UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

Automatic Speech Recognition negli assistenti vocali, funzionamento e confronto

Candidato: *Matteo Colella*

Relatore: *Maria Simi*

Correlatore: *Mirko Luigi Aurelio Tavosanis*

Anno Accademico 2018 - 2019

Indice

A Giulia,
per avermi sempre
annodato la cravatta
senza stringerla al massimo
nonostante mille ragioni.

Introduzione

L'oggetto di questo lavoro è il sistema di *Automatic Speech Recognition* (ASR) impiegato nell'assistente vocale italiano **Bixby** di Harman-Samsung, sul quale ho lavorato durante il tirocinio curricolare.
Gli scopi che mi prefiggo di raggiungere sono dimostrare come il sistema di ASR alla base dell'assistente sia migliorato grazie alle attività svolte durante il tirocinio, e tracciare la direzione da intraprendere per incrementare le prestazioni attraverso un'analisi degli errori su due *test-set* differenti.
Per raggiungere gli obiettivi, ho innanzitutto descritto il funzionamento dell'ASR di Bixby contestualizzandone i processi; dopodiché ho confrontato le prestazioni del sistema con quelle del concorrente più diffuso ed *intelligente*[1], Assistente Google, in due corpus aventi domini diversi.

Le motivazioni e l'importanza della ricerca sono facilmente intuibili: siamo entrati nel periodo in cui il linguaggio naturale si configura come successore virtuoso delle interfacce classiche, le quali, seppur mature nel loro ambito mostrano ormai il fianco alla vita dinamica ed agli scenari d'utilizzo moderni degli strumenti d'informazione.
Come appassionato di nuove tecnologie e come informatico umanista ho quindi avvertito l'interesse e la necessità di esplorare il settore degli assistenti vocali e del riconoscimento del linguaggio.

[1] Uno studio (Yu, Shi, Yu 2017) stima l'intelligenza di sette assistenti virtuali calcolandone il QI tramite metodo Binet-Simon.

“**Human language is the new user interface layer**”

- Satya Nadella

Capitolo 1

L’assistente vocale

1.1 Contesto

Con l’avvento delle *Graphical User Interface* (GUI), i personal computer sono diventati uno strumento accessibile a tutti.
Chiunque, oggi, ha familiarità col paradigma di finestre, icone, menu e puntatori. Questi concetti sono stati ereditati dalle nuove tipologie di dispositivi interattivi affermatisi negli anni, ed hanno adattato punti di forza e debolezze del paradigma al mezzo ed all’utilizzo che ne viene fatto.

Si potrebbe tuttavia sostenere che la GUI stia via via diventando eccessivamente sovraffollata e meno intuitiva, in quanto deve supportare un numero crescente di funzionalità. Spesso ci troviamo a spulciare menu fin troppo intricati o schermate sovraffollate di icone difficilmente organizzabili. La situazione è aggravata nei dispositivi con schermi più piccoli e in contesti in cui mani e occhi sono già occupati (esempio: alla guida), sebbene si tenda parallelamente a semplificare le interfacce grafiche.
Anche la curva di apprendimento per interfacciarsi con i sistemi digitali sta diventando man mano più ripida, poiché abbiamo a che fare con un numero sempre maggiore di dispositivi e servizi.

Questo problema crescente, a cui spesso si fa riferimento col nome di *Digital Overload*, motiva la tendenza verso un nuovo paradigma, ovvero l'utilizzo del

linguaggio naturale come *layer*, intermediario tra utente e complessità dell'interfaccia. Si parla allora di *Language User Interface* (LUI).
La LUI, indipendentemente dal dispositivo o dall'applicazione, viene percepita dall'utente in maniera familiare su ogni mezzo, creando di fatto un'esperienza uniforme in tutti i dispositivi che la adottano. L'interfaccia viene ulteriormente resa più naturale assegnandole un'identità, quella di assistente digitale, assistente virtuale, assistente vocale che dir si voglia e dandole un nome.
"Siri", "Cortana", "Alexa" e "OK, Google" sono nomi già noti e suggeriscono quanto interesse e quali investimenti siano già stati effettuati per inseguire questa tendenza.

