

A questo punto il sistema utilizza la tabella stato-suono che rappresenta una rete neurale e che viene utilizzata per registrare i tentativi di imitazione svolti e dedicata all'apprendimento delle prime parole. Ogni riga rappresenta uno stato differente, mentre ogni colonna corrisponde ad un diverso suono prodotto.

Per ogni nuova parola target, una volta identificato lo stato corrente, si utilizzerà unicamente quella riga per registrare i diversi tentativi secondo i seguenti casi:

- Se il vettore di indice è vuoto (cosa che avviene soltanto alla prima parola appresa), il sistema genera in modo casuale diversi tentativi di babbling, ognuno dei quali è estratto dal dataset esteso di primitive e dunque è fisiologicamente sensato perché rientra nell'insieme dei suoni possibili da eseguire. Per ogni suono generato viene creata una nuova colonna della tabella e viene registrato all'interno del vettore indice. Inoltre viene calcolata la vicinanza tra il suono (da cui sono estratte le feature) e quello target. Una volta nota la distanza, si estrae la probabilità che i due suoni rappresentino la stessa parola e questa viene salvata nella tabella, nella riga e nella colonna corrispondenti rispettivamente allo stato attuale e al suono generato. Se la probabilità supera una certa soglia definita a priori, il sistema termina il ciclo di apprendimento, altrimenti si fa un altro

tentativo identico. Si continua in questo modo finché non viene generato un suono accettabilmente simile.

- Se il vettore di indice contiene già dei valori, ma nessuno di questi è stato testato con lo stato attuale e dunque con una parola target simile a quella corrente, si calcola la distanza tra ogni suono imparato e quello attuale. Questa distanza viene registrata nella tabella, nella colonna relativa al suono già precedentemente appreso e nella riga relativa allo stato corrente. Vengono testati subito i suoni già appresi perché sono già in memoria: è dunque immediato confrontarli con il nuovo suono alla ricerca di una corrispondenza. Se uno di questi suoni è abbastanza simile alla parola target l'elaborazione termina e viene eseguito il babbling. Altrimenti, testati tutti i suoni già presenti nel vettore indice, se ne generano di nuovi.

- Se il vettore di indice contiene già dei valori e alcuni di essi sono stati testati con lo stato attuale e dunque con parole simili alla parola target, viene testato prima di tutto il suono con il valore di somiglianza maggiore, poi tutti gli altri suoni il cui valore di somiglianza supera una certa soglia e infine, nel caso in cui nessuno dei suoni già imparati sia abbastanza simile alla parola target corrente, si generano nuovi suoni in modo casuale, finché non ne viene individuato uno adeguato.

Dunque, ad ogni iterazione con una nuova parola, la tabella stato-suono viene riempita con nuovi suoni appresi, uno per ogni colonna, registrati nel vettore di indice. Per ogni stato vengono dunque registrate nella tabella le varie probabilità di appartenenza suono-stato e naturalmente maggiore è il numero delle prove svolte, più alta è la probabilità di trovare subito una buona imitazione per la parola target, con conseguente riduzione del tempo di elaborazione.

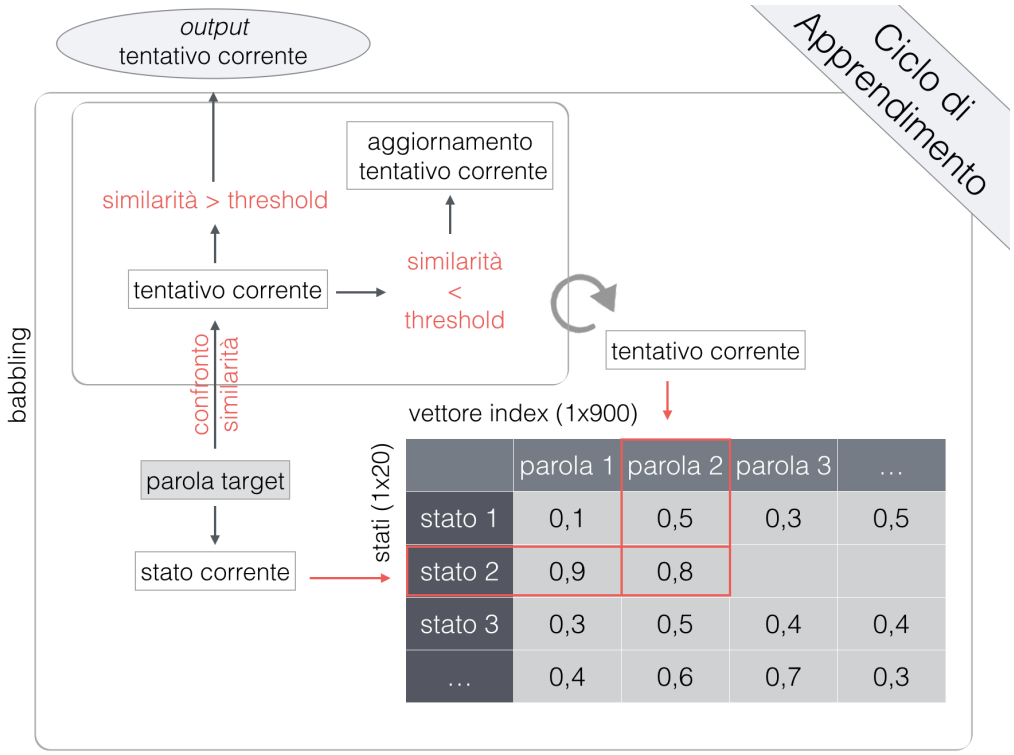


Figura 6.13 - Schema dell'architettura del modulo di apprendimento e del funzionamento della tabella stato-suono. Nel caso mostrato il sistema sta tentando di imitare la parola target utilizzando un suono già appreso per cui non era noto il grado di similarità nello stato corrente.

6.4 Integrazione del Modello delle Primitive Linguistiche con IDRA

6.4.1 Architettura di IDRA

La capacità propria degli esseri umani di sviluppare numerose abilità a partire da diversi input sensoriali, generando automaticamente un proprio bagaglio di conoscenze passate, è una delle caratteristiche della mente più misteriose e complesse da comprendere.

Il cervello permette infatti all'uomo di adattarsi a diverse situazioni e costruire la sua esperienza nel tempo, sia che si tratti dell'ambito motorio, che di quello linguistico o logico; sebbene questo organo resti ancora perlopiù sconosciuto, sono numerose le aree di cui è chiara, perlomeno in parte, la funzione.

Sfruttando queste conoscenze è stato creato IDRA (*Intentional Distributed Robotic Architecture*), che si fonda su processi che simulano quelli interni al cervello umano per sviluppare nuovi obiettivi sulla base di input sensoriali e istinti innati [60]. Questa architettura cerca di imitare tre precise aree cerebrali

coinvolte nell'apprendimento e nella generazione di nuovi obiettivi: corteccia cerebrale, talamo, amigdala.

La corteccia cerebrale è l'area più recente, dal punto di vista evolutivo, che si è sviluppata nel cervello dell'uomo ed è la sezione che più di tutte contraddistingue l'essere umano dalle altre specie animali perché assolve a molte funzioni legate alla coscienza, al ragionamento e al controllo volontario. Ad esempio, sono proprio le aree associative della corteccia che permettono di avere una rappresentazione mentale del mondo circostante e su di esse si fonda la percezione stessa.

Gli stimoli sensoriali infatti, passando per il talamo, vengono poi elaborati dalle aree sensoriali della corteccia, messi in relazione con le esperienze passate già immagazzinate e poi memorizzati, permettendo di ottenere un bagaglio di informazioni nuovo su cui fondare le decisioni future e il significato stesso dei concetti mentali.

Il talamo è profondamente collegato alla corteccia (tramite i circuiti talamo-corticali) e svolge un ruolo molto importante nella coscienza. Agisce inoltre da filtro per gli input sensoriali, determinando quali passare ai centri superiori di elaborazione e all'apparato di memorizzazione.

A completamento di questo sistema c'è l'amigdala, un gruppo di nuclei da cui partono numerosi collegamenti verso le altre aree del cervello, che sono implicati nelle risposte emozionali forti come la paura, che sono legate ad istinti innati e alla modulazione di alcune funzioni cognitive, nonché della memoria stessa.

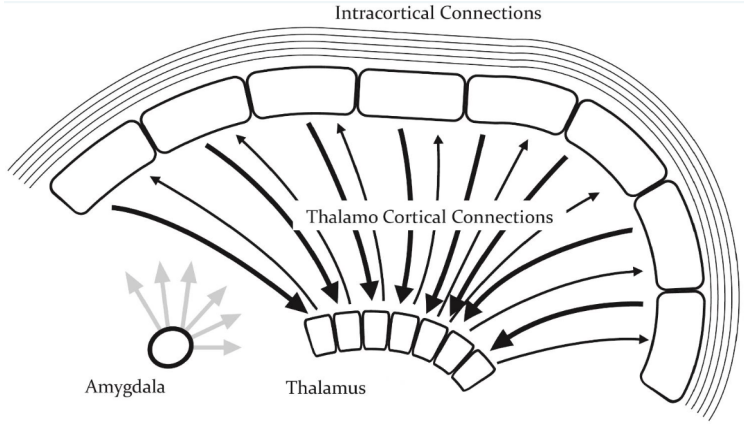


Figura 6.8 - Schema che mostra l'interazione tra corteccia, talamo e amigdala nel cervello. Talamo e corteccia sono strettamente collegati dalle connessioni talamo-corticali, tramite le quali vengono trasmessi gli stimoli sensoriali. L'amigdala modula la trasmissione dei segnali sensoriali utilizzando conoscenza pregressa e istinti innati.

Per simulare il funzionamento di questo apparato composto da corteccia, talamo e amigdala, IDRA utilizza numerosi moduli collegati tra di loro in una

rete neurale con numerosi livelli. L'elemento più importante all'interno del sistema è il Modulo Intenzionale *(IM)* che rappresenta la corteccia e il talamo, le cui funzioni sono svolte rispettivamente dal sotto-modulo di Categorizzazione *(CM)* e da quello Ontogenetico *(OM)*.

Il sotto-modulo di Categorizzazione funge da memoria per gli eventi passati e possiede una capacità associativa di base in grado di categorizzare gli stimoli in entrata. Il sotto-modulo Ontogenetico invece è centrale nella generazione di nuovi obiettivi ed è in grado di valutare la relazione tra lo stimolo corrente e quelli passati: il modulo OM riceve dal modulo CM un vettore che descrive l'attivazione neurale in risposta ad un input e, utilizzando una funzione di apprendimento hebbiano, sviluppa nuovi obiettivi.

L'amigdala invece è esterna ai moduli intenzionali ed è modellizzata dal Modulo Filogenetico Globale, nel quale sono contenute informazioni relative agli istinti innati: questo modulo, sulla base degli input giunti dall'esterno, è in grado di generare un segnale che indichi quanto questi siano conformi agli obiettivi innati, ossia quanto gli input siano interessanti per il sistema.

Mentre il modulo Filogenetico è unico, in IDRA sono presenti numerosi moduli Intenzionali (che costituiscono l'Architettura Intenzionale), disposti su vari livelli e collegati tra di loro per mezzo di link vettoriali e scalari, diretti o in retroazione, in base alle esigenze. Quando uno stimolo sensoriale viene rilevato, viene filtrato ed inviato ai moduli Intenzionali del primo livello; parallelamente viene inviato anche al Modulo Filogenetico Globale che calcola il segnale filogenetico da passare all'Architettura Intenzionale.

A questo punto ogni Modulo Intenzionale riceve in ingresso un input (proveniente direttamente dall'esterno oppure proveniente da altri livelli di moduli intenzionali) e il segnale filogenetico. Questi due segnali vengono processati e inviati in output agli altri moduli sotto forma di altri due segnali distinti: un vettore che codifica l'input iniziale e uno scalare che indica il grado di interesse del modulo verso l'input.

Infine il risultato dell'elaborazione di tutti i moduli viene passato all'apparato motorio che, tramite il meccanismo delle primitive motorie, sceglie qual'è il movimento migliore da effettuare.

Questa architettura distribuita in più moduli, divisi in numerosi livelli, permette a IDRA di maneggiare correttamente qualsiasi tipo di dato in input: che sia uno stimolo visivo, uditivo o tattile sarà correttamente elaborato ed interpretato. Tale modo di operare è simile a quello utilizzato realmente dal cervello, che grazie all'immensa rete di neuroni di cui è formato, è in grado di adattarsi correttamente e trasformarsi in base agli stimoli ricevuti, apprendendo e specializzandosi nel tempo per svolgere determinati compiti.

L'enorme volubilità del sistema ci permette di utilizzarlo anche con il nostro modello linguistico che fa uso di stimoli uditivi e di primitive linguistiche e non di stimoli visivi e primitive motorie; permetterà inoltre anche di integrare tutti gli altri aspetti dell'apprendimento del linguaggio che

abbiamo analizzato in precedenza e che sono legati, ad esempio, al senso della vista.

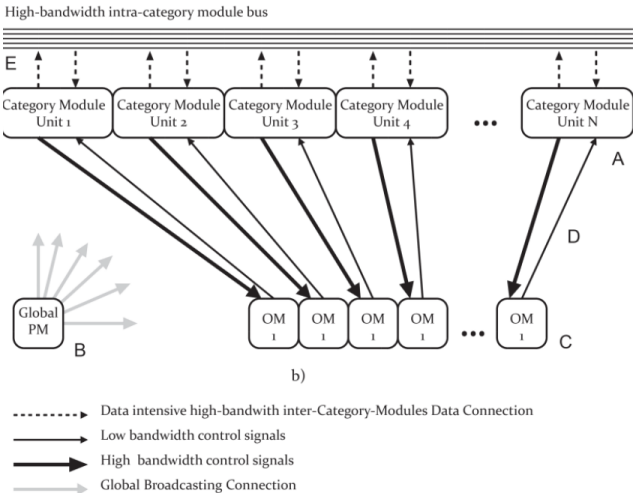


Figura 6.9 - L'immagine riprende i concetti esposti precedentemente e li schematizza in base ai moduli presenti in IDRA: i moduli di categorizzazione rappresentano la corteccia, i moduli Ontogenetici svolgono le funzioni relative al talamo e l'amigdala è modellizzata dal Modulo Filogenetico Globale.

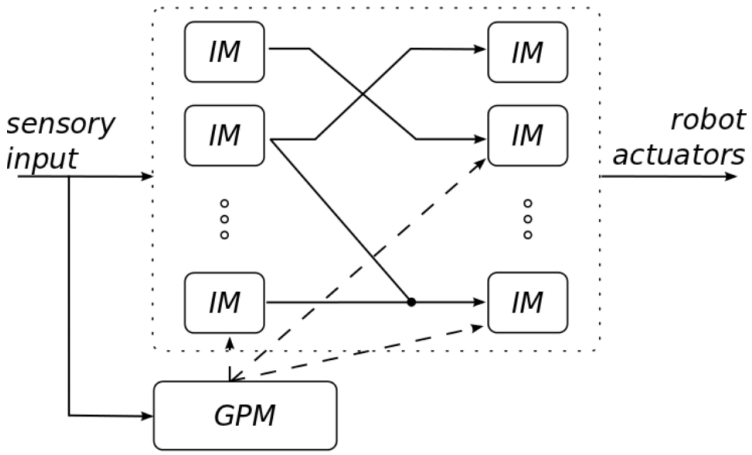


Figura 6.10 - L'immagine mostra il percorso del segnale di input all'interno del sistema IDRA: inizialmente passa al primo livello di Moduli Intenzionali e parallelamente viene inviato al Modulo Filogenetico Globale. Il MFG invia il segnale filogenetico a tutti i livelli di IM, mentre ogni livello di IM passa le informazioni di input codificate e il suo indice di interesse agli altri livelli di IM. Alla fine viene trasmesso un segnale alla parte motoria del robot nella quale viene deciso il movimento migliore da compiere sulla base dei segnali ricevuti dall'architettura.

6.4.2 Funzionamento del Modello delle Primitive Linguistiche utilizzando IDRA

Anche se IDRA è stato utilizzato in passato con input visivi e sfruttando il modello delle primitive motorie, esso può funzionare correttamente anche con il nostro modello, che fa uso di input uditivi e primitive linguistiche. Inoltre, dato che per la tabella stato-suono ci siamo ispirati al funzionamento della tabella stato-azione utilizzata con IDRA nell'ambito del movimento, non sono necessarie molte modifiche per integrare i due sistemi.

Prima di utilizzare IDRA per il babbling è necessaria una fase di inizializzazione in cui viene effettuato il training dell'architettura intenzionale, in particolare dell'algoritmo ICA *(Indipendent Component Analisys*) [60]. Questo algoritmo è alla base del funzionamento di IDRA e permette di separare un segnale multivariato nelle sue sottocomponenti statisticamente indipendenti, che possono essere in seguito ricomposte in modo differente. ICA funziona correttamente con diversi tipi di input (veniva utilizzato negli esperimenti passati con gli input visivi), ma una delle applicazioni più comuni è proprio con segnali audio: a partire da una registrazione contenente diversi suoni sovrapposti (ad esempio una persona che parla, un cane che abbaia, una sirena), l'algoritmo è in grado di separare ciascun suono, restituendo tracce audio separate.