1.2 Vantaggi

Perché, dunque, in un dispositivo informatico l'interfaccia vocale dovrebbe essere preferibile ad una classica interfaccia grafica?
Fondamentalmente per cinque motivi:

L'interazione vocale è:

1. **Naturale**, non richiede nessuno specifico addestramento.
2. **Flessibile**, lascia liberi mani ed occhi.
3. **Efficiente**, considerando il rapporto tra informazioni scambiate e tempo di trasmissione.
4. **Economica**, sia per costo di trasmissione/ricezione che per apparecchiature necessarie.
5. **L'Unica scelta** in alcune circostanze, come ad esempio al volante o durante l'attività fisica.

Se l'assistente vocale avesse un manuale, esso conterrebbe una sola indicazione: "Pronuncia o digita quello che vuoi".

Fin dagli albori dell'informatica, l'idea di interagire con i computer attraverso la voce è stata un ideale, rimasto irrealizzabile a causa di limitazioni tecniche importanti.
Con l'avvento delle interfacce grafiche il sogno è stato rimosso dall'immaginario collettivo, in favore di tastiere, mouse, pennini e quant'altro; seppure di tanto in tanto qualche designer ci ha ricordato che questa possibilità esiste: è il caso della

presentazione del concept di “Knowledge Navigator” immaginato da Apple nel 1987.

1.3 Panorama attuale

La citazione in apertura capitolo di Satya Nadella, amministratore delegato di Microsoft, è stata pronunciata durante l’evento di presentazione di un nuovo aggiornamento del sistema operativo Windows, che potenziava le capacità e l’integrazione dell’assistente virtuale Cortana[2].
L’assistente in questione attualmente è preinstallato in ogni personal computer che abbia una recente versione del sistema operativo Windows a bordo.
A gennaio 2019 la base di installazione di Cortana superava i 400 milioni di unità; cifre simili sono riscontrabili nella base di installazione dei principali assistenti della concorrenza.

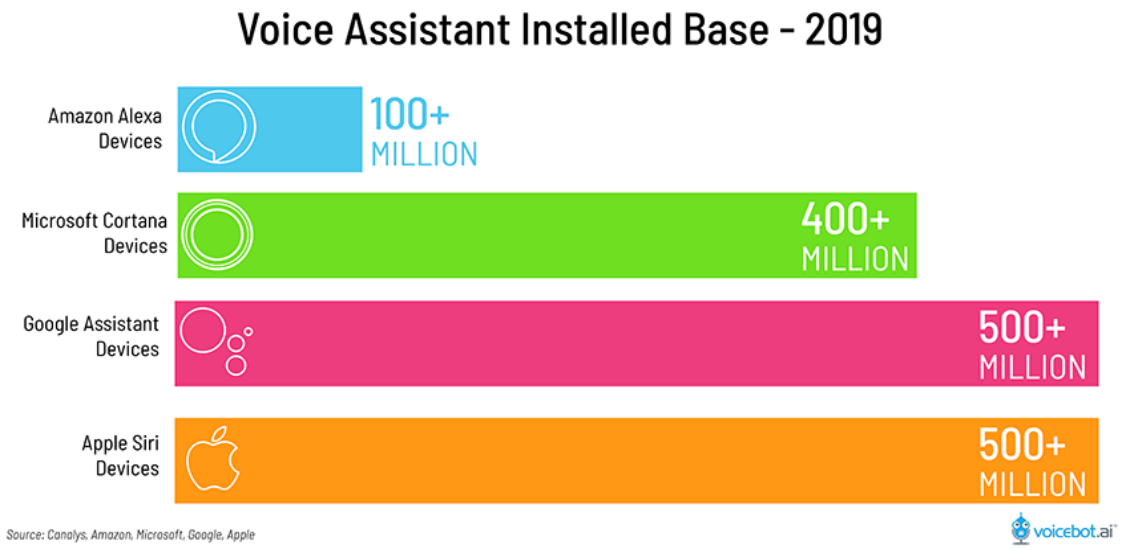

Figura 1 - Base di installazione di alcuni assistenti vocali, gennaio 2019. Fonti: Canalys, Amazon, Microsoft, Google, Apple.

L’adozione può vantare numeri così importanti anche grazie al contributo delle vendite di un nuovo genere di dispositivo, progettato appositamente per rendere disponibili gli assistenti vocali anche in assenza di personal computer o

[2] La presentazione in questione è quella di Windows Anniversary Update, rilasciato il 2 agosto 2016.

smartphone. Si tratta degli *Smart Speaker*, altoparlanti resi "intelligenti" dall'assistente virtuale che li governa.

Tali successi, riscossi nel mercato internazionale, hanno portato le aziende leader ad investire nel supporto ad altre lingue per gli assistenti.
In Italia, al momento, il mercato è conteso da almeno cinque grandi aziende e relativi prodotti:

1. Amazon - "Alexa"
2. Google - "Assistente Google"
3. Apple - "Siri"
4. Microsoft - "Cortana"
5. Samsung - "Bixby"

La creazione di un gruppo di lavoro per localizzare in italiano l'assistente virtuale di Samsung, Bixby, è stata l'occasione che mi ha permesso di apprendere i dettagli del funzionamento dell'assistente e dei suoi componenti.

Come anticipato nell'introduzione, nelle prossime pagine verranno trattati argomenti inerenti a Bixby e, più nello specifico, al suo componente di riconoscimento del linguaggio verbale, più noto come *Automatic Speech Recognition* (ASR).

Capitolo 2

Bixby e l'ASR

2.1 Bixby, piattaforma ed ecosistema

Bixby è un assistente vocale che offre nuovi modi per facilitare la vita degli utenti. Samsung, l'azienda che l'ha ideata, afferma che gli utenti grazie all'assistente eseguiranno compiti in maniera più efficiente, naturale e personalizzata.
La caratteristica cardine dell'assistente è l'adattabilità: Bixby è stato progettato tenendo presenti i concetti di domotica e di ecosistema. L'idea è quella di avere la stessa piattaforma d'assistenza vocale su ogni apparecchio personale e casalingo, creando una rete di interazioni che amplificano le potenzialità del singolo dispositivo.
Al momento la versione italiana dell'assistente, lanciata ufficialmente il 21 febbraio 2019, è installata esclusivamente sugli ultimi modelli di smartphone Samsung, sebbene a breve dovrebbe arrivare su televisori, smartwatch ed elettrodomestici dello stesso marchio. La base di installazione può quindi contare su grandi numeri: si parla di milioni di dispositivi pronti *out-of-the-box*, più alcune centinaia di migliaia già in mercato che potranno sfruttare la piattaforma tramite un aggiornamento software (Gartner Inc. 2019).

La piattaforma per lo sviluppo dei servizi è aperta a chiunque: gli strumenti forniti agli sviluppatori di Bixby sono stati resi pubblici nel novembre 2018.
Sviluppare per Bixby è un processo coadiuvato dalla piattaforma: quando un utente effettua una richiesta, Bixby utilizza dei modelli per costruire dinamicamente la risposta.
Gli sviluppatori 'insegnano' a Bixby i concetti e le azioni che i loro servizi possono eseguire creando le cosiddette Capsule, all'interno delle quali definiscono gli elementi per generare la risposta. Ogni Capsula contiene tutto quello che l'assistente ha bisogno di conoscere, dalle definizioni dei modelli e dei casi d'uso, a dialoghi e layout.
La Capsula viene poi collegata dallo sviluppatore a varie *Application Programming Interface* (API) di terze parti ed addestrata tramite esempi di richieste in linguaggio naturale per creare l'esperienza di conversazione.
Permettendo a Bixby di associare richieste degli utenti a concetti ed azioni presenti nella Capsula, l'assistente imparerà ad interpretare efficacemente nuove parole e richieste in linguaggio naturale.