Proprio grazie a questa capacità di ICA di lavorare con input differenti non è necessario scomporre i suoni in feature prima di passarglieli, il che riduce la complessità computazionale ed è un processo più corretto da un punto di vista biologico: la capacità di discriminare i suoni è innata nei bambini e si sviluppa fisiologicamente insieme al progresso cognitivo.

Dopo il training di ICA, IDRA può iniziare a funzionare: a questo punto è necessario creare la tabella stato-suono. Per farlo vengono passati ad IDRA molti suoni casuali, i quali vengono elaborati tramite l'architettura intenzionale (ancora non viene sfruttato né il modulo filogenetico globale, né il segnale rilevante) in cui il modulo di categorizzazione genera le categorie e restituisce un vettore che codifica l'input. I dati così ottenuti vengono collezionati ed utilizzati successivamente per ottenere gli stati iniziali, lanciando l'algoritmo K-Means, esattamente come avveniva nel LPM. Gli stati così ottenuti andranno a comporre la tabella stato-suono, inizialmente vuota.

A questo punto la fase di training è conclusa e si può iniziare quella di testing, sfruttando tutte le componenti di IDRA, compreso il modulo filogenetico globale. Questo modulo, come detto in precedenza, rappresenta l'amigdala e contiene gli istinti innati che determinano l'indice di gradimento per un determinato input (il segnale filogenetico). Nel caso di un bambino che pronuncia una parola tentando di imitarne un'altra ascoltata nell'ambiente circostante, il segnale di gradimento indica il grado di somiglianza tra le due parole.

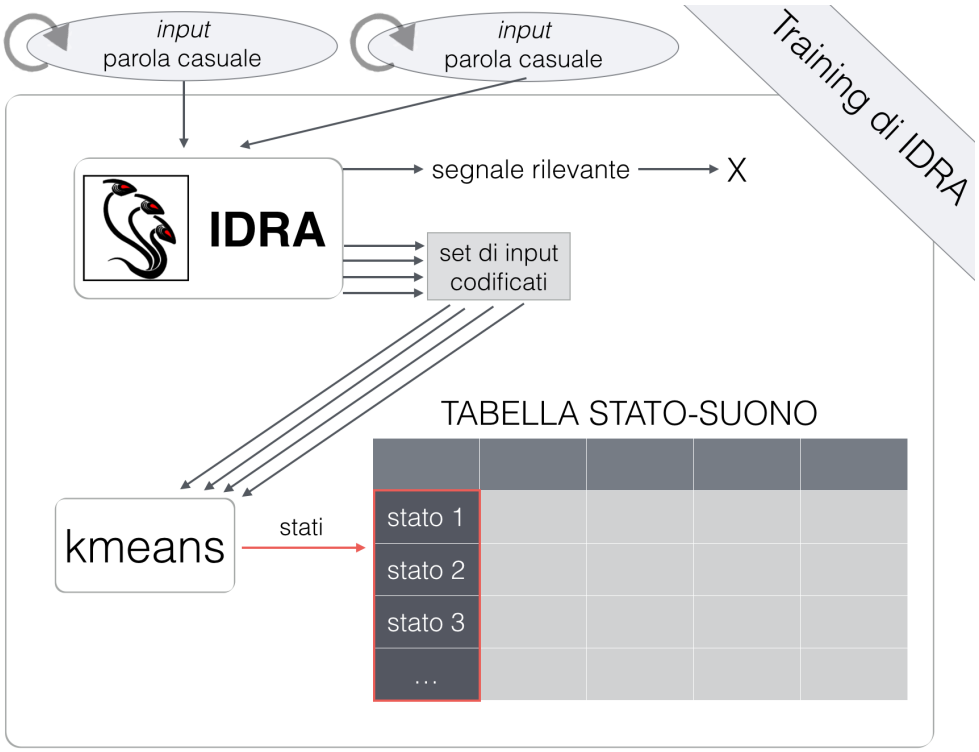


Figura 6.11 - Training di IDRA in cui vengono passati in input suoni casuali così da raccogliere i dati e passarli all'algoritmo kmeans per l'estrazione degli stati iniziali.

D'altra parte abbiamo visto nei capitoli precedenti che nel momento in cui il bambino pronuncia effettivamente una parola simile a quella ascoltata, avviene un'attivazione neurale registrabile nel cervello, processo assolutamente innato e dunque perfetto per essere simulato da questo modulo [11].

Durante la fase di babbling, in cui viene svolto il test, vengono passati ad IDRA due suoni ad ogni tentativo: il target da imitare e un segnale propriocettivo, ossia il suono prodotto dal sistema nella fase d'imitazione precedente. Questo corrisponde al momento in cui un bambino che cerca di imitare una parola la pronuncia e contemporaneamente si ascolta, valutando il grado di somiglianza e decidendo se fermarsi soddisfatto o tentare di migliorare la pronuncia ripetendo nuovamente la parola. Poiché ad ogni ciclo è necessario passare in input a IDRA oltre che il target, anche il suono generato dal sistema nel ciclo precedente, per superare il dislivello temporale utilizziamo un suono casuale per avviare il primo tentativo.

I due suoni passati in input vengono prima di tutto duplicati: una coppia viene concatenata in un'unico vettore e passata all'architettura intenzionale. Questo viene proiettato sulle componenti indipendenti di ICA e poi sottoposto al modulo di categorizzazione (la corteccia) dove viene calcolata l'attivazione di tutte le categorie, ossia la distanza tra il segnale e le categorie già create. Se

questa distanza, insieme al segnale filogenetico, superano una certa soglia impostata a priori, viene creata una nuova categoria. Alla fine viene restituito un vettore che contiene il segnale iniziale codificato. L'altra coppia in input viene invece prima filtrata (fase in cui vengono estratte le feature) e poi viene calcolata la similarità tra i due segnali. La similarità sarà inviata al modulo filogenetico globale e costituirà la metrica per determinare il grado di gradimento. Questo segnale sarà infine inviato all'architettura intenzionale che estrarrà il segnale rilevante, che rappresenta quanto lo stato attuale è gradito al sistema.

Dunque alla fine IDRA restituirà due output: il primo è una codifica dell'input iniziale e viene utilizzato per calcolare lo stato corrente del sistema; il secondo output è costituito invece dal segnale rilevante che sarà inserito nella tabella stato-suono, così come nel sistema senza IDRA veniva registrata la similarità. Se il segnale rilevante di un suono prodotto supera la threshold allora il suono sarà accettato come imitazione valida della parola target, altrimenti si procederà in un altro tentativo.

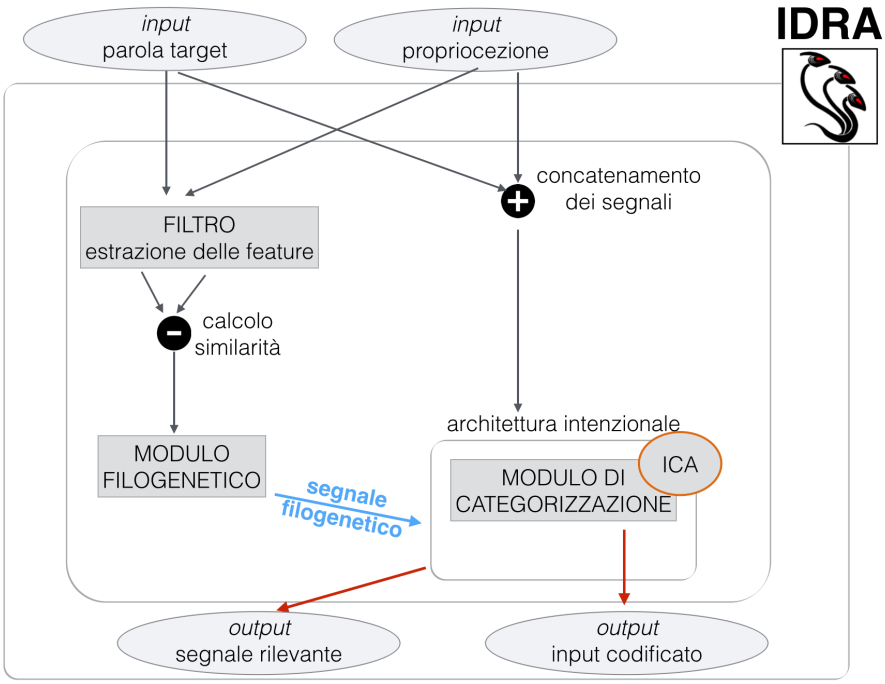


Figura 6.12 - Architettura interna di IDRA. Il sistema riceve in input il suono target e quello propriocettivo, li passa al modulo filogenetico e all'architettura intenzionale restituendo in output un segnale che rappresenta la codifica dei due suoni iniziali e il segnale rilevante che determina il grado di interesse del sistema per l'input.

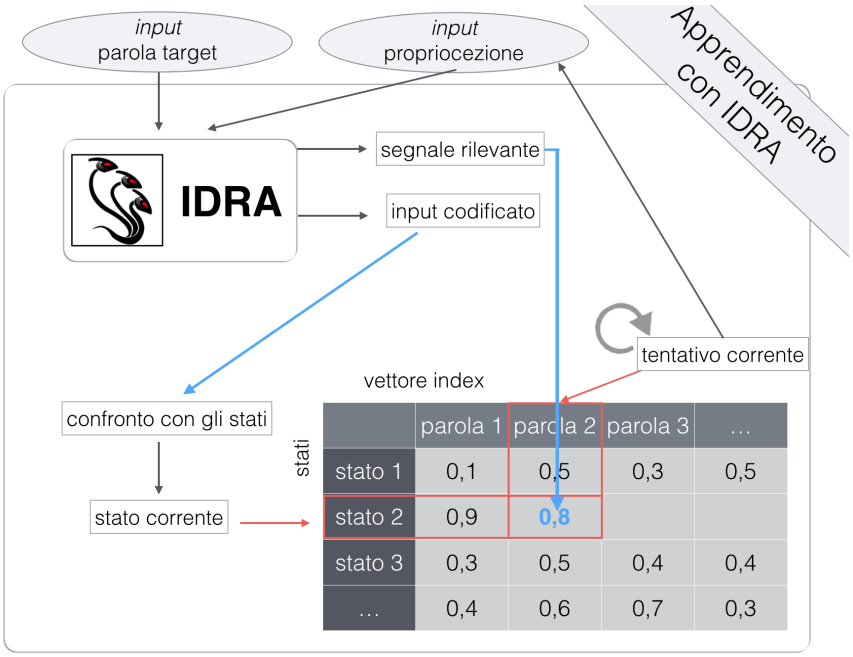


Figura 6.13 - Processo di apprendimento linguistico utilizzando IDRA. L'input codificato determina lo stato corrente e il segnale rilevante viene registrato nella tabella stato-suono.

Capitolo 7

Risultati Sperimentali

"Il mondo, in sé, non è ragionevole: è tutto ciò che si può dire."

Albert Camus da "Il Mito di Sisifo"

Per valutare la validità e la correttezza del nostro modello abbiamo svolto diversi tipi di esperimenti. Data la novità del sistema realizzato e l'assenza in letteratura di tentativi assimilabili al nostro, non abbiamo potuto fare un confronto oggettivo con altre architetture simili.

Questo tuttavia non incide sulla veridicità dei risultati ottenuti dato che lo scopo del nostro lavoro è quello di proporre un modello nuovo di apprendimento del linguaggio che sia bioispirato ed integrabile con sistemi più complessi e che sfrutti, allo stesso tempo, meccanismi già utilizzati in passato per il movimento.

Dunque l'obiettivo che abbiamo perseguito nella fase di sperimentazione è stato quello di validare il nostro modello, dimostrando che sia effettivamente funzionante e che i risultati ottenuti siano significativi anche da un punto di vista biologico.

7.1 Motivazioni per gli esperimenti e metriche utilizzate

Per dimostrare che il LPM funziona, abbiamo dovuto impostare delle metriche oggettive che ci permettessero di quantificare le prestazioni del sistema e confrontarle con quelle biologiche.

Trattandosi di un compito di imitazione, la funzionalità base del sistema è quella di imitare correttamente una parola inviata in input, utilizzando le primitive base a sua disposizione, solo dopo aver svolto la fase iniziale di training del sottosistema involontario. Allo stesso tempo, ripetendo il processo di imitazione per più parole, il sistema deve essere in grado effettivamente di apprendere: dato un gruppo limitato di parole e di stati, il sistema deve riuscire ad imitare il suono in input in un tempo sempre minore.

Gli esperimenti saranno dunque tesi a valutare da una parte l'abilità del sistema di imitare correttamente una parola fornita in ingresso, dall'altra la capacità generale di apprendimento nel tempo. Di volta in volta saranno variati alcuni parametri del sistema e sarà fornito il significato di quello che si sta facendo da un punto di vista biologico.

Poiché basarci unicamente su parametri qualitativi per discriminare quanto effettivamente il suono restituito fosse simile non ci avrebbe permesso di ottenere dei dati oggettivi, abbiamo deciso di utilizzare come metrica di base la similarità, ossia il grado di somiglianza tra un suono e il rispettivo target da imitare.

Dati i due suoni da paragonare, come primo passo ne abbiamo calcolato la distanza vettoriale, utilizzando una distanza euclidea sulle 34 dimensioni dei vettori di feature:

```
vectorDistance = norm(vect1(1,:) - vect2(1,:), 2).^2
```
(7.1)

dove *norm(vector,type)* è una funzione implementata nella libreria Matlab che calcola la norma euclidea di un vettore, in questo caso del vettore differenza tra i due suoni rappresentati da *vect1* e *vect2*.

Le distanze così calcolate tuttavia non rappresentano un parametro sufficientemente esplicativo per definire la somiglianza tra i due suoni; questo costituisce un primo problema dato che non sono comprese in un intervallo fisso ma variano molto in base a diversi fattori che riguardano l'algoritmo: ad esempio se si aumenta il numero di stati la distanza media tra i vettori dello stesso stato si riduce, mentre avviene il contrario se si utilizza un dataset più ampio.

Si è scelto quindi di estrarre la *similarità* tra i due vettori, data la loro distanza, con la seguente formula:

```
similarity = tanh(1 ./ vectorsDistance);
```
(7.2)

La *similarità* costituisce quindi la probabilità che i due suoni rappresentino la stessa parola e va da zero (parole completamente differenti) ad uno (parole identiche).

Un'altra metrica che abbiamo preso in considerazione nei nostri esperimenti è il numero di cicli impiegato dal sistema per restituire un risultato valido. Questo dato è molto rilevante anche da un punto di vista biologico: ciascun ciclo del sistema rappresenta un diverso tentativo del bambino di imitare la parola ascoltata; se l'imitazione va a buon fine, ossia se il suono prodotto è abbastanza simile il sistema si ferma, altrimenti fa un nuovo tentativo utilizzando un diverso suono e valuta nuovamente la similarità, continuando finché non raggiunge un risultato che supera il grado di somiglianza voluto. Il numero di cicli necessario non solo permette di valutare la velocità del sistema nel risolvere un dato compito, ma permette di fare dei confronti oggettivi tra un esperimento e l'altro.

Inoltre, ripetendo il processo di imitazione per molte parole, il numero di cicli necessari medio fornisce una metrica per valutare il processo di apprendimento generale, ossia quanto il sistema stia effettivamente imparando e se le prestazioni migliorano nel tempo, così come avviene in natura.

7.2 Inizializzazione del Sottosistema Involontario e training del sistema

Il dataset di primitive di base è stato estratto da 20 tracce audio contenenti la registrazione di diversi bambini durante il babbling. I file sono stati segmentati in modo semi-manuale utilizzando Praat tramite il procedimento visto in precedenza, così da ottenere circa 100 singoli suoni segmentati. Da questo insieme sono stati selezionati i 30 suoni migliori dal punto di vista della qualità audio e che componessero un dataset omogeneo ed esaustivo, contenente tutte le combinazioni vocale-consonante comuni tra i bambini. I suoni hanno una durata totale simile, anche se non precisamente la stessa, dato che dipende dal tipo di vocale e consonante; questo non crea alcun tipo di problema in quanto il sistema è in grado di lavorare correttamente anche con suoni di durate diverse.

I 30 suoni ottenuti, che rappresentano dunque il dataset iniziale di primitive linguistiche; queste sono state immesse nel sistema come input e sono state normalizzate ed elaborate con il risultato di ottenere un dataset esteso di 900 suoni composti tutti differenti ma qualitativamente omogenei tra di loro, da utilizzare nelle fasi successive.

Questo dataset esteso è stato passato come input al modulo di estrazione delle feature, il quale è in grado di estrarre un numero differente di caratteristiche, in base ai metodi e alle statistiche selezionati.

Nella fase di elaborazione *short-term* abbiamo scelto di utilizzare le seguenti feature:

Dominio del tempo

- *Energy:* descrive i punti di variazione dell'intensità all'interno del segnale audio. Abbiamo scelto di utilizzarla perché i segnali sono piuttosto irregolari dal punto di vista dell'intensità dato che rappresentano combinazioni vocale-consonante e non parole ben formate.
- *Zero Crossing:* descrive i punti in cui l'onda attraversa lo zero. E' molto utilizzato in caso di segnali rumorosi, in particolare per l'analisi delle parole.

Dominio della frequenza

- *Spectral Centroid:* descrive il centro di gravità dello spettro e la deviazione delle frequenze dal centro. E' molto utilizzato per l'analisi del parlato.