2.2 L'attivazione di Bixby

L'interazione tra utente e assistente avviene tramite linguaggio naturale: un comando di risveglio (*wake-up word*) attiverà l'assistente, che ascolterà le richieste dell'utente ed eseguirà operazioni ed elaborerà risposte.

Il comando di risveglio attuale, "Hey Bixby", al momento non è personalizzabile, tuttavia lo diverrà in futuro.
A questo proposito durante il tirocinio mi è stato richiesto di individuare parole chiave alternative per il risveglio. Un'idea presa in considerazione è stata quella di aggiungere alcune declinazioni diatopiche di saluto informale, come ad esempio il sardo *Aiò* (Bixby), o il romanesco *Aoò* (Bixby).
Tuttavia la soluzione migliore per l'utente resta la completa personalizzazione della parola di risveglio.

2.2.1 Problemi e soluzioni del Wake-Up

Utilizzare un comando di risveglio è fondamentale per l'usabilità dell'assistente nei tipici scenari d'impiego, ma ciò implica almeno due conseguenze:

1. Il dispositivo su cui è installata la piattaforma deve perennemente essere in ascolto, con conseguente dispendio di risorse.
2. Il dispositivo deve riconoscere efficacemente la parola di risveglio, per evitare attivazioni mancate o accidentali.

Per ovviare al primo problema si ricorre a varie strategie.
Anzitutto occorre comprendere che ogni dispositivo che ospita la piattaforma è in grado di gestire il flusso audio in entrata da uno o più microfoni, convertendolo in segnale digitale facilmente analizzabile ed immagazzinabile.
La conversione dell'audio in ingresso da analogico a digitale avviene tramite un circuito elettronico chiamato *Convertitore Analogico-Digitale* (ADC), che percepisce le vibrazioni sonore captate tramite microfoni a elettrete e le converte in funzioni matematiche, sotto forma di segnali digitali.
I bit vengono analizzati da un processore che compara in tempo reale la struttura dell'audio catturato con la struttura della parola di risveglio che l'utente ha precedentemente registrato durante la configurazione dell'assistente.
Se il segnale digitale in ingresso somiglia, entro una certa tolleranza, al segnale digitale della parola di risveglio registrata dall'utente, allora avviene un *match*.
Il dispendio di risorse è dovuto sia all'ascolto continuo, sia alle operazioni svolte per riconoscere un match; tuttavia una soluzione adottata per limitare entrambi i costi energetici è l'integrazione lato hardware e software dei componenti necessari a svolgere tali operazioni, tramite l'adozione di circuiti e memorie predisposti allo scopo all'interno del chip, massimizzando i tempi e la resa energetica dei processi coinvolti nel riconoscimento della parola di risveglio.

Per ovviare al secondo problema si adottano due accorgimenti:

1. Si estrae la voce ripulita dell'utente dal segnale audio grezzo in ingresso eliminando rumori ambientali e di fondo tramite l'ausilio di microfoni supplementari oltre a quello principale, situati in posizioni differenti e volti alla cattura spaziale dei suoni in ingresso.
Questa accortezza comporta vantaggi dal punto di vista dell'affidabilità del riconoscimento sia della parola di risveglio che dei successivi comandi vocali, i quali non saranno sporcati da rumore indesiderato.

2. Si confronta la voce ripulita con la firma sonora della voce utente. La firma sonora non è altro che un insieme di caratteristiche intrinseche ed univoche della voce dell'utente a cui appartiene il dispositivo, tra cui timbro e frequenze. La firma sonora viene costantemente migliorata ed aggiornata con l'utilizzo dell'assistente, divenendo così immune ai cambiamenti non repentini che la voce subisce nel naturale corso della vita.