- *Cepstrum:* Permette di estrarre 13 differenti feature, è un metodo simile a quello utilizzato per l'identificazione del pitch e descrive molto bene parole diverse anche in caso di rumore.

Nella fase di *mid-term* abbiamo invece utilizzato 2 statistiche: la media e la deviazione standard. Quindi ogni segnale audio del dataset è descritto da 17 feature di *short-term* dalle quali sono ricavate 2 statistiche di *mid-term*, per un totale di 34 caratteristiche, che sono le dimensioni dello spazio vettoriale che andremo a considerare nelle fasi successive.

Il dataset esteso (sotto forma di vettori di feature) è stato poi inviato nel modulo di clustering dove è avvenuto il training del sistema; sono stati estratti gli stati iniziali limitandone il numero a 20, seguendo un'euristica molto utilizzata per determinare un buon numero di partizioni per un dato insieme di elementi:

$$k \approx \sqrt{\frac{n}{2}} \tag{7.3}$$

dove k è il numero di partizioni e n il numero di elementi [55].

Il clustering, eseguito in questo spazio di 34 dimensioni, ha separato i 900 elementi del dataset esteso in 20 partizioni, restituendo il vettore di ognuno dei 20 centroidi che andranno a comporre gli stati iniziali, utilizzati per inizializzare la tabella stato-suono, che è l'output del sottosistema involontario.

Per testare il corretto funzionamento del modulo di clustering abbiamo inizializzato il sistema utilizzando soltanto due feature per suono e lavorando quindi in uno spazio di due sole dimensioni facilmente visualizzabile in un grafico cartesiano.

I risultati ottenuti dimostrano che il clustering suddivide correttamente i dati, anche se in realtà questi non sono distribuiti in modo perfettamente omogeneo nello spazio: questo non dipende da una scarsa qualità del dataset iniziale, ma al fatto che la maggior parte dei suoni prodotti dal bambino sono molto simili tra di loro per motivi fisiologici e ci sono vocali e consonanti che vengono ripetute in modo molto più frequente rispetto ad altre data la loro maggiore semplicità di pronuncia.

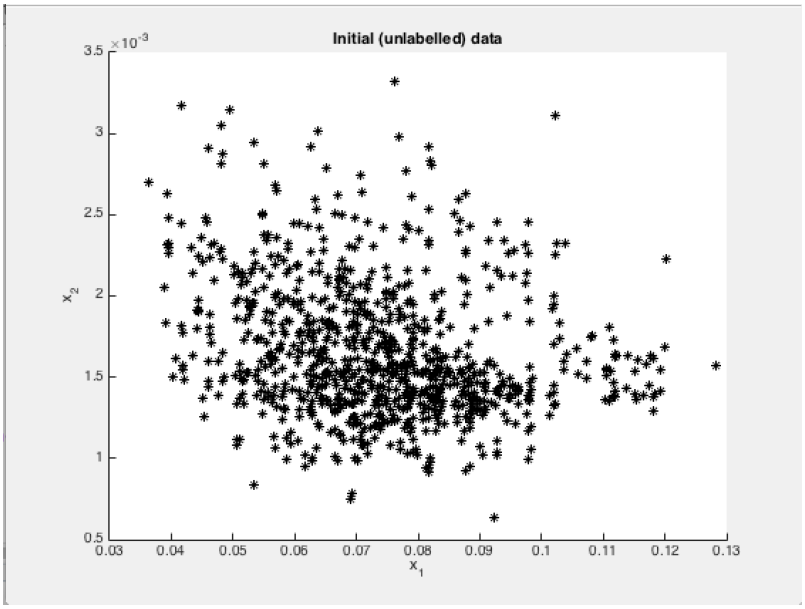


Figura 7.1 - Nel piano cartesiano sono mostrati i 900 suoni che compongono il dataset esteso individuati dalle 2 feature caratteristiche estratte per il test 2D. Questo insieme di elementi è l'input passato al modulo di clustering che dovrà partizionarlo.

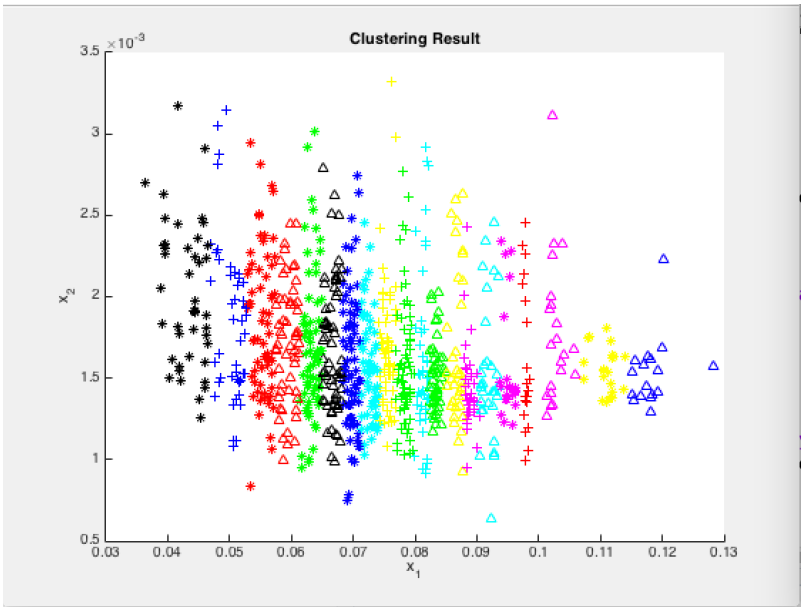


Figura 7.2 - Nel piano cartesiano sono mostrati i 900 suoni che compongono il dataset esteso partizionati in 20 gruppi dall'algoritmo di clustering, ciascuno dei quali rappresenta un diverso stato iniziale del sistema.

7.3 Valutazione sperimentali delle variabili dell'architettura

Una prima importante variabile del sistema è il livello minimo di similarità richiesto: il sistema può essere infatti impostato con threshold differenti che variano tra zero ed uno e che rappresentano il grado di somiglianza minimo necessario per poter accettare il suono prodotto. In pratica dati due suoni differenti, come ad esempio il suono target da imitare e il suono prodotto come imitazione, viene calcolato il grado di similarità: se questo supera la threshold impostata viene accettato come valido e il sistema non esegue altri tentativi. Se invece la similarità con il suono target è inferiore della threshold richiesta il sistema dovrà compiere altri tentativi, fino a raggiungerla.

Dal punto di vista biologico la threshold è strettamente connessa all'età del bambino: con lo sviluppo del sistema uditivo e delle aree celebrali migliora l'abilità di discriminare due suoni simili, dunque l'infante sarà in grado di rendersi meglio conto se il suono prodotto sia o meno simile a quello ascoltato nell'ambiente circostante. Allo stesso tempo anche i genitori, con il trascorrere dei mesi, cercheranno di spronare il figlio all'apprendimento esigendo parole sempre più corrette, invitandolo a ripetere più volte quei termini la cui pronuncia non è adeguata, così da raggiungere un grado di somiglianza maggiore.

Nel nostro sistema questo valore è un parametro molto importante perché è collegato sia al grado di similarità medio ottenuto, che al numero di tentativi necessari al sistema per apprendere e imitare una data parola. Dunque è necessario scegliere questa soglia minima considerando il trade-off tra grado di similarità desiderato e tempo impiegato dal sistema, cercando di ottenere buone prestazioni in un tempo limitato.

Per compiere il processo di valutazione e i successivi esperimenti abbiamo suddiviso il dataset esteso di 900 suoni in due parti: un dataset di 600 suoni che è utilizzato unicamente nella fase preliminare per il training del sistema e la generazione degli stati, e un dataset di 300 suoni utilizzato unicamente nella fase di testing. Questa suddivisione del dataset in due parti assicura che le parole passate al sistema durante gli esperimenti siano differenti da quelle utilizzate nella fase di training, garantendo risultati attendibili.

Per effettuare questa valutazione abbiamo implementato uno script che estrae in modo casuale 200 parole diverse dal dataset di testing e le passa al modulo di apprendimento in altrettanti cicli, valutando il risultato dell'apprendimento. Questo processo sarà ripetuto per 19 volte, testando ogni volta una threshold più alta di 0.05, andando quindi da un minimo di 0.05 ad un massimo di 0.95. Ad ogni ciclo sarà innalzata la threshold, si estrarranno 200 parole differenti scelte in modo casuale e sarà inizializzata nuovamente la tabella stato-suono e il vettore di input; questo è necessario perché variando la soglia di accettazione i valori con i quali è stata precedentemente riempita la tabella non sono più validi.

7.3.1 Risultati della valutazione

Per ogni threshold testata è stato salvato il numero di cicli medio impiegato per imitare le parole in input: come si può vedere dalla tabella e dal grafico bastano pochi tentativi per imitare una parola target fino ad una soglia di 0.8. Il numero di cicli aumenta molto per valori maggiori, per cui comunque il grado di somiglianza è molto elevato.

Ascoltando alcune delle parole imitate e confrontandole con quelle target fornite in input, si nota che già due parole con un grado di similarità di 8.5 sono difficili da distinguere; non serve quindi giungere ad un numero di cicli molto elevato per avere ottimi risultati.

THRESHOLD	CICLI MEDI
0,05	1,0000
0,10	1,0000
0,15	1,0000
0,20	1,0000
0,25	1,0000
0,30	1,0000
0,35	1,0653
0,40	1,0653
0,45	1,0452
0,50	1,4422
0,55	1,6482
0,60	2,0402
0,65	2,5578
0,70	2,2513
0,75	3,4523
0,80	6,9347
0,85	12,3367
0,90	32,2412
0,95	64,9799

Tabella 7.1 - Nella tabella sono indicati, per ogni diversa threshold, il numero di tentativi medi necessari a completare il processo di imitazione. Sono evidenziati i valori giudicati accettabili dal punto di vista delle prestazioni.

Figura 7.3 - Grafico che mostra l'andamento del numero medio di tentativi necessari al sistema per imitare una parola in funzione della soglia minima impostata nell'algoritmo

Un dato ancora più interessante è fornito dall'analisi del grado di similarità media tra le parole imitate e quella target, in base alla threshold selezionata.

Come si evince dai dati la threshold minima impostata nell'algoritmo non corrisponde esattamente al grado di similarità media delle parole trovate, che resta più sempre alto. Questo implica che non è necessario impostare una soglia molto alta per garantire buoni risultati, ossia non c'è bisogno di molti tentativi perché le parole trovate nel processo di imitazione siano simili alla parola target. Impostare una soglia troppo alta riduce anzi l'efficenza dell'algoritmo, perché la differenza tra similarità effettiva e similarità richiesta (delta) si riduce aumentando la seconda.

Dunque buoni valori di threshold sono quelli compresi tra 0.65 e 0.8, per i quali si ha una similarità dallo 0.8 allo 0.9, che garantiscono un'imitazione molto fedele della parola target in soli 10 tentativi medi. Questo primo esperimento già mostra che il sistema è veloce ad apprendere e non necessita di molti tentativi per raggiungere buoni risultati, né di molti stati iniziali. E' una situazione simile a quella di cui abbiamo discusso in precedenza, quando abbiamo parlato del gap tra input forniti ai bambini e capacità di apprendere: anche con pochi stimoli dall'esterno i bambini dimostravano una capacità di apprendimento molto elevata.

THRESHOLD	SIMILARITA' MEDIA	DELTA (threshold-similarità media)
0,05	0,5712	0,5212
0,10	0,5739	0,4739
0,15	0,6452	0,4952
0,20	0,6796	0,4796
0,25	0,6994	0,4494
0,30	0,6009	0,3009
0,35	0,8046	0,4546
0,40	0,5101	0,1101
0,45	0,7591	0,3091
0,50	0,6776	0,1776
0,55	0,7917	0,2417
0,60	0,7308	0,1308
0,65	0,8271	0,1771
0,70	0,8588	0,1588
0,75	0,8523	0,1023
0,80	0,8947	0,0947
0,85	0,9044	0,0544
0,90	0,9568	0,0568
0,95	0,9867	0,0367

Tabella 7.2 - Nella tabella sono indicati, per ogni diversa threshold, il livello di similarità effettivamente ottenuto. Inoltre è riportato il delta tra similarità richiesta e similarità ottenuta. Sono infine evidenziati i valori giudicati accettabili dal punto di vista della similarità tra parola target e parola imitata restituita.

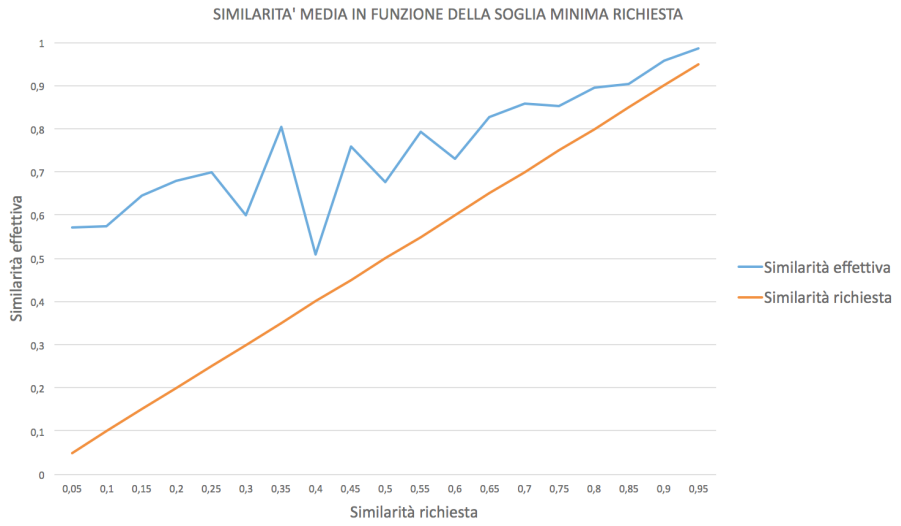


Figura 7.4 - Grafico che mostra l'andamento della similarità media tra parola target e risultati ottenuti in funzione della soglia minima di similarità impostata nell'algoritmo. La linea rossa mostra come sarebbe l'andamento della similarità ottenuta se coincidesse con quella richiesta.

Figura 7.5 - Nel primo grafico in alto a sinistra, che mostra l'andamento del numero di tentativi medio in funzione della threshold richiesta, è stata evidenziata la parte giudicata accettabile dal punto di vista del numero dei cicli e del tempo impiegato dal sistema per giungere ad una soluzione, ossia fino a circa 10 tentativi per parola. Nel grafico in alto a destra, rappresentante il grado di similarità media in funzione della threshold, è stata evidenziata invece la sezione corrispondente ad una similarità che garantisce imitazioni molto

fedeli alla parola target. L'ultimo grafico è composto dalla sovrapposizione dei precedenti e mostra l'intervallo ottimale per mantenere buone prestazioni per quanto riguarda il numero di cicli, pur ottenendo una similarità molto buona.

7.4 Primo esperimento: apprendimento tramite un numero ridotto di parole in input

Il primo esperimento è svolto con lo scopo di ottenere una valutazione oggettiva dell'importanza che riveste il numero di parole in input fornite al sistema per ogni test.

Quando i familiari si rivolgono al bambino tendono ad utilizzare un lessico semplificato, con forme verbali non coniugate, cercando di evitare termini troppo articolati o di difficile pronuncia e scandendo bene ogni sillaba. Il bambino tenderà quindi a memorizzare e ad imparare inizialmente quelle parole che vengono pronunciate più volte in sua presenza o a cui viene assegnata un'importanza maggiore data dal tono, dal contesto e dal numero di volte che quel termine viene ripetuto.

Abbiamo svolto il nostro test preliminare scegliendo in modo casuale 200 parole prese dal dataset di testing composto a sua volta da 300 termini. Per simulare il linguaggio semplificato da parte dei familiari abbiamo ridotto il dataset di testing a sole 20 parole, dalle quali vengono selezionate, sempre in modo casuale, le 200 parole del test. Dunque non solo i termini immessi in input al sistema sono molto simili, ma sono spesso ripetuti.

Il primo esperimento è stato dunque condotto con gli stessi parametri di default indicati in precedenza, ma variando il dataset di parole di testing:

- dataset di training: 600 parole
- **dataset di testing: 20 parole**
- test effettuati: 19
- variazione della threshold per ogni test: 0.05 (da 0.05 a 0.95)
- numero parole in input ad ogni test: 200 parole

7.4.1 Risultati del primo esperimento e confronto con i dati biologici

I risultati ricavati dal primo esperimento, ossia l'andamento del numero dei tentativi e della similarità ottenuta in funzione della threshold, sono stati confrontati con quelli ottenuti nella fase preliminare, così da poter valutare la variazione nelle prestazioni del sistema.