Una volta riconosciute parola di risveglio e voce dell'utente, l'assistente entrerà in modalità registrazione, durante la quale immagazzinerà in una memoria rapida predisposta allo scopo (*buffer*) l'audio in ingresso ripulito dal rumore sotto forma di segnale onda.
Il segnale onda verrà poi elaborato in tempo reale dal sistema di Automatic Speech Recognition (ASR).

2.3 Componenti e funzionamento dell'ASR

Il sistema di ASR di Bixby ha quattro componenti principali:

1. Elaboratore del Segnale ed Estrattore delle Caratteristiche
2. Modello Acustico (Acustic Model, AM)
3. Modello del Linguaggio (Language Model, LM)
4. Motore di Ricerca Ipotesi

La componente che **elabora il segnale ed estrae le caratteristiche** prende in input il segnale audio, migliora il contenuto rimuovendo rumori e distorsioni, converte il dominio della funzione del segnale da tempo a frequenza ed estrae le caratteristiche salienti della voce rendendole fruibili dal modello acustico.

Il **modello acustico** integra modelli di acustica e fonetica; prende come input le caratteristiche generate dal componente di estrazione e genera un punteggio relativo alla sequenza di caratteristiche.

Il **modello del linguaggio** stima la probabilità (punteggio LM) di una sequenza di parole ipotizzata imparando la correlazione tra le parole nei corpora di addestramento. Serve anche a disambiguare i casi di omofonia (miglio: cereale/misura), e a ridurre lo spazio di ricerca per un dato suono ricercando soltanto tra le sequenze presenti nei corpora di addestramento corrispondenti alla parola in esame.

Il componente di **ricerca ipotesi (RI)** combina i punteggi del modello acustico e del modello del linguaggio in base alla sequenza del vettore delle caratteristiche e alla sequenza di parole ipotizzata e restituisce la sequenza di parole con il punteggio più alto come risultato del riconoscimento.

Il motore di RI in un sistema ASR può seguire due approcci: uno statistico ed uno a regole. Nel caso di Bixby si utilizza l'approccio statistico quasi sempre, salvo che in alcuni ambiti ristretti che saranno spiegati più avanti (v. 2.3.2).

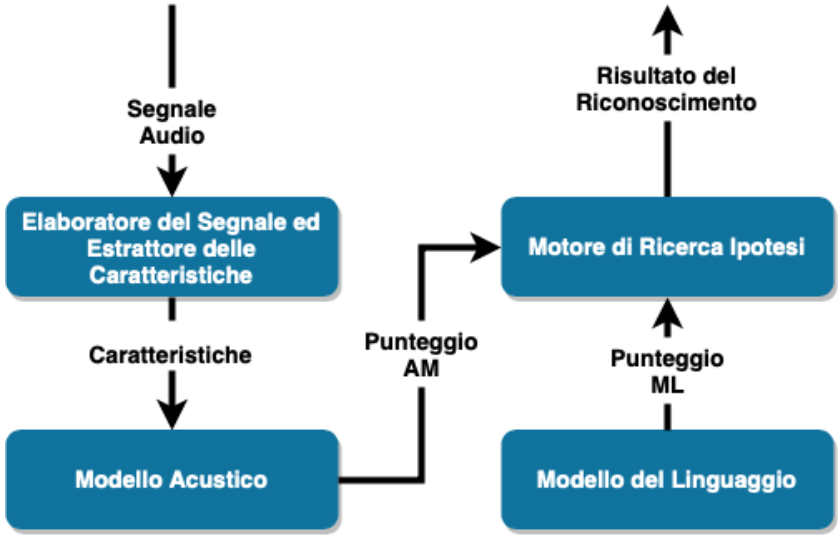

Figura 2 - Architettura dell'ASR

2.3.1 Approccio statistico

L'approccio statistico è possibile grazie alla raccolta di grandi quantità di dati, preferibilmente estrapolati dallo stesso contesto nel quale dovrà agire l'ASR.
Samsung parte in vantaggio, avendo avuto a disposizione un'enorme quantità di registrazioni vocali fornite nel tempo dagli utilizzatori di S-Voice, un servizio di assistenza vocale precedente a Bixby che l'azienda installava nei suoi smartphone; è possibile considerare S-Voice come l'antenato di Bixby.
I dati, sotto forma di corpora e segmenti audio, verranno analizzati dal sistema di inferenza, nel caso di Bixby una particolare rete neurale appartenente alla categoria delle *Random Decision Forests* (Tin Kam 1995).
Il risultato dell'analisi dei corpora costituirà il Modello del Linguaggio, mentre il risultato dell'analisi dei segmenti audio costituirà il Modello Acustico, ovvero le combinazioni di suoni più probabili e significativi della lingua.
È bene tenere presente che la rete neurale viene costantemente rifornita di nuovi dati, quelli provenienti dalle interazioni utente-assistente, che verranno poi utilizzati per un nuovo ciclo di apprendimento: un miglioramento costante sia del

Modello Acustico che del Modello del Linguaggio che avviene in maniera ciclica, grazie ad un sistema di apprendimento per rinforzo.
Una parte del lavoro svolto nel tirocinio consisteva nella registrazione vocale e revisione della trascrizione automatica di particolari frasi che si prevede verranno pronunciate dall'utente, come ad esempio "Com'è il tempo a Roma?", "Quando atterrerà il volo Alitalia AB7864?", "Quali partite di Serie A ci sono domenica?", e così via.

2.3.2 Approccio a regole

Nell'approccio a regole si utilizzano dei lessici creati a mano, ovvero delle liste di parole a cui sono associate una o più rappresentazioni fonetiche.
Gli elementi dei lessici verranno combinati secondo regole stabilite all'interno di grammatiche.
Le rappresentazioni fonetiche descrivono il modo in cui la parola può essere pronunciata; sappiamo che la pronuncia varia in base alla provenienza geografica del parlante, alla sua istruzione, al contesto, in base a colui a cui è rivolta, e via dicendo.
Associando più rappresentazioni fonetiche ad un elemento si garantisce la riconoscibilità anche nel caso in cui venga pronunciato diversamente dal normale.
Le rappresentazioni fonetiche utilizzano solitamente un alfabeto fonetico standardizzato, come l'*Alfabeto Fonetico Internazionale* (AFI) o l'*Alfabeto Fonetico Americanista* (APA). Tuttavia, per ragioni legate a flessibilità e riutilizzo nella piattaforma, le rappresentazioni fonetiche in Bixby utilizzano un alfabeto fonetico proprietario ideato da Samsung, chiamato SDARPA.
L'approccio manuale viene utilizzato in contesti particolari con limiti ben definiti, come ad esempio per calcoli numerici in cui è necessario distinguere operatori e numeri da preposizioni e verbi ("x"/"per", "6"/"sei"), o nel caso in cui bisogna riconoscere la pronuncia di un personaggio famoso straniero ("Stephen Hawking", "Sigmund Freud", …).
I vantaggi di questo approccio sono l'efficacia del riconoscimento nei ristretti ambiti coperti dalle regole, ed il controllo che il linguista può esercitare nell'imporre le regole di trascrizione.

2.4 Creazione di risorse linguistiche

Uno dei compiti svolti durante il tirocinio ha riguardato la creazione di risorse linguistiche per vari scopi.
Alcune di queste impattano direttamente sulla qualità del riconoscimento vocale, altre rendono più efficiente la ricerca delle ipotesi di riconoscimento. Di seguito esporrò brevemente parte del lavoro effettuato per ASR inerente alla creazione delle risorse linguistiche.