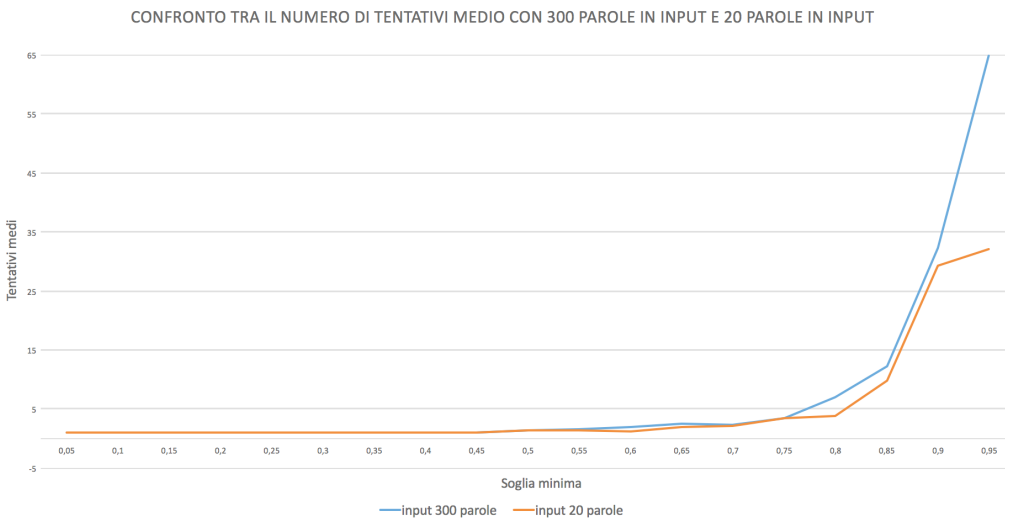


Figura 7.6 - Grafico che mostra l'andamento del numero medio di tentativi per apprendere una parola in funzione della soglia minima di similarità impostata nell'algoritmo. I dati (in blu) preliminari, ottenuti utilizzando un dataset di testing di 300 parole, sono confrontati con quelli (in rosso) ottenuti utilizzando un dataset di testing di sole 20 parole. In quest'ultimo caso le prestazioni sono migliori, in particolare per threshold molto alte.

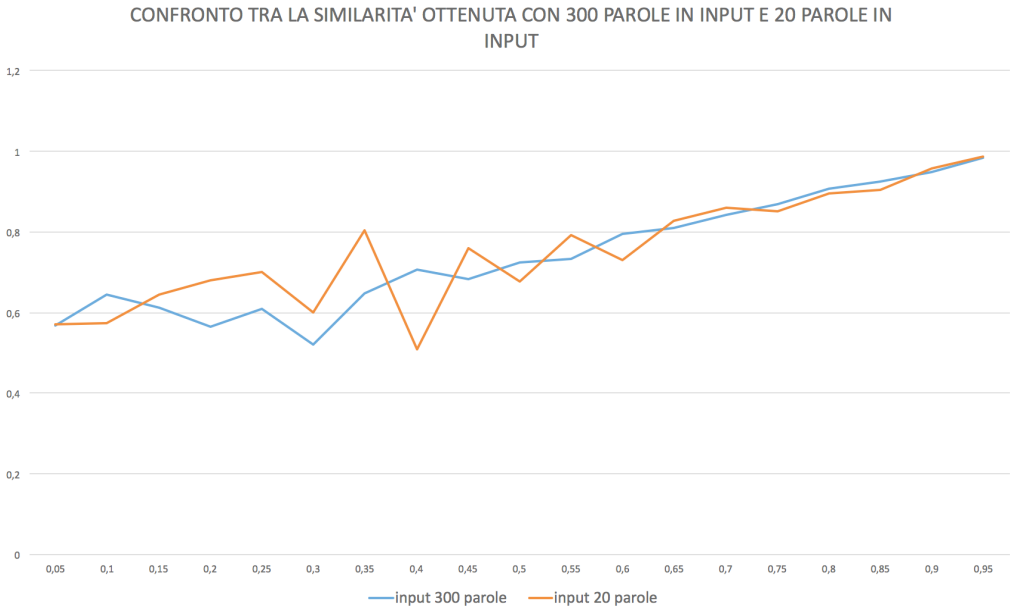


Figura 7.7 - Grafico che mostra l'andamento della similarità media ottenuta in funzione della soglia minima di similarità impostata nell'algoritmo. I dati (in blu) preliminari, ottenuti utilizzando un dataset di testing di 300 parole, sono confrontati con quelli (in rosso) ottenuti utilizzando un dataset di testing di sole 20 parole. Le prestazioni dei due diversi casi coincidono, in particolare per valori alti, dunque l'utilizzo di meno tentativi nel processo di apprendimento non conduce a risultati peggiori, se si riduce il numero delle parole in input.

Dal confronto emerge che nel caso in cui sono utilizzate meno parole il sistema riesce a completare il processo di imitazione in un numero di cicli minore rispetto alla fase preliminare, soprattutto nel caso di threshold molto elevate. Quindi il sistema ha bisogno di meno tentativi per poter raggiungere risultati molto buoni. Inoltre, pur diminuendo il numero di tentativi medio, la similarità ottenuta non si discosta molto da quella della fase preliminare, dunque la qualità media delle imitazioni trovate non peggiora, soprattutto per l'intervallo di nostro interesse che va da una threshold di 0.65 a 0.8, in cui i risultati dell'esperimento coincidono con quelli precedenti.

Da un punto di vista biologico questa accelerazione nella velocità di apprendimento coincide con quanto osservato nei bambini e rilevato tramite i dati sperimentali dello *Speechrome Project*, che mostrano un picco di apprendimento, per una determinata parola, esattamente nel momento in cui viene pronunciata con più frequenza dai familiari e in modo più omogeneo. In concomitanza con i momenti di maggior utilizzo avviene infatti l'apprendimento della stessa da parte del bambino.

La vicinanza tra i dati ottenuti utilizzando il LPM e quelli ricavati da osservazioni sperimentali su bambini veri, conferma la correttezza del modello utilizzato e lascia ben sperare per possibili applicazioni future nel campo del linguaggio. Anche il numero di cicli medio necessari per imitare una parola che è stato ricavato è realistico e compatibile con il numero medio di tentativi eseguiti da un bambino per tentare di pronunciare correttamente un termine.

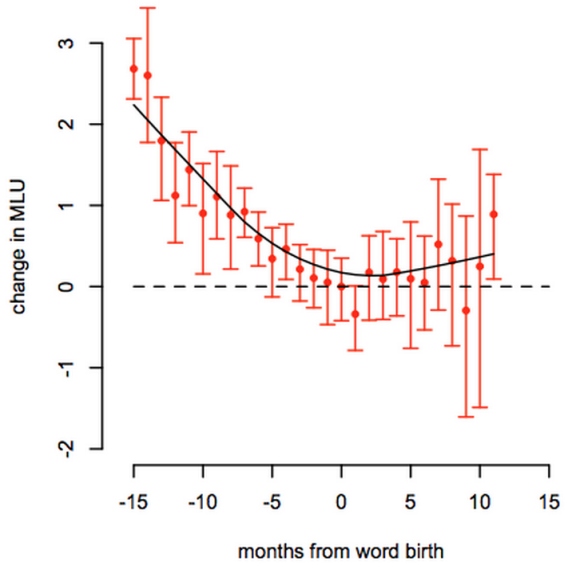


Figura 7.8 - Il grafico, ottenuto basandosi sui dati sperimentali dello Speechrome Project, mostra che l'apprendimento di una parola da parte di un bambino avviene in concomitanza con un momento in cui i familiari pronunciano il termine più volte e in modi molto simili l'uno all'altro. Il risultato conferma la correttezza di quanto ottenuto utilizzando il LPM nel processo di apprendimento.

7.5 Secondo esperimento: aumento della velocità di apprendimento nel tempo

In questo esperimento abbiamo voluto testare la capacità del sistema di imparare nuove parole e in particolare abbiamo analizzato la variazione della velocità di apprendimento in funzione del numero di parole passate in input.

Subito dopo l'inizializzazione del sistema la tabella stato-suono creata dal sottosistema involontario è vuota. Dunque per la prima parola in input il sistema, dopo aver calcolato lo stato corrente, esegue diversi tentativi di imitazione producendo suoni casuali, memorizzando i parametri di similarità per ciascun suono e fermandosi soltanto nel momento in cui questi siano maggiori della threshold richiesta.

Susseguendosi numerosi tentativi, la tabella viene riempita e per ogni stato vengono memorizzati i valori di similarità relativi a tutti i suoni prodotti in precedenza. Ad ogni iterazione di apprendimento il sistema valuterà, prima di tutto per lo stato corrente, il suono già prodotto in passato con similarità maggiore poi, se questo non dovesse risultare abbastanza buono, tenterà gli altri suoni già testati, prima di generarne nuovi in modo casuale. Dunque procedendo con i tentativi la probabilità di trovare, al primo ciclo, una parola abbastanza simile a quella target, aumenterà nel tempo a mano a mano che si provano ad imitare diverse parole simili.

Questo esperimento è stato condotto con gli stessi parametri utilizzati nella fase preliminare, utilizzando però una threshold fissa di 0.8, un valore che garantisce un'ottima similarità dei risultati e allo stesso tempo un numero di cicli medi piuttosto basso:

- dataset di training: 600 parole
- dataset di testing: 300 parole
- test effettuati: 1
- threshold per il test: 0.8
- numero parole in input: 200 parole

7.5.1 Risultati del secondo esperimento e confronto con i dati biologici

Dall'esperimento svolto abbiamo ricavato il numero di cicli necessario, per ogni parola target inviata come input al sistema, per imitarla in modo adeguato. Da questi valori abbiamo poi estratto la media mobile, ossia una media fatta sui valori precedenti, col fine di prevedere il valore del prossimo dato. Per farla abbiamo utilizzato una finestra di 30 dati: i primi dati sono utilizzati per eseguire la prima media e non sono considerati nei risultati, mentre per i dati successivi viene effettuata una stima a partire dai precedenti, così da avere il valore stimato secondo quelli passati. Utilizzando la media

mobile siamo riusciti ad ottenere un grafico dell'andamento medio del numero di cicli necessari.

Dai risultati si evince come il numero di cicli richiesti scende susseguendosi le parole in input, mentre aumenta la frequenza delle parole per cui è necessario un solo ciclo per essere imitate a dovere, che corrispondono in pratica a parole già apprese o molto simili a quelle imparate in precedenza. Anche il numero medio di tentativi, nel caso in cui la prima parola tentata non sia abbastanza simile a quella target, scende proseguendo nel processo di apprendimento, dato che la tabella stato-suono conterrà sempre più valori.

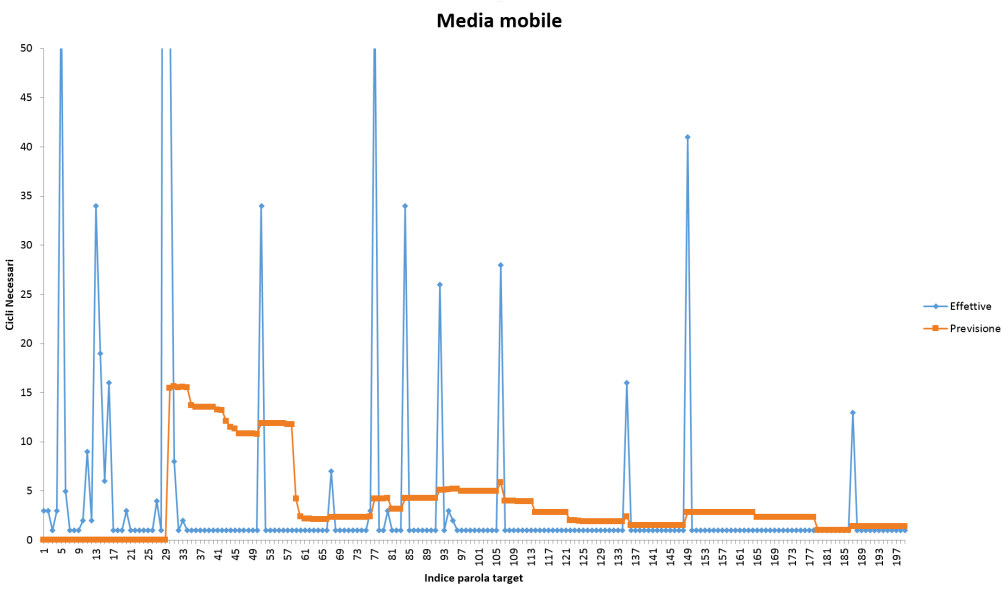


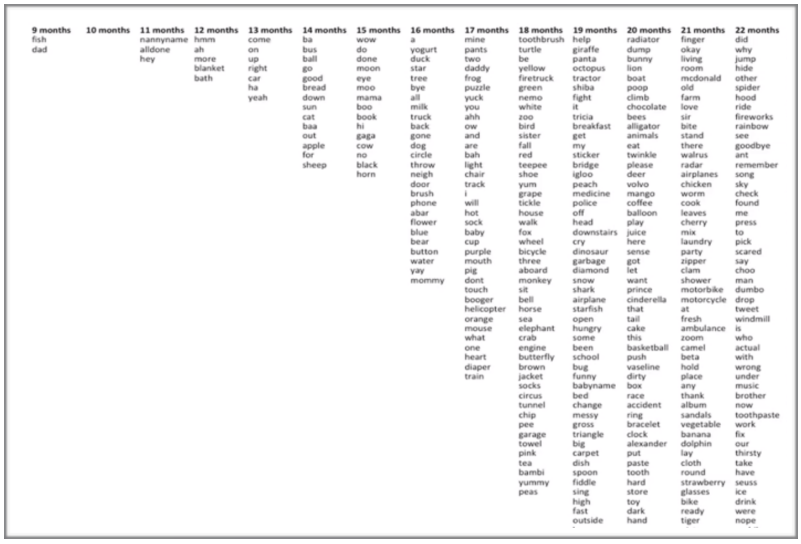
Figura 7.9 - Il grafico mostra, per ognuna delle 200 parole passate in input al sistema, il numero di tentativi necessario per ottenere un'imitazione abbastanza fedele del target. Sul grafico è mostrata anche la previsione dei cicli necessari per ogni parola, ossia la media mobile, che permette di ottenere un andamento dinamico del numero di cicli effettuati. Si può notare come i tentativi necessari decrescono attraverso il tempo, con l'aumentare delle parole apprese.

I dati ottenuti mostrano che il modello migliora le sue prestazioni nel tempo a mano a mano che la tabella stato-suono viene riempita, aumentando il numero di suoni appresi i cui valori sono salvati in memoria.

E' noto da risultati scientifici che un bambino, allo stesso modo, memorizza i tentativi effettuati e prima di esplorarne di diversi prova nuovamente quelli già appresi che gli sembrano più simili alla parola che vuole imitare. Durante i mesi, a mano a mano che nuove parole vengono apprese, il bambino diventerà più veloce nell'apprendimento: i dati sperimentali dello Speechrome Project mostrano chiaramente che con il procedere dei mesi la velocità di apprendimento del bambino verso nuove parole aumenta, insieme al numero delle parole apprese.

Anche in questo caso i dati ottenuti vanno di pari passo con quelli sperimentali svolti su un bambino. Naturalmente nel caso biologico sono molti altri i fattori che portano questo risultato, come ad esempio lo sviluppo delle abilità cognitive, che però non sono oggetto di trattazione in questa tesi.

Per muoversi in questa direzione sarebbe necessario implementare una tabella dinamica degli stati, in grado di aggiungere nuovi stati e quindi nuove categorie a mano a mano che vengono svolti nuovi tentativi di imitazione e le parole da imitare sono molto lontane da quelle apprese fino ad ora. IDRA, con il quale abbiamo in seguito integrato il nostro sistema, prevede una simile abilità: il modulo di categorizzazione crea infatti nuove categorie in modo dinamico.



9 months	10 months	11 months	12 months	13 months	14 months	15 months	16 months	17 months	18 months	19 months	20 months	21 months	22 months
fish		nannyname	hmm	come	ba	wow	a	mine	toothbrush	help	radiator	finger	did
dad		alldone	ah	on	bus	do	yogurt	pants	turtle	giraffe	dump	okay	why
		hey	more	up	ball	done	duck	two	be	panta	bunny	living	jump
			blanket	right	go	moon	star	daddy	yellow	octopus	lion	room	hide
			bath	car	good	eye	tree	frog	firetruck	tractor	boat	mcdonald	other
				ha	bread	moo	bye	puzzle	green	shiba	poop	old	spider
				yeah	down	mama	all	yuck	nemo	fight	climb	farm	hood
					sun	boo	milk	you	white	it	chocolate	love	ride
					cat	book	truck	ahh	zoo	tricia	bees	sir	fireworks
					baa	hi	back	ow	bird	breakfast	alligator	bite	rainbow
					out	gaga	gone	and	sister	get	animals	stand	see
					apple	cow	dog	are	fall	my	eat	there	goodbye
					for	no	circle	bah	red	sticker	twinkle	walrus	ant
					sheep	black	throw	light	teepee	bridge	please	radar	remember
						horn	neigh	chair	shoe	igloo	deer	airplanes	song
							door	track	yum	peach	volvo	chicken	sky
							brush	i	grape	medicine	mango	worm	check
							phone	will	tickle	police	coffee	cook	found
							abar	hot	house	off	balloon	leaves	me
							flower	sock	walk	head	play	cherry	press
							blue	baby	fox	downstairs	juice	mix	to
							bear	cup	wheel	cry	here	laundry	pick
							button	purple	bicycle	dinosaur	sense	party	scared
							water	mouth	three	garbage	got	zipper	say
							yay	pig	aboard	diamond	let	clam	choo
							mommy	dont	monkey	snow	want	shower	man
								touch	sit	shark	prince	motorbike	dumbo
								booger	bell	airplane	cinderella	motorcycle	drop
								helicopter	horse	starfish	that	at	tweet
								orange	sea	open	tail	fresh	windmill
								mouse	elephant	hungry	cake	ambulance	is
								what	crab	some	this	zoom	who
								one	engine	been	basketball	camel	actual
								heart	butterfly	school	push	beta	with
								diaper	brown	bug	vaseline	hold	wrong
								train	jacket	funny	dirty	place	under
									socks	babyname	box	any	music
									circus	bed	race	thank	brother
									tunnel	change	accident	album	now
									chip	messy	ring	sandals	toothpaste
									pee	gross	bracelet	vegetable	work
									garage	triangle	clock	banana	fix
									towel	big	alexander	dolphin	our
									pink	carpet	put	lay	thirsty
									tea	dish	paste	cloth	take
									bambi	spoon	tooth	round	have
									yummy	fiddle	hard	strawberry	seuss
									peas	sing	store	glasses	ice
										high	toy	bike	drink
										fast	dark	ready	were
										outside	hand	tiger	nope

Figura 7.10 - Schema ottenuto nello Speechrome Project che mostra le parole nuove apprese dal bambino in ogni mese. Come si può notare la velocità di apprendimento di nuove parole aumenta durante i primi mesi di vita.

7.6 Terzo esperimento: training e testing sullo stesso dataset esteso di parole

Nel terzo esperimento abbiamo voluto testare quanto la scelta del dataset iniziale e la suddivisione dello stesso tra dataset di training e testing sia importante per un buon funzionamento del sistema. Per ottenere dei dati abbiamo deciso di creare un dataset di parole piuttosto esteso (900 parole) da utilizzare sia per la fase di training che per la fase di testing e valutare le prestazioni rispetto al caso standard con il dataset suddiviso in due parti.

Abbiamo quindi utilizzato i seguenti parametri per impostare l'esperimento:

- dataset di training: 900 parole
- dataset di testing: 200 parole (prese dal dataset di training)
- test effettuati: 19
- variazione della threshold per ogni test: 0.05 (da 0.05 a 0.95)
- numero parole in input: 200 parole

7.6.1 Risultati del terzo esperimento

Ci si potrebbe aspettare che, utilizzando lo stesso dataset sia per il training degli stati che per il testing, si ottengano risultati migliori, dato che le parole target che vengono fornite in input sono state già processate dal sistema durante la fase di creazione degli stati.

In realtà i risultati ottenuti mostrano che le prestazioni del sistema, in questo caso, sono peggiori, in particolare per quanto riguarda la velocità di apprendimento: sono necessari più tentativi per imitare correttamente una parola, soprattutto in caso di soglia elevata. Anche la similarità ottenuta risente del dataset comune: per threshold basse si riduce il delta tra similarità richiesta ed ottenuta, dunque le prestazioni del sistema sono inferiori.

Questo risultato in realtà non deve sorprendere, perché è vero che senza dividere il dataset si testano parole già processate dal sistema durante l'inizializzazione, ma allo stesso tempo si è maggiormente sottoposti ai fenomeni di *overfitting*. Tali fenomeni rappresentano casi in cui il modello si adatta ai dati utilizzando troppi parametri e perde la capacità di generalizzare: pur mostrando un comportamento in linea con i risultati previsti, il sistema diventa meno efficiente perché aumenta la probabilità di errore.

Per evitare l'insorgere di questo fenomeno uno dei metodi che viene utilizzato è proprio il partizionamento dei dati tra training set e testing set (detto *cross-validation*), come svolto negli esperimenti precedenti.

Dunque affinché il modello sia in grado di imparare e generalizzare è fondamentale utilizzare dei dati di addestramento differenti da quelli passati in seguito come input durante le fasi successive.

Da un punto di vista biologico questo non è un problema: l'insieme di suoni utilizzati per creare la mappa mentale nei primi mesi di vita sono quelli prodotti dal bambino in modo involontario (ossia le primitive linguistiche di base) e sono in effetti molto differenti da quelli recepiti dal mondo circostante e dalle parole pronunciate dai familiari che sono il corrispettivo biologico dell'input target dell'architettura qui presentata.

Naturalmente la capacità di adattamento del cervello e la sua complessità data dalle reti di milioni di neuroni da cui è composto è ben lungi dall'essere imitata da un qualsiasi software e lo rende molto più resistente a fenomeni di *overfitting* rispetto a qualsiasi modello.

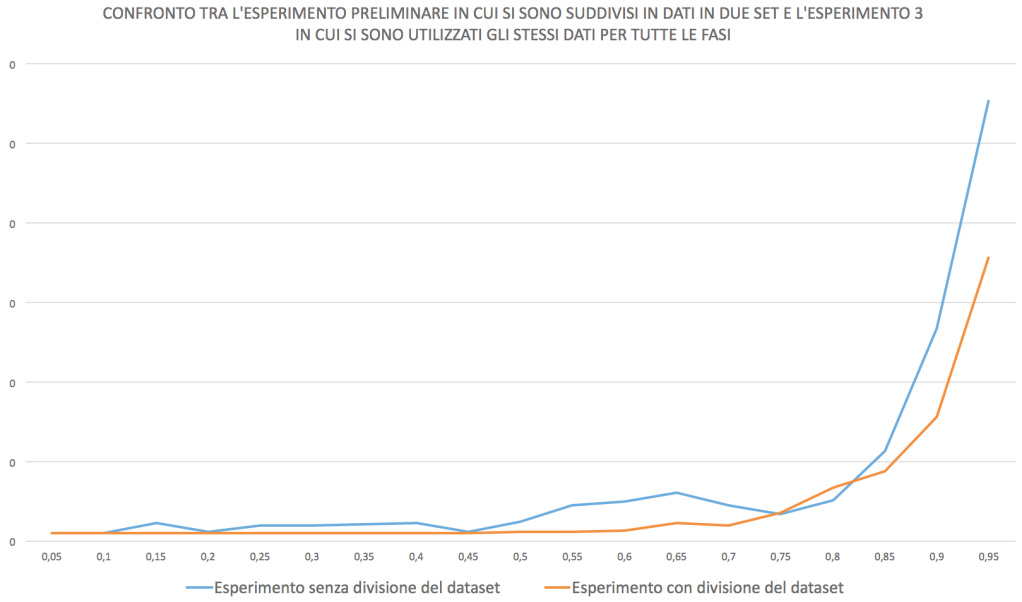


Figura 7.11 - Nel grafico è mostrato l'andamento del numero di tentativi necessari per imitare una parola in funzione della threshold. Dal confronto tra l'esperimento con divisione del dataset in training e testing e quello senza divisione si nota che il secondo impiega più cicli e dunque più tempo, a parità di threshold, per imitare una parola, in particolare per valori alti di similarità.

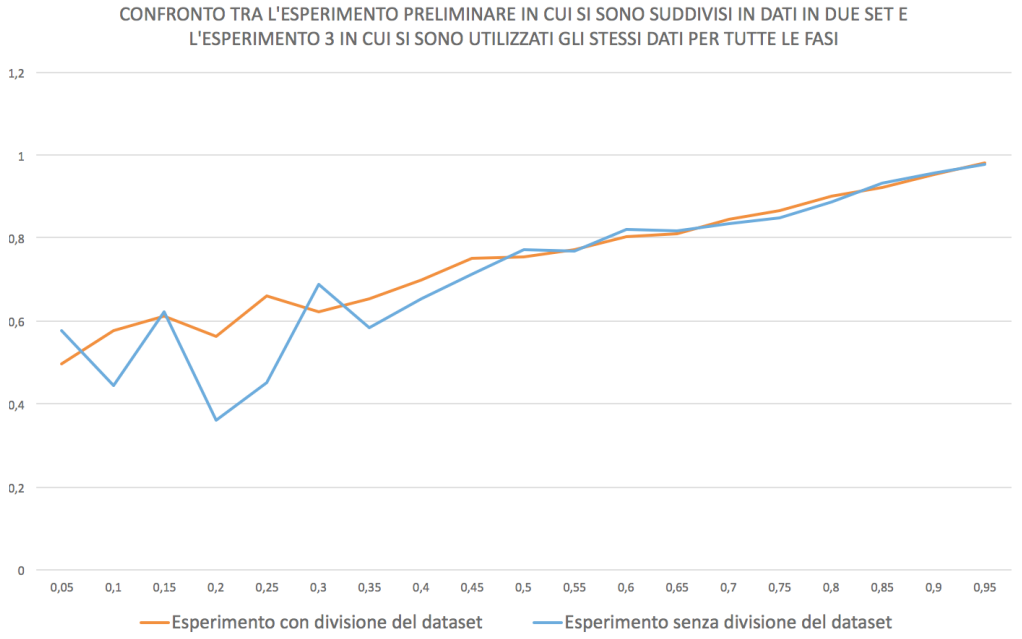


Figura 7.12 - Nel grafico è mostrato l'andamento della similarità media ottenuta in funzione della threshold. Dal confronto tra l'esperimento preliminare con suddivisione del dataset e l'esperimento 3 in cui il dataset è unico si evince che per valori bassi di threshold risulta più efficiente il primo caso.

7.7 Quarto esperimento: integrazione del sistema con IDRA

Nel quarto esperimento abbiamo voluto testare il funzionamento e le prestazioni del sistema che integra insieme il LPM e IDRA. Il sistema risultante si basa sulle stesse tecniche di apprendimento di quello precedente, ma utilizza IDRA per la gestione degli stati e per l'identificazione del grado di somiglianza tra suono target e suono prodotto, tramite processi bioispirati che simulano il funzionamento del cervello.

- dataset di training: 600 parole
- dataset di testing: 300 parole
- test effettuati: 19
- variazione della threshold per ogni test: 0.05 (da 0.05 a 0.95)
- numero parole in input: 200 parole

7.7.1 Risultati del quarto esperimento

Pur utilizzando meccanismi simili al sistema precedente, l'integrazione con IDRA modifica alcuni processi in modo piuttosto rilevante, in particolare per quanto riguarda la simulazione di amigdala, talamo e corteccia, parti che prima non erano considerate e che introducono una maggiore complessità.

Tuttavia, nonostante le differenze, i risultati ottenuti non solo sono assimilabili con quelli ottenuti negli esperimenti precedenti, ma per alcuni valori di threshold persino migliori. In particolare per soglie superiori a 0.85 le prestazioni utilizzando IDRA migliorano di molto e non crescono in modo quasi esponenziale come avviene invece per il sistema precedente.

Anche i risultati per quanto riguarda la similarità ottenuta sono perfettamente comparabili nelle due versioni; utilizzando IDRA si nota inoltre un'andamento molto più regolare. Per valori di threshold superiori a 0.9 la similarità ottenuta con IDRA è leggermente inferiore, ma pur sempre sopra lo 0.9, valore per il quale le parole sono praticamente identiche: la differenza tra le similarità con i due sistemi non è dunque rilevabile.

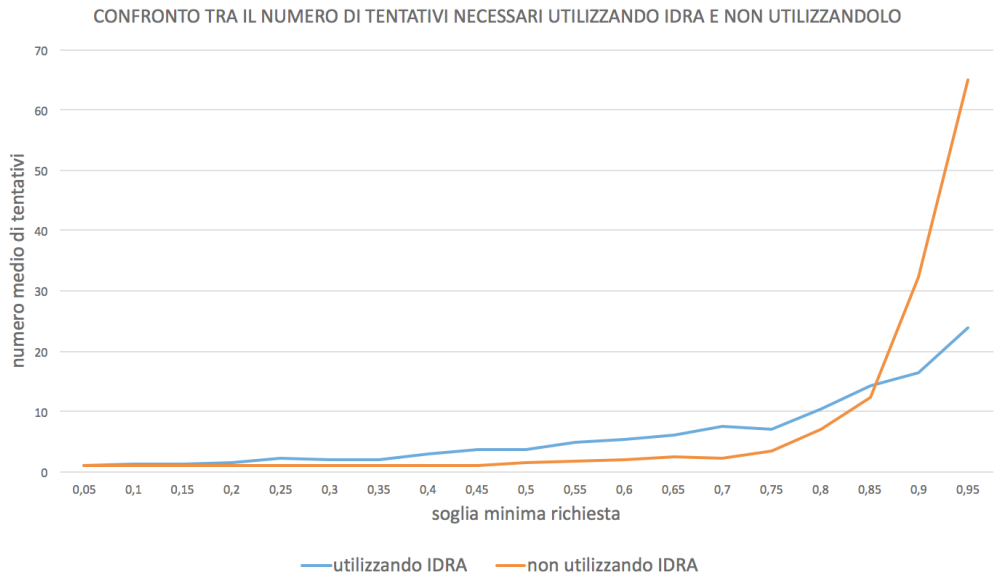


Figura 7.13 - Nel grafico è mostrato l'andamento del numero di tentativi medi ottenuto in funzione della threshold. Dal confronto tra l'esperimento utilizzando IDRA e quello senza, si nota che per valori superiori a 0.85 le prestazioni migliorano nel primo caso. Per soglie più basse i sue andamenti sono comparabili e presentano poche differenze nelle prestazioni.

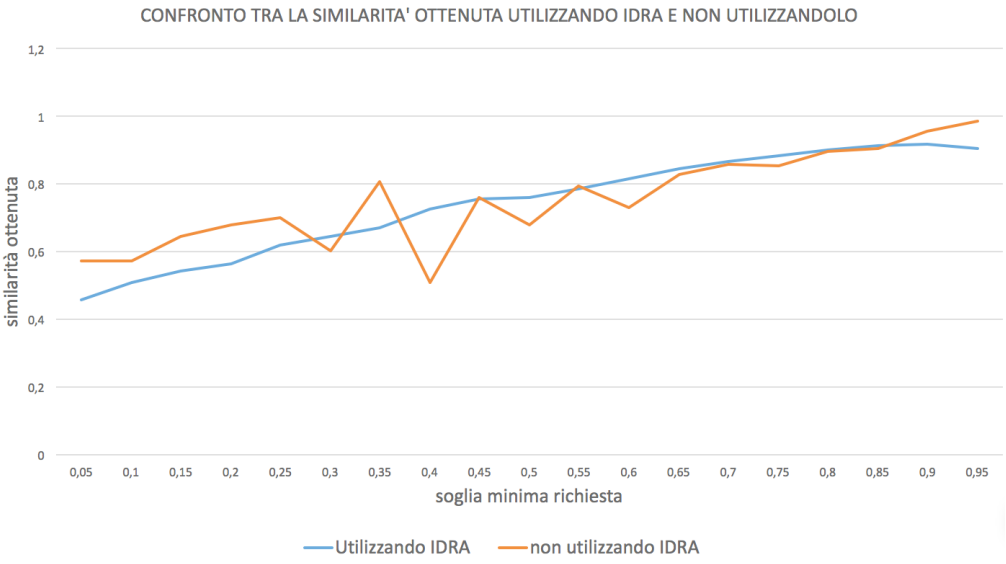


Figura 7.14 - Nel grafico è mostrato l'andamento della similarità media ottenuta in funzione della threshold. Dal confronto tra l'esperimento utilizzando IDRA e quello senza, si nota che nel primo caso l'andamento è molto più regolare, seppur comparabile con quello degli esperimenti precedenti.

Capitolo 8

Conclusioni e sviluppi futuri

"I manoscritti non bruciano."

Bulgakov da "Il Maestro e Margherita"

L'utilizzo del linguaggio naturale in robotica è stato sempre un campo di ricerca a sé stante, nato con il fine di creare interfacce intelligenti e immediate tra l'uomo e la macchina. Oggi sono disponibili diverse applicazioni di questo tipo, che vanno dai dispositivi mobili ai videogiochi, nelle quali il linguaggio rappresenta una caratteristica ricercata ma non essenziale, sfruttata in modo piuttosto limitato. Tuttavia l'importanza del linguaggio non si esaurisce unicamente nell'ambito della comunicazione: esso rappresenta l'immagine stessa della mente, è profondamente legato alla sua organizzazione interna e ne guida lo sviluppo in innumerevoli fasi, tra le quali quella dell'apprendimento.

Lo scopo di questa tesi è stato quello di compiere un primo passo in avanti verso un modello di linguaggio che possa essere integrato alle altre capacità dei robot, con il fine di cooperare con lo sviluppo della mente e con l'apprendimento: per far si che i concetti espressi dal robot nel linguaggio abbiano significato per la macchina, devono fondarsi non solo sulla conoscenza e sulle regole implementate dagli sviluppatori, ma anche sui dati acquisiti dall'automa stesso tramite l'esperienza, sfruttando un apparato sensoriale. Attraverso l'organizzazione dei concetti svolta dal linguaggio, si potrebbe riuscire a sviluppare anche la mente degli automi, raggiungendo risultati migliori di quelli ottenuti fino ad oggi, in cui di fronte a situazioni impreviste e non simulate i robot non sono del tutto in grado di affrontare delle scelte in modo autonomo.

Per ottenere questo risultato ci siamo concentrati sulle fasi iniziali dello sviluppo del linguaggio, quelle che si susseguono nei bambini durante il primo anno di vita e durante le quali si passa dalla produzione involontaria di suoni all'utilizzo volontario di sillabe e vocali, il babbling, necessario per imitare parole sentite nel mondo circostante. Per distaccarci dai modelli attuali basati principalmente su regole e sintassi predefinite impostate a priori, abbiamo scelto di implementare un modello che sia bioispirato, ossia prenda spunto

dalla biologia per risolvere il complesso problema del linguaggio naturale, con l'obiettivo di risultare più resistente di fronte a rumore e ambiguità e più efficiente.