2.4.1 Grammatiche generative

Come visto ogni Capsula dona a Bixby la capacità di effettuare operazioni e fornire risposte relative all'ambito della capsula.
Se l'utente chiedesse di cercare un fioraio nelle vicinanze sul servizio Pagine Gialle, la capsula Pagine Gialle ed i relativi strumenti verrebbero utilizzati allo scopo.
Per ottimizzare il riconoscimento vocale all'interno di una capsula, vengono creati tramite Grammatiche Generative dei corpora di riferimento contenenti prototipi di richiesta.
Ad esempio, la Capsula di *PrezziBenzina* contiene mille possibili richieste che ho creato con l'ausilio di Grammatiche Generative:

p_benzina1.expand

```
p_benzina1  la benzina costa meno a roma o a parigi o o o o o b-_CITY o o b-_CITY
p_benzina1  localizzami le benzine esso o o o b-_BRAND
p_benzina1  il gpl viene meno a los angeles o a parigi  o o o o o b-_CITY i-_CITY o o b-_CITY
p_benzina1  vediamo le stazioni esso  o o o b-_BRAND
p_benzina1  il gasolio  o o
p_benzina1  trovami la benzina  o o o
p_benzina1  il gpl new york o o b-_CITY i-_CITY
p_benzina1  il metano viene più a parigi o a los angeles  o o o o o b-_CITY o o b-_CITY i-_CITY
p_benzina1  il metano o o
p_benzina1  localizza il metano o o o
p_benzina1  vediamo la stazione o o o
p_benzina1  cerca la benzina a roma o o o o b-_CITY
p_benzina1  la benzina viene meno a londra o a roma o o o o o b-_CITY o o b-_CITY
p_benzina1  il gasolio viene più a parigi o a los angeles o o o o o b-_CITY o o b-_CITY i-_CITY
p_benzina1  mostrami le stazioni tesla  o o o b-_BRAND
p_benzina1  la benzina viene più a milano o a londra  o o o o o b-_CITY o o b-_CITY
p_benzina1  localizza il gasolio  o o o
p_benzina1  fammi vedere la benzina esso  o o o o b-_BRAND
p_benzina1  fammi vedere la benzina a los angeles o o o o o b-_CITY i-_CITY
p_benzina1  cercami il metano a parigi  o o o o b-_CITY
p_benzina1  il metano roma  o o b-_CITY
p_benzina1  cercami i distributori  o o o
p_benzina1  il gpl viene più a los angeles o a parigi o o o o o b-_CITY i-_CITY o o b-_CITY
p_benzina1  trovami il metano o o o
p_benzina1  localizza il gasolio a londra o o o o b-_CITY
p_benzina1  il gpl viene meno a parigi o a milano o o o o o b-_CITY o o b-_CITY
```

Figura 3 - Esempi creati con Grammatiche Generative

L'utilizzo di dizionari per generare delle variazioni, sostituendo ad esempio i nomi delle compagnie di carburanti, facilita il compito del programmatore.

2.4.2 Web Crawling per espansione corpora

Il modello acustico, il modello del linguaggio ed il componente di ricerca ipotesi operano su vari corpora di addestramento.
È compito del linguista espandere ed affinare i corpora per migliorare il riconoscimento del sistema di ASR. La fonte ideale per ottenere dati di questo tipo è ovviamente il web.
Tramite degli script in Bash e delle librerie Python (*Beautiful Soup* ed altre) si raccolgono e ripuliscono grandi quantità di testo dalla rete, che vengono poi integrate nei corpora di addestramento.
Solitamente il *web crawling* è mirato: per l'ambito tecnologico si guarda a forum tecnologici, per la cucina andranno esaminati blog gastronomici, e così via.