Abbiamo così realizzato il LPM che sfrutta gli stessi principi delle primitive motorie, già utilizzati nell'ambito della robotica bioispirata, e li applica al processo di apprendimento del linguaggio. A partire da un set di suoni base (composto da coppie vocale-consonante) simile a quello disponibile ai bambini durante i primi mesi di vita, il sistema è in grado di imitare ciò che ascolta nell'ambiente circostante e apprendere le nuove parole prodotte. Si tratta dello stesso procedimento che si mette in atto nei bambini, nella fase di babbling, nel momento in cui prova ad imparare una parola nuova per tentativi. Il sistema realizzato, che si basa su principi mai utilizzati in applicazioni pratiche in questo campo fino ad ora, è studiato per essere resistente verso il rumore sia grazie alla struttura intrinseca basata su metodi statistici, sia grazie ai vari processi di analisi ed elaborazione delle onde sonore che sono implementati al suo interno.

Per testare il LPM abbiamo realizzato diversi esperimenti con il fine di valutarne le abilità di imitazione data una parola target e per capire se effettivamente il sistema fosse in grado di apprendere. I risultati ottenuti non solo validano il modello, ma sono strettamente correlati con quelli ricavati da osservazioni sperimentali su bambini reali e dunque corretti anche da un punto di vista biologico, lasciando ben sperare per possibili sviluppi futuri.

Inoltre, trattandosi di un modello bioispirato, risulta piuttosto semplice integrarlo in altri sistemi di questo tipo e collegarlo quindi ad altre funzionalità come quelle sensoriali o del movimento, integrando il linguaggio con le altre componenti della mente. Per ottenere un sistema completo abbiamo scelto di utilizzare IDRA, un software che simula il funzionamento di alcune aree del cervello (corteccia, talamo e amigdala) da un punto di vista biologico e permette di processare diversi tipi di input e di generare automaticamente nuovi obiettivi a partire da istinti innati e stimoli sensoriali.

Unendo insieme IDRA e il LPM siamo riusciti ad ottenere un sistema completo che rappresenta un primo passo verso la possibilità di sviluppare agenti in grado di apprendere tramite l'esperienza del mondo circostante, sviluppare automaticamente obiettivi e comunicare tramite il linguaggio naturale con l'uomo.

8.1 Sviluppi futuri

Sono numerosi e di diversa tipologia gli sviluppi futuri che possono partire da questa tesi, integrando il LPM ad altri sistemi, sviluppandolo ed applicandolo a contesti differenti dal babbling. Abbiamo dunque scelto di soffermarci unicamente su tre aree di sviluppo perché sono quelle più in linea

con il lavoro svolto fino ad ora e perché possono aprire le porte verso applicazioni molto interessanti.

Le prime due applicazioni riguardano più l'ambito medico e psicologico che quello della robotica: prendono in considerazione infatti la simulazione dello sviluppo del linguaggio da un punto di vista fisiologico, considerando non solo la mente ma anche la crescita e le trasformazioni del fisico dei bambini, come ad esempio quella del tratto vocale. Ricerche di questo genere possono aiutare non solo a chiarire le zone d'ombra che riguardano l'apprendimento linguistico, ma anche a studiare casi patologici approfondendone i meccanismi.

La terza applicazione riguarda invece l'ambito della robotica e in particolare l'utilizzo del nostro sistema su agenti reali e non simulati e l'integrazione di indizi visivi nel sistema, che possono accelerare lo sviluppo di abilità linguistiche, esattamente come accade nei bambini.

8.1.1 Analisi del babbling tramite formant

Nel corso della tesi abbiamo utilizzato numerose tecniche per elaborare i suoni e descritto un metodo efficace per estrarre feature, così da poterle utilizzare per la fase di imitazione [55].

L'estrazione delle caratteristiche utilizzata tuttavia non è bioispirata, ma si basa su analisi matematiche delle frequenze dell'onda e dell'andamento dell'onda. Questo non ha costituito un grave limite perché in questo modello non abbiamo preso in considerazione la parte biologica riguardante il tratto vocale e l'apparato uditivo, concentrandoci più sui processi di imitazione, la cui architettura generale risulta ispirata alla natura.

Tuttavia, se si vuole successivamente provare a costruire un modello che comprenda anche la parte biologica di sviluppo fisico dei bambini, bisogna trovare un metodo di analisi dei suoni maggiormente ispirato alla natura: questo metodo potrebbe fondarsi sul concetto dei *formant*, dei quali discutiamo alcuni punti chiave e che potrebbero costituire un valido punto di partenza per gli sviluppi futuri di questa tesi [61].

I formant sono già utilizzati nella fonetica, ed esprimono il modo in cui le vocali (o gruppi di vocali-consonanti) risuonano nel tratto vocale umano: una volta identificati, possono essere dunque utilizzati per distinguere una vocale dall'altra, senza bisogno di estrarre numerose statistiche matematiche a partire dalle forme d'onda.

Data una vocale (o vocale-consonante) si può estrarre un numero variabile di formant (nella maggior parte dei casi fino a quattro, in altri più di sei), ognuno dei quali ha un particolare significato fisico ed una frequenza crescente: è in grado quindi di considerare diversi gradi di risonanza.

Average vowel formants[4]

Vowel (IPA)	Formant F_1 (Hz)	Formant F_2 (Hz)
i	240	2400
y	235	2100
e	390	2300
ø	370	1900
ɛ	610	1900
œ	585	1710
a	850	1610
œ	820	1530
ɑ	750	940
ɒ	700	760
ʌ	600	1170
ɔ	500	700
ɤ	460	1310
o	360	640
ɯ	300	1390
u	250	595

Tabella 8.1 - La tabella mostra le possibili vocali (e alcuni gruppi vocale-consonante) e le frequenze dei primi due formant estratti.

I primi due formant (detti F1 ed F2), ad esempio, esprimono quantitativamente la vocale in termini di apertura e chiusura della bocca e posizione della lingua, mentre quelli successivi aiutano a raggiungere un'accuratezza maggiore, in particolare per i casi di parole più complesse dal punto di vista fonico, discriminando i suoni in base all'architettura delle numerose cavità oro-rino-faringee. Da un'analisi dettagliata di un gran numero di formant si potrebbero persino rilevare "sfumature" linguistiche come ad esempio forme dialettali o particolari pronunce personali.

I formant possono essere misurati in diversi modi, tra i quali estraendo i picchi di ampiezza nello spettro delle frequenze o utilizzando un'analizzatore di spettro; gli esperimenti effettuati mostrano che l'affidabilità delle previsioni sono corrette in oltre dell'80% dei casi utilizzando anche sistemi molto semplici [61]; essi, inoltre, possono essere sfruttati per la sintesi di vocali e didsegmenti vocale-consonante, che sono il fondamento del nostro set di primitive linguistiche, mentre per parole più complesse occorrono altri dati.

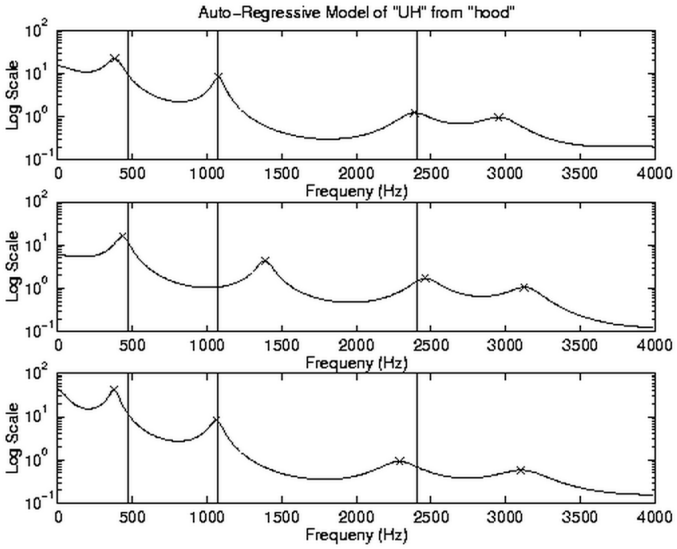


Figura 8.1 - L'immagine mostra i tre formant che rappresentano la vocale "UH"estratti da tre segmenti audio differenti, contenenti la registrazione di tre persone diverse che pronunciano la stessa parola "hood". Come si può notare sono abbastanza simili per essere riconosciuti come appartenenti alla stessa vocale.

8.1.2 Simulazione del tratto vocale biologico utilizzando un modello autoregressivo

Come abbiamo visto nel capitolo della tesi relativo allo sviluppo delle abilità vocali nei bambini, le modifiche fisiologiche del tratto vocale che avvengono durante la crescita rivestono particolare importanza nel processo di acquisizione di nuove primitive linguistiche di base. I bambini infatti possiedono un tratto vocale con una curva molto accentuata che non permette loro, inizialmente, di produrre un gran numero di suoni. Crescendo, il tratto vocale si distende e permette ai bambini, unito ad una maggior capacità motoria dei muscoli facciali, di produrre più tipologie di suono; lo sviluppo di queste consentiranno al bambino, dopo il primo anno e mezzo di vita, di riprodurre qualsiasi tipo di parola.

Nel sistema che abbiamo sviluppato per la tesi non è stato preso in considerazione il funzionamento biologico del tratto vocale, né la sua evoluzione: le primitive linguistiche iniziali che si sviluppano grazie alla conformazione fisica del tratto vocale e ai riflessi involontari sono state assunte come innate esse stesse.

Sarebbe tuttavia molto interessante, in futuro, simulare il funzionamento del tratto vocale di una persona, approssimandolo con un sistema matematico ed integrandolo nel resto dell'architettura.

La pronuncia di una primitiva linguistica potrebbe essere approssimata con un sistema composto da una serie di impulsi (che in natura è il suono generato dalle corde vocali) che passano attraverso una serie di tubi cilindrici (il tratto vocale appunto), modellizzato come un filtro [62]. Il tratto vocale è infatti formato da una serie di tubi cilindrici collegati l'uno all'altro da cartilagine e da un denso strato di muscolatura.

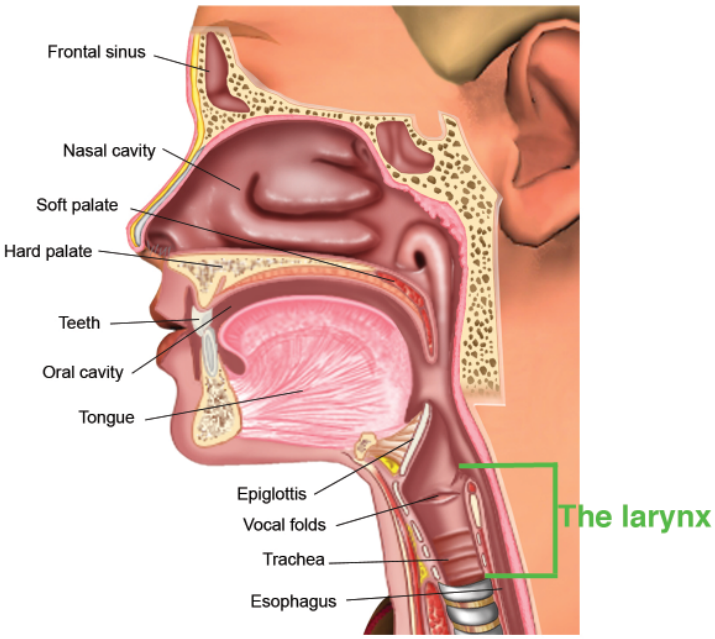


Figura 8.2 - La laringe è composta da diversi tubi cilindrici di cartilagine collegati l'uno all'altro da uno strato di muscolatura liscia.

Questa struttura, con le dovute semplificazioni, potrebbe essere modellizzata da una funzione autoregressiva, un particolare tipo di funzione che è in grado di descrivere un processo predicendo la variabile di uscita che è linearmente dipendente dai valori delle uscite precedenti [62]. Un esempio di modello autoregressivo è:

```
ye[n] = a1*y[n-1] + a2*y[n-2]... + b0*x[n] +          (8.1)
b1*x[n-1] + ...
```

dove ye[n] è la variabile predetta, (y[n-1],y[n-2]…) sono gli output precedenti e (x[n], x[n-1], x[n-2]…) sono gli input.

Per derivare il modello è necessario determinare i coefficienti a1,a2,.. e b0,b1,b2,… dell'equazione.

Utilizzare un modello autoregressivo è semplice anche perché possiede numerose implementazioni in Matlab, come ad esempio la funzione *"ar"* che

utilizza, di default, un'implementazione forward-backward e che è già disponibile nei toolbox forniti insieme all'ambiente di sviluppo.

Infine, questo modello può essere utilizzato unitamente al calcolo dei formant (il quale si può basare proprio su questo sistema), così da fondare il processo di analisi dei suoni, di imitazione e riproduzione proprio sui meccanismi biologici coinvolti nel babbling e nel successivo sviluppo del linguaggio negli esseri umani.

8.1.3 Utilizzo del sistema in un robot umanoide e integrazione della vista nel processo di apprendimento

La finalità della tesi è stata sin dall'inizio quella di presentare un modello di sviluppo del linguaggio alternativo a quelli già esistenti e soprattutto bioispirato, che potesse fungere da punto di partenza per gli studi futuri, orientati alla creazione di un sistema in grado di incrementare le proprie capacità linguistiche sulla base degli stimoli esterni ricevuti e del bagaglio di conoscenze accumulate nel tempo.

Sicuramente uno degli sviluppi futuri più interessanti sarà quello di utilizzare il LPM con un robot umanoide e testarlo in un ambiente reale, non simulato.

Il sistema IDRA, su cui ci basiamo per portare avanti il processo di imitazione, è già stato testato su un modello di robot umanoide molto utilizzato negli ambiti medici e della ricerca: il *NAO Robot* [63].

Il NAO è alto circa 60cm e possiede 21 gradi di libertà (in alcune versioni 25), che gli permettono di camminare, muovere braccia e testa ed avere una presa discreta sulle mani, sensori di pressione (ad esempio sulla testa) e un sistema multimediale piuttosto evoluto: 4 microfoni, due altoparlanti e due videocamere CMOS.

E' stato sviluppato inizialmente dalla società francese Aldebaran Robotics ed attualmente è utilizzato in molti ambiti medici (ad esempio all'ospedale San Raffaele di Milano) con fini terapeutici, principalmente nell'interazione con bambini.

Grazie ad un sistema di sintesi vocale e di controllo wireless a distanza è infatti in grado di narrare storie, scambiare battute con i bambini che si trovano più a loro agio ad interagire con un robot animato che con i medici e che possono giocarci durante la degenza.

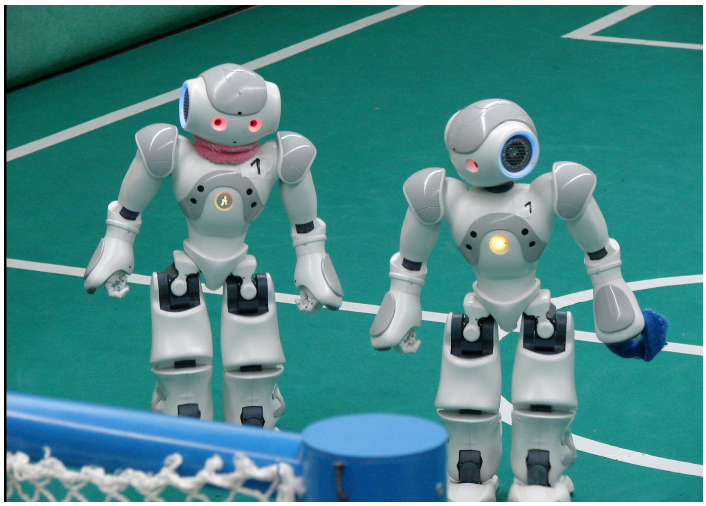


Figura 8.3 - Due modelli di NAO robot.

Inoltre il NAO viene utilizzato anche nel caso di bambini affetti da gravi forme di autismo che non riescono ad esprimersi e a costruire un rapporto con i medici, mentre mostrano di essere più aperti verso quello che può sembrare un gioco. Allo stesso tempo il robot può essere guidato dai medici e dagli psicologi a distanza e dunque utilizzato per spronare i bambini a sviluppare capacità relazionali.

Il sistema NAO supporta diversi linguaggi di programmazione (come C++, Python, Urbi, Microsoft.NET), dunque basterebbe tradurre il sistema da noi sviluppato in questa tesi in un qualsiasi linguaggio di programmazione a più alto livello per poter essere utilizzato sul robot e testato dunque in un ambiente reale.

Naturalmente, oltre al lavoro di traduzione, sarebbe necessario considerare tutti i problemi dovuti al rumore e a situazioni di utilizzo non ideali, problemi che tuttavia sono già stati previsti e trattati in parte nei moduli relativi alla normalizzazione.

Oltre a testare il LPM in un ambiente reale, utilizzare un robot reale permetterebbe di integrare diversi stimoli sensoriali nel processo di apprendimento del linguaggio, come ad esempio la vista (per osservare l'ambiente circostante) o il tatto (per ricevere feedback sul proprio operato ad esempio tramite la pressione della testa).

Il sistema IDRA che abbiamo integrato con il nostro modello è già stato testato sul robot NAO, sfruttando l'apparato visivo, per sviluppare autonomamente nuovi goal da perseguire [63].

Ad esempio, dopo aver impostato nel robot un interesse innato per gli oggetti di colore rosso, gli sono stati mostrati numerosi oggetti di questo colore tutti a forma di stella. Dopo una fase di apprendimento il sistema ha mostrato di sviluppare uno spontaneo interesse per gli oggetti con questa forma, anche in caso di colori differenti dal rosso.

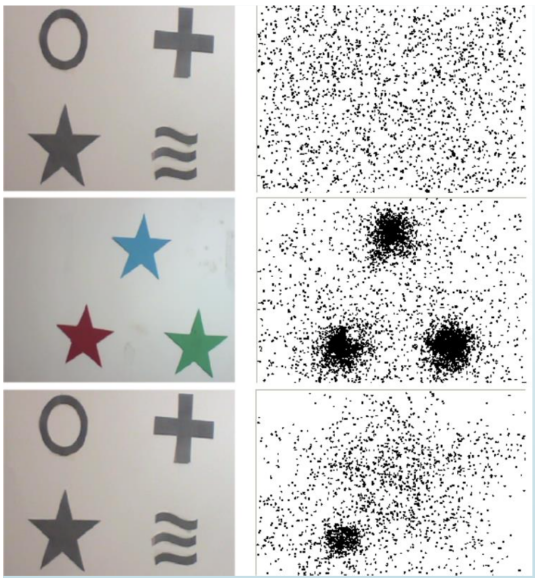


Figura 8.4 - Nella prima fase al robot vengono mostrate molte forme a stella rosse, colore per cui prova un interesse innato. Dopo la fase di apprendimento il robot sviluppa interesse per tutte le forme a stella, indipendentemente dal loro colore, come si può notare nella figura.

Come studiato approfonditamente nel capitolo sull'apprendimento del linguaggio da parte dei bambini, la vista gioca un ruolo centrale nei primi mesi di crescita e durante la fase di babbling. In particolare il meccanismo di object labeling aiuta i bambini a memorizzare una parola se associata ad un oggetto del mondo reale, o ad un particolare contesto. Trasformandola in etichetta, questa parola diventa semplice da ricordare e da distinguere dalle altre che, benché simili, sono associate ad oggetti diversi.

Negli sviluppi futuri lo stesso meccanismo potrebbe essere portato avanti, tramite il robot, utilizzando anche il concetto di primitive linguistiche: se il sistema linguistico fosse in grado di cooperare insieme a quello visivo nell'apprendimento, si potrebbe mostrare al robot una determinata forma e, allo stesso tempo, si potrebbe fargli ascoltare un suono preciso, magari un audio con registrata la parola che si riferisce a quell'oggetto.

Ogni volta che lo stesso oggetto viene mostrato al robot si pronuncia la stessa parola parola e viceversa, aspettandosi che il robot sviluppi uno spontaneo interesse per quell'oggetto ogni volta che sente pronunciare il suo nome, tralasciando tutti gli altri oggetti nelle vicinanze.

Allo stesso modo potrebbe essere in grado di associare un nome e pronunciarlo (il NAO possiede un sistema di sintesi vocale) ogni volta che vede un oggetto di cui ha sentito molte volte parlare, così come il bambino nello *Speechrome Project* che ha imparato la parola "*water*" sentendola pronunciare ogni volta in un determinato contesto.

Bibliografia

[1] Stern, J. "*Apple Siri: Loved, But Underused*", ABC News, (2012).

[2] Turing, A. "*Computing machinery and intelligence*." Mind, (1950): 433-460.

[3] Frege, G. "*Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought*." From Frege to Gödel: A source book in mathematical logic 1931, (1879): 1-82.

[4] Winfree, E. , Tony, E., et al. "*String tile models for DNA computing by self-assembly*." DNA computing. Springer Berlin Heidelberg, (2001): 63-88.

[5] Ford Dominey, P. "*How are Grammatical Constructions Linked to Embodied Meaning Representations?*." AMD Newsletter, (Fall 2013): 3.

[6] Spiliotopoulos, D. , Androutsopoulos, I., et al. "*Human-robot interaction based on spoken natural language dialogue*." Proceedings of the European Workshop on Service and Humanoid Robots, (2001).

[7] Lignos, Constantine, et al. "*Provably correct reactive control from natural language*." *Autonomous Robots,* (2014): 1-17.

[8] Roy, D., et al. "*The human speechome project*." Symbol Grounding and Beyond. Springer Berlin Heidelberg, (2006): 192-196.

[9] Saffran, J. R., et al. "*Dog is a dog is a dog: Infant rule learning is not specific to language*." Cognition 105.3 (2007): 669-680.

[10] Pastra, K. "*Autonomous Acquisition of Sensorimotor Experiences: Any Role for Language?*". AMD Newsletter, (Fall 2013): 12-13.

[11] Weng, J. " *These Questions Arose because You Used Symbolic Representations*." AMD Newsletter, (Fall 2013): 11.

[12] Taddeo, Mariarosaria, et al. "*Solving the symbol grounding problem: a critical review of fifteen years of research*." Journal of Experimental & Theoretical Artificial Intelligence 17.4 (2005): 419-445.

[13] MacDorman, K. F. "*Grounding symbols through sensorimotor integration*." Journal of the Robotics Society of Japan, 17(1), (1999): 20-24.

[14] Searle, J. R."*Minds, brains, and programs*."Behavioral and Brain Sciences 3(3) (1980): 417-457.

[15] Winograd, T. "*Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.”* MIT AI Technical Report 235. (1971)

[16] Weizenbaum, J. "*ELIZA—a computer program for the study of natural language communication between man and machine*." Communications of the ACM 9.1 (1966): 36-45.

[17] Liddy, E.D. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc. (2001)

[18] Fenstad, J. E. "*Tarski, truth and natural languages*." Annals of Pure and Applied Logic 126.1 (2004): 15-26.

[19] Singh, P. P. "*Survey of Most Powerful Language Software's*." (2014)

[20] Shen, E. "*Comparison of online machine translation tools.”* Archived from the original on February 10, 2011, retrieved December 15, 2010.

[21] Bosker, B. "*Siri rising: The Inside Story Of Siri's Origins -- And Why She Could Overshadow The iPhone*." Huffington Post. January 24, 2013. (January 27, 2013).
http://www.huffingtonpost.com/2013/01/22/siri-do-engine-apple-iphone_n_2499165.html

[22] Spiliotopoulos, D., Androutsopoulos, I. et al. "*Human-robot interaction based on spoken natural language dialogue*." Proceedings of the European Workshop on Service and Humanoid Robots. (2001).

[23] Goh, O. S. and Fung, L. "*An Intelligent Personal Robot Assistant*." arXiv preprint arXiv:1411.1170 (2014).

[24] Matuszek, Cynthia, et al. "*Learning to parse natural language commands to a robot control system*." Experimental Robotics. Springer International Publishing. (2013).

[25] Bell, David, et al. "*Microblogging as a mechanism for human–robot interaction*." Knowledge-Based Systems (2014)

[26] Chopra, A. K., Artikis, A., Bentahar, J., Colombetti, M., et al. "Research directions in agent communication." ACM Transactions on Intelligent Systems and Technology (TIST) 4.2 (2013): 20.

[27] Kollar, Thomas, et al. "*Toward understanding natural language directions*." Human-Robot Interaction (HRI) 5th ACM/IEEE International Conference on. IEEE. (2010).

[28] Vella, M. "*Robots have failed Fukushima Daiichi and Japan*". Fortune (March 20, 2013).

[29] Spenko, M. J., et al. "*Biologically inspired climbing with a hexapedal robot*." Journal of Field Robotics 25.4−5 (2008): 223-242.

[30] Pfeifer, Rolf, et al. "*Morphological computation–connecting brain, body, and environment*." Creating Brain-Like Intelligence. Springer Berlin Heidelberg. (2009): 66-83.

[31] Pfeifer, Rolf, et al. "*Designing intelligent robots-on the implications of embodiment*." 日本ロボット学会誌 24.7 (2006): 783-790.

[32] Konczak, J. "*On the notion of motor primitives in humans and robots*." (2005): 47-53.

[33] Fod, Ajo, et al. "*Automated derivation of primitives for movement classification*." Autonomous robots 12.1 (2002): 39-54.

[34] Dimitrijevic, Milan R., et al."*Evidence for a spinal central pattern generator in humansa*." Annals of the New York Academy of Sciences 860.1 (1998): 360-376.

[35] Roy, D. "*Grounded spoken language acquisition: Experiments in word learning*." Multimedia, IEEE Transactions on 5.2 (2003): 197-209.

[36] Roy, D, and Ehud Reiter. "*Connecting language to the world*." Artificial Intelligence 167.1 (2005): 1-12.

[37] Khayrallah, Huda, et al. "*Towards a Meaningful Natural Language Interface*." (2013)

[38] She, Lanbo, et al. "*Teaching Robots New Actions through Natural Language Instructions*." Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on. IEEE. (2014).

[39] Cangelosi, A.. "*Grounding language in action and perception: from cognitive agents to humanoid robots*." Physics of life reviews 7.2 (2010): 139-151.

[40] Guellaï, Streri, et al."*The development of sensorimotor influences in the audiovisual speech domain: some critical questions*." Front Psychol. (2014 Aug 6): 5-12.

[41] Yeung H.H. , Chen LM, et al. "*Referential labeling can facilitate phonetic learning in infancy.*" Child Dev. 85(3). (2014 May-Jun): 1036-49.

[42] Yeung H.H. , Werker, J. F. "*Lip movements affect infants' audiovisual speech perception*." Psychological Science (2013).

[43] Scott, Mark, et al. "*Inner speech captures the perception of external speech*." The Journal of the Acoustical Society of America 133.4 (2013): EL286-EL292.

[44] Yeung H.H. , Werker, J. F. "*Learning words' sounds before learning how words sound: 9-Month-olds use distinct objects as cues to categorize speech information*." Cognition 113 (2009): 234–243.

[45] Miyan, Kajal, et al. "*WWN-text: Cortex-like language acquisition with "what" and "where"*." Development and Learning (ICDL), 2010 IEEE 9th International Conference on. IEEE. (2010).

[46] Gleitman, Lila R., et al. "*The invention of language by children: Environmental and biological influences on the acquisition of language*." An invitation to cognitive science 1 (1995): 1-24.

[47] Oller, D. Kimbrough, et al. "*Precursors to speech in infancy: the prediction of speech and language disorders*." Journal of communication disorders 32.4 (1999): 223-245.

[48] Kuhl, Patricia K., et al."*Infant vocalizations in response to speech: Vocal imitation and developmental change*." The journal of the Acoustical Society of America 100.4 (1996): 2425-2438.

[49] Harding, Gibb, C., et al. "*The origins of intentional vocalizations in prelinguistic infants*." Child development (1979): 33-40.

[50] Oller, Kimbrough, D., et al. "*Infant babbling and speech*." Journal of child language 3.01 (1976): 1-11.

[51] Roy, D, et al. "*The human speechome project*." Symbol Grounding and Beyond. Springer Berlin Heidelberg. (2006): 192-196.

[52] Roy, D. "*New horizons in the study of child language acquisition*." (2009).

[53] Serkhane, Jihène, et al."*Simulating vocal imitation in infants, using a growth articulatory model and speech robotics*." International Congress of Phonetic Sciences (ICPhS), Barcelona, Spain. (2003).

[54] Styler, W. "*Using Praat for Linguistic Research*." University of Colorado at Boulder Phonetics Lab (2014).

[55] Giannakopoulos, Theodoros, et al."*Introduction to Audio Analysis: A MATLAB® Approach*." Academic Press. (2014).

[56] Seo, N. "*ENEE632 Project4 Part I: Pitch Detection*." (2008).

[57] Nakano, T., Goto, et al."*Frontiers of Music Information Processing. VocaListener to make the Vocaloid imitate a given song*". DTM magazine (in Japanese) (Terajima Joho Kikaku) 15 (9). (August 2008): 72–73.

[58] Laroche, Jean, et al. "*Improved phase vocoder time-scale modification of audio*." Speech and Audio Processing, IEEE Transactions on 7.3 (1999): 323-332.

[59] Kanungo, Tapas, et al. "*An efficient k-means clustering algorithm: Analysis and implementation*." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7 (2002): 881-892.

[60] Burrafato, M., Florio, L. "*A cognitive architecture based on an amygdala thalamo cortical model for developing new goals and behaviors: application in humanoid robotics*." (2012).

[61] Deterding, D. "*The formants of monophthong vowels in Standard Southern British English pronunciation*." Journal of the International Phonetic Association 27.1-2 (1997): 47-55.

[62] Serkhane, Jihène, et al. "*Simulating vocal imitation in infants, using a growth articulatory model and speech robotics*." International Congress of Phonetic Sciences (ICPhS), Barcelona, Spain. (2003).

[63] Mutti, F. "*Towards the integration of neural mechanisms and cognition in biologically inspired robots*." Diss. Italy. (2013).

Appendice A

Il Modello delle Primitive Linguistiche - Codice Matlab

A.1 System Bootstrap

Il codice riportato riguarda la fase di inizializzazione del sistema che avviene nell'ambito del sottosistema involontario: le tracce audio vengono caricate, normalizzate e fuse insieme per formare il dataset esteso. Infine viene chiama la funzione di clustering che crea gli stati, con i quali viene inizializzata la tabella stato-suono, insieme al vettore indice.

```
%load dataset
dataSet = loadAudioFiles('Dataset');

%normalize dataset channel, frequency and intensity
normDataset = normalizeDataSet(dataSet, 44000);

%normalize the pitch
newPitchFs = 440;
normVett=[];
startVett = [];
modPitch = [];

for ii = 1:size(dataSet,2)
    oldPitchFs = pitchDetector(normDataset(ii).sig,
normDataset(ii).freq);
    step = 12*log2(newPitchFs/oldPitchFs);
    shiftedNormDataset(ii).sig =
pitchShift(normDataset(ii).sig, 1024, 256, step);
    normVett = [normVett shiftedNormDataset(ii).sig];
    tempPitch = pitchDetector(shiftedNormDataset(ii).sig,
normDataset(ii).freq);
    modPitch = [modPitch tempPitch];
    difference = tempPitch-oldPitchFs;
end

%create the crossed dataset composing the language primitives
crossedDataset = [];
```

```
rr = 1;
for jj = 1:size(dataSet,2)
 for tt = 1:size(dataSet,2)
     crossedDataset(rr).sig =
crossFade(shiftedNormDataset(jj).sig ,
shiftedNormDataset(tt).sig);
     rr = rr+1;
end
end

%extract the features from the crossed dataset and create the
states making the clustering
statesNumber = 20;
[Features, normFeatures,composedDatasetClusters, Centroids,
sums, distances, Ps] = featuresClustering(crossedDataset,
statesNumber);

%inizialize the state table and the index vector
table = zeros(statesNumber, size(crossedDataset,2));
indiceVector = [];
```

A.2 Load Audio Files

Estratto del codice che effettua il caricamento delle primitive linguistiche iniziali salvandole in una struttura apposita e identificando la frequenza.

```
function dataSet = loadAudioFiles(folder, maxNumberOfFiles)
% loadAudioFiles: load all the audio .wav files contained in
the specified folder
%   input: the target directory
%   output: the dataset as structure, sig and sampling
frequency

d=dir([folder '/*.wav']);

if(nargin == 2)
    maxNumberOfFiles = min([maxNumberOfFiles; length(d)]);
else
    maxNumberOfFiles = length(d);
end

for ii=1:maxNumberOfFiles
    fname=d(ii).name;
    [sig , freq] = audioread(fname);
    % Only one channel signal
    dataSet(ii).sig = sum(sig,2);
    % Frequency of the audio file
    dataSet(ii).freq = freq;
end
```

A.3 Normalize Dataset

Questa funzione si occupa della normalizzazione delle primitive linguistiche. Il particolare esegue il resampling e rende le ampiezze dei suono omogenee, sottraendo la media (dunque centrando il segnale) e uniformando l'errore quadratico medio per evitare distorsioni del suono.

```
function dataSet = normalizeDataSet(dataSet, newFs)
% normalizeDataSet: normalize the given dataSet;
%                1. resample all the signals;
%                2. remove the mean;
%                3. normalize to [-1;1]
%   input: the dataset and a target sampling frequency (if not
specified
%   newFs = 44100
%   output: the normalized dataset as structure, signal and
sampling
%   frequency

RMSs = [];
rmsVec = [];
maxScale = [];
if(nargin<2)
    newFs = 44100;
end

for ii=1:length(dataSet)
    [P,Q] = rat(newFs/dataSet(ii).freq);
    % Resample the audio file
    dataSet(ii).sig = resample(dataSet(ii).sig ,P,Q);
    dataSet(ii).freq = newFs;


    %scales and shifts the sound vectors so they have a max
amplitude of one and have an average value of zero
    dataSet(ii).sig = (dataSet(ii).sig -
mean(dataSet(ii).sig));
    rmsVec = [rmsVec sqrt(sum(dataSet(ii).sig.^2)/
length(dataSet(ii).sig))];
    maxScale = [maxScale 0.999 / max(abs(dataSet(ii).sig))];
end

RMSs = maxScale .* rmsVec;
targetRMS = min(RMSs);
for ii=1:length(dataSet)
    dataSet(ii).sig = dataSet(ii).sig*targetRMS/rmsVec(ii);
end
```