2.4.3 Registrazioni vocali in camera silente

Alcune frasi pronunciate dall'utente devono necessariamente essere riconosciute da Bixby per non minare l'esperienza complessiva.
In questo ambito ricadono le richieste che interagiscono direttamente con funzionalità basilari dello smartphone, come ad esempio «Attiva la torcia», «Chiudi tutte le applicazioni», «Spegni lo schermo», eccetera.
Per irrobustire e migliorare il riconoscimento il team di sviluppo si è occupato di registrare tali frasi in camera anecoica, avendo l'accortezza di farle pronunciare ad un eterogeneo gruppo di persone; l'insieme delle registrazioni costituisce un *golden master*, un paragone di riferimento attraverso il quale è possibile raggiungere un indice di infallibilità vicino alla perfezione per i casi che devono essere riconosciuti ad ogni costo.
Avendo un tangibile accento salentino ho potuto far parte del gruppo di registrazione. Mi sono occupato altresì di classificare i parlanti in base a provenienza e grado di presenza dell'accento, sesso ed età.

2.4.4 Revisione di trascrizioni automatiche

L'apprendimento del sistema passa attraverso l'addestramento delle reti neurali. L'addestramento nel caso di Bixby è di tipo *Reinforcement* (Sutton, Barto 1998), in quanto il sistema migliora le proprie performance attraverso le interazioni con l'ambiente.
Per migliorare il processo è possibile rivedere ed eventualmente correggere manualmente alcuni dei casi più ostici, in modo da utilizzarli come modelli per l'addestramento.
In questi casi ricadono, ad esempio, il riconoscimento di stringhe alfanumeriche (codici di voli, treni, targhe, etc…), numeri di telefono, acronimi e così via. Il mio compito è stato quello di esaminare ed eventualmente correggere tali casistiche, per aumentare l'efficienza del sistema ASR.

2.4.5 Disambiguazione semantica di omofoni

I casi di omofonia sono facilmente fraintendibili dal sistema, che deve interpretare il significato corretto di ciò che analizza basandosi sul contesto.
Per addestrare la rete neurale a riconoscere correttamente alcune parole omofone, quali "è"/"e", "hanno"/"anno", eccetera, sono stato incaricato di creare alcuni corpora contenenti vari esempi complementari dei casi in esame.
I corpora sono stati creati con l'ausilio di *web crawler* e modificati tramite script in BASH con l'ausilio di Regular Expression.

Capitolo 3

Confronto con la concorrenza

3.1 Differenze nell'ASR

Lo scopo del lavoro, come ribadito in apertura, è di valutare l'impatto delle attività svolte durante il tirocinio sulle prestazioni del sistema di riconoscimento; l'obiettivo è verificare se l'assistente virtuale Bixby ha beneficiato delle risorse linguistiche appositamente create per raggiungere una soglia di riconoscimento accettabile e paragonabile alla migliore concorrenza.
In questo capitolo si effettuerà dunque un duplice confronto tra l'ASR di Bixby e quello di Google Assistant, mettendoli alla prova dapprima su una serie di frasi estratte dal corpus CLIPS (sottosezione "letto"), per una valutazione delle performance in contesto generale; successivamente verranno testati entrambi gli ASR su un mini *test set* preparato appositamente ed appartenente al dominio di competenza di Bixby, per evidenziare meglio il ruolo dei miglioramenti descritti nella sezione "Creazione di risorse linguistiche" (v. 2.4).

3.2 Metodologia del confronto

3.2.1 Test set e registrazioni

Affinché valutazione e confronto siano oggettivi, occorre che siano anche verificabili: l'insieme delle enunciazioni deve essere registrata, in modo da fornire

ad entrambi i sistemi di ASR la stessa fonte, oltre che per ripetere eventualmente la medesima prova in un momento successivo con strumenti diversi.
Inoltre, tutte le enunciazioni devono essere corredate di trascrizione ben controllata da utilizzare come riferimento.
Le **prime registrazioni** analizzate appartengono al corpus CLIPS (Corpora e Lessici dell'Italiano Parlato e Scritto), un progetto finanziato dal MIUR tra 1999 e 2004, nato allo scopo di creare risorse utili al trattamento automatico della lingua italiana, sia in forma scritta che orale.
Le registrazioni selezionate, venti frasi per speaker, sono state incise da tre speaker distinti provenienti dalle città di Milano, Roma e Palermo, nati e vissuti nella città in esame o nella