A.3.1 Pitch Detector

Questa funzione è utilizzata dal sottosistema involontario nell'ambito della normalizzazione del pitch: applicando il metodo Cepstrum viene identificato il pitch delle tracce audio.

```
function freq = pitchDetector(x, fs)
% Extract the pitch of the signal x
% INPUT x = audio signal, fs = frequency
% OUTPUT the pitch frequency

 N = length(x);
 x = x(:) .* hamming(N);
 y = fft(x, N);
 c = ifft(log(abs(y)+eps));

 % search for maximum  between 2ms (=500Hz) and 20ms (=50Hz)
 ms2=floor(fs*0.002); % 2ms
 ms20=floor(fs*0.02); % 20ms
 [maxi,idx]=max(abs(c(ms2:ms20)));
 freq = fs/(ms2+idx-1);
```

A.3.2 Pitch Shifter

Questa funzione è utilizzata dal sottosistema involontario nell'ambito della normalizzazione del pitch: effettua prima il framing del segnale e poi lo shifting di un numero di step tale da potare il pitch a 440 Hz.

```
function outputVector = pitchShift(inputVector, windowSize,
hopSize, step)
% pitchShift: takes a vector of samples in the time-domain and
shifts the pitch
% by the number of steps specified. Each step corresponds to
half a tone.


hopOut = round((2^(step/12))*hopSize);
%Creation of the window
wn=.5*(1 - cos(2*pi*(0:windowSize-1)'/(windowSize)));
y = createFrames(inputVector,hopSize,windowSize);
output = zeros(size(y,1),windowSize);
phaseCumulative = 0;
previousPhase = 0;

for indice=1:size(y,1)
    currentFrame = y(indice,:);
    %Window the frame
    currentFrameWindowed = currentFrame .* wn'/
sqrt(((windowSize/hopSize)/2));% / (hopSize/windowSize); %
    %Get the FFT of the fftshif
```

```
    currentFrameWindowedFFT =
fft(fftshift(currentFrameWindowed));
    %Get the magnitude
    magFrame = abs(currentFrameWindowedFFT);
    %Get the angle
    phaseFrame = angle(currentFrameWindowedFFT);
    %Get the phase difference
    deltaPhi = phaseFrame - previousPhase;
    previousPhase = phaseFrame;
    deltaPhiPrime = deltaPhi - hopSize * 2*pi*(0:
(windowSize-1))/windowSize;
    deltaPhiPrimeMod = mod(deltaPhiPrime+pi, 2*pi) - pi;
    % Get the true frequency
    trueFreq = 2*pi*(0:(windowSize-1))/windowSize +
deltaPhiPrimeMod/hopSize;
    %Get the final phase
    phaseCumulative = phaseCumulative + hopOut * trueFreq;
    % Produce output frame
    outputFrame = fftshift(real(ifft(magFrame .*
exp(j*phaseCumulative))));
    output(indice,:) = outputFrame .* wn' / sqrt(((windowSize/
hopOut)/2));
end
%Overlap add in a vector
outputTimeStretched = fusionFrames(output,hopOut);
%Resample with linear interpolation
outputTime = interp1((0:
(length(outputTimeStretched)-1)),outputTimeStretched,
(0:2^(step/12):(length(outputTimeStretched)-1)),'linear');
outputVector = outputTime;
```

A.4 Cross Fade

Questa funzione è utilizzata dal sottosistema involontario per concatenare in modo omogeneo le primitive linguistiche di base ed ottenere il dataset esteso.

```
function crossFaded = crossFade(y1 , y2)

% cross fade of y1 and y2 sounds

% Create Cross fade half width of wave y1 for xfade window

xfadewidth = floor(length(y1)/1.5);
ramp1 = (1:xfadewidth)/xfadewidth;
ramp2 = 1 - ramp1;

%apply crossfade centered over the join of y1 and y2

xramp1 = [ones(1,ceil(length(y1)-xfadewidth/2)), ramp2,
zeros(1,ceil(length(y2)-xfadewidth/2))];
xramp2 = 1- xramp1;
```

```
% Create two period waveforms to fade between

ywave1 = [y1 , zeros(1,length(xramp1)-length(y1))];
ywave2 = [zeros(1,length(xramp2)-length(y2)), y2];

crossFaded = xramp1.*ywave1 + xramp2.*ywave2;
```

A.5 Clustering and Feature Extraction

Questa funzione è utilizzata dal sottosistema involontario per estrarre, dal dataset esteso di primitive linguistiche, gli stati di partenza che servono per inizializzare la tabella stato-suono. Viene utilizzato K-Means come algoritmo di clustering.

```
function [Features, Clusters, Centroids,  Ps] =
featuresClustering(dataset, clusterNumber)

% ARGUMENTS:
%  - dataset:           dataset with the audio signal of the
primitives
%  - clusterNumber      the number of the clusters to create
%
% RETURNS:
%  - Features:          contains the features matrix
%  - Clusters:          contains the centroid's indices
%                       feature of all data
%  - Centroids:         returns the centroids vectors
%  - Ps:                contains the probability that each
vector belong to a specific cluster

Features = featureExtractionDataset(dataset);

normFeatures = Features;

normFeatures = normFeatures';

[Clusters, Centroids, sums, distances] = kmeans(normFeatures,
clusterNumber);

% Distance to probability estimation:
Ps = tanh(1 ./ distances);
```

A.5.1 Short term Features Extraction

Questa funzione è utilizzata dal sottosistema involontario per estrarre dai suoni le caratteristiche di short term. Ogni finestra viene suddivisa in numerosi segmenti sovrapposti, sui quali vengono applicati diversi metodi per estrarre le feature. Sono attivati solo i metodi che risultano più utili nel nostro caso.

```
function Features = stFeatureExtraction(signal, fs, win, step)

% function Features = stFeatureExtraction(signal, fs, win,
step)
%
% This function computes basic audio feature sequencies for an
audio
% signal, on a short-term basis.
%
% ARGUMENTS:
%  - signal:    the audio signal
%  - fs:        the sampling frequency
%  - win:       short-term window size (in seconds)
%  - step:      short-term step (in seconds)
%
% RETURNS:
%  - Features: a [MxN] matrix, where M is the number of
features and N is
%  the total number of short-term windows. Each line of the
matrix
%  corresponds to a separate feature sequence
%
% Based on the work of T. Giannakopoulos, A. Pikrakis

% if STEREO ...
if (size(signal,2)>1)
    signal = (sum(signal,2)/2); % convert to MONO
end

% convert window length and step from seconds to samples:
windowLength = round(win * fs);
step = round(step * fs);

curPos = 1;
L = length(signal);

% compute the total number of frames:
numOfFrames = floor((L-windowLength)/step) + 1;
% number of features to be computed:
numOfFeatures = 17;
Features = zeros(numOfFeatures, numOfFrames);
Ham = window(@hamming, windowLength);
mfccParams = feature_mfccs_init(windowLength, fs);

for i=1:numOfFrames % for each frame
    % get current frame:
```

```
    frame  = signal(curPos:curPos+windowLength-1);
    frame  = frame .* Ham;
    frameFFT = getDFT(frame, fs);

    if (sum(abs(frame))>eps)
        % compute time-domain features:

        %zero-crossing
        Features(1,i) = feature_zcr(frame);
        %energy
        Features(2,i) = feature_energy(frame);

        %Features(3,i) = feature_energy_entropy(frame, 10);

        % compute freq-domain features:

        %centroid
        if (i==1) frameFFTPrev = frameFFT; end;
        [Features(3,i) Features(4,i)] = ...
            feature_spectral_centroid(frameFFT, fs);

        %Features(6,i) = feature_spectral_entropy(frameFFT,
10);
        %Features(7,i) = feature_spectral_flux(frameFFT,
frameFFTPrev);
        %Features(8,i) = feature_spectral_rolloff(frameFFT,
0.90);

        %Mel-Frequency Cepstrum Coefficients (MFCCs)
        MFCCs = feature_mfccs(frameFFT, mfccParams);
        Features(5:17,i)  = MFCCs;

        %[HR, F0] = feature_harmonic(frame, fs);
        %Features(22, i) = HR;
        %Features(23, i) = F0;
        %Features(23+1:23+12, i) = feature_chroma_vector(frame,
fs);
    else
        Features(:,i) = zeros(numOfFeatures, 1);
    end
    curPos = curPos + step;
    frameFFTPrev = frameFFT;
end
Features(17, :) = medfilt1(Features(17, :), 3);
```

A.5.2 Mid term features extraction

Questa funzione è utilizzata dal sottosistema involontario per estrarre dai suoni le statistiche sui segmenti mid term, rappresentati, dopo la fase di short term, da un vettore di feature.

```
function [mtFeatures, shortFeaturesCell] = ...
    mtFeatureExtraction(stFeatures, listOfStatistics)

% This function is used for extracting mid-term statistics
%
% ARGUMENTS:
%  - stFeatures:        a matrix that contains all short-term %
%                       feature vectors
%                       (dimension: dFeatures x
%                       numOfShortTermWindows)
%  - listOfStatistics:  a cell array that contains the names of
%                       the statistics to be calculated
%
% RETURNS:
%  - mtFeatures:        a matrix whose columns contains the
%                       mid-term
%                       feature statistics for each mid-term
%                       segment
%  - shortFeaturesCell: a cell array, in which each element is
%                       a matrix that contains the feature
%                       vector sequences
%                       of the corresponding mid-term segment.

[numOfFeatures, numOfStWins] = size(stFeatures);
numOfMidFrames = 1;

mtFeatures = zeros(numOfFeatures * length(listOfStatistics),
numOfMidFrames);
if (nargout==2)
    shortFeaturesCell = cell(1, numOfMidFrames);
end

    CurStFeatures  = stFeatures;
    if (nargout==2)
        shortFeaturesCell{1} = CurStFeatures;
    end
    for (j=1:length(listOfStatistics))
        mtFeatures( (j-1)*numOfFeatures + 1: j*numOfFeatures,
1) = ...
            computeStatistic(CurStFeatures',
listOfStatistics{j});
    end

function S = computeStatistic(seq, statistic)
    if strcmpi(statistic, 'mean')
        S = mean(seq); return;
    end
```

```
    if strcmpi(statistic, 'median')
        S = median(seq); return;
    end
    if strcmpi(statistic, 'std')
        S = std(seq); return;
    end
    if strcmpi(statistic, 'stdbymean')
        S = std(seq) ./ (mean(seq)+eps); return;
    end
    if strcmpi(statistic, 'max')
        S = max(seq); return;
    end
    if strcmpi(statistic, 'min')
        S = min(seq); return;
    end
    if strcmpi(statistic, 'meanNonZero')
        for i=1:size(seq, 2)
            curSeq = seq(:, i);
            S(i) = mean(curSeq(curSeq>0));
        end
        return;
    end
    if strcmpi(statistic, 'medianNonZero')
        for i=1:size(seq, 2)
            curSeq = seq(:, i);
            S(i) = median(curSeq(curSeq>0));
        end
        return;
    end
```

A.6 Babbling Module

Questa funzione fa parte del sottosistema volontario ed è responsabile dell'apprendimento. Partendo dagli output del sistema involontario e dalla parola target da imitare, viene compilata la tabella stato-suono, effettuando diversi tentativi di imitazione e registrando i risultati ottenuti.

```
function [PsCentroid, centroidDistances, table, BabbleIndicees,
bestBabble, bestBabbleSimilarity, nBabbling] =
babbling(crossedDataset, centroidsVect, table, BabbleIndicees,
target, threshold)

% Create the state-babbling table
%
% ARGUMENTS:
%  - centroidsVect:       the features vectors of the centroids
%  - target               target word
%
% RETURNS:
%  - Table:              the state-babbling table
```

```
%extract the features of the target
targetFeatures = featureExtractionDataset(target);

targetFeatures = targetFeatures';

%calculate the similarity with the centroid vectors
for ii = 1:size(centroidsVect,1)
      centroidDistances(ii) = norm(centroidsVect(ii,:) -
      targetFeatures(1,:), 2);
end

centroidDistances = centroidDistances.^2;

% Distance to probability estimation:
PsCentroid = tanh(1 ./ centroidDistances);

%find the current state in the table
currentState = find(PsCentroid ==
max(PsCentroid));BabbleIndicees

%parameters initialization
condition = 0;
start = 0;
endState=0;
ii=0;
jj=0;
tt=0;

%try to use the best babble in the current state
if table(currentState, 1 ) ~= 0
    tempBest = find(table(currentState, : ) ==
      max(table(currentState, : )));BabbleIndicees
    tempTarget(1) = crossedDataset(BabbleIndicees(tempBest));

    tempBabbleFeatures = featureExtractionDataset(tempTarget);
    tempBabbleFeatures = tempBabbleFeatures';

    tempBabbleDistance = norm(tempBabbleFeatures(1,:) -
      targetFeatures(1,:), 2);BabbleIndicees
    PsTempBabble = tanh(1 ./ tempBabbleDistance);BabbleIndicees

    %the babble is very similar
    if PsTempBabble > threshold
        %update the value of the best babble in the table
        table(currentState, tempBest ) = PsTempBabble;
        bestPs = find(table(currentState, : ) ==
        max(table(currentState, : )));BabbleIndicees
        bestBabble = BabbleIndicees(bestPs);BabbleIndicees
        bestBabbleSimilarity = table(currentState, bestPs);
        condition = 1;
        nBabbling = 1;
        endState=1;
        %the babble is NOT very similar
        %try to use the other babbles in the current state
    else
```

```
        table(currentState, tempBest ) = PsTempBabble;
        for jj = 1:size(BabbleIndicees,2)

            jj;BabbleIndicees
            tempTarget(1) = crossedDataset(BabbleIndicees(jj));

            tempBabbleFeatures =
            featureExtractionDataset(tempTarget);
            tempBabbleFeatures = tempBabbleFeatures';

            tempBabbleDistance = norm(tempBabbleFeatures(1,:) -
            targetFeatures(1,:), 2);
            PsTempBabble = tanh(1 ./ tempBabbleDistance);

            % update the babble in the table
            table(currentState, jj ) = PsTempBabble;

            if PsTempBabble > threshold
                condition = 1;
                bestPs = find(table(currentState, : ) ==
                max(table(currentState, : )));BabbleIndicees
                bestBabble =
                BabbleIndicees(bestPs);BabbleIndicees
                bestBabbleSimilarity = table(currentState,
                bestPs);
                nBabbling = jj;
                endState=1;
                break;
            end

        end
        start=size(BabbleIndicees,2);
    end
    % the row is empty (that is you are for the first time in
that state), find
    % new babble OR there isn't good value in the table
elseif table(currentState, 1 ) == 0
    % try different babbling until the threshold or the max
number of try is reached
    PsCurrentBabble = 0;

    for ii = 1:size(BabbleIndicees,2)
        ii;BabbleIndicees
        currentBabbleIndice = BabbleIndicees(ii);
        currentBabble(1) = crossedDataset(currentBabbleIndice);
        currentBabbleFeatures =
      featureExtractionDataset(currentBabble);
        currentBabbleFeatures = currentBabbleFeatures';

        currentBabbleDistance = norm(currentBabbleFeatures(1,:)
      - targetFeatures(1,:), 2);
        PsCurrentBabble = tanh(1 ./ currentBabbleDistance);

        % save the new babble in the table
        table(currentState, ii) = PsCurrentBabble;
```

```
        if PsCurrentBabble > threshold
            condition = 1;
            bestPs = find(table(currentState, : ) ==
            max(table(currentState, : ))); BabbleIndicees
            bestBabble = BabbleIndicees(bestPs); BabbleIndicees
            bestBabbleSimilarity = table(currentState, bestPs);
            nBabbling = ii;
            endState=1;
            break;
        end
    end

    start = size(BabbleIndicees,2);

end

if condition == 0
    condition;
    % try different babbling until the threshold or the max
      % number of tries is reached
    PsCurrentBabble = 0;

    for tt = 1:(900-size(BabbleIndicees,2))
        tt;BabbleIndicees

        for k = 1:10000
            %obtain a new indice and check if it is already
            %present in the vector
            currentBabbleIndice = randi(length(crossedDataset),
            1);
            if sum(BabbleIndicees == currentBabbleIndice) == 0
                break;
            end
        end

        BabbleIndicees = [BabbleIndicees currentBabbleIndice];

        currentBabble(1) = crossedDataset(currentBabbleIndice);
        currentBabbleFeatures =
      featureExtractionDataset(currentBabble);
        currentBabbleFeatures = currentBabbleFeatures';

        currentBabbleDistance = norm(currentBabbleFeatures(1,:)
      - targetFeatures(1,:), 2);
        PsCurrentBabble = tanh(1 ./ currentBabbleDistance);

        % save the new babble in the table
        table(currentState, start + tt) = PsCurrentBabble;

        if PsCurrentBabble > threshold
            nBabbling = tt + ii + jj;

             bestPs = find(table(currentState, : ) ==
max(table(currentState, : )));BabbleIndicees
```

```
            bestBabble = BabbleIndicees(bestPs);BabbleIndicees
            bestBabbleSimilarity = table(currentState,
bestPs);
            endState=1;
            break;
        end
    end

    if endState == 0
    disp('No similar babbling for that threshold')
    end

end
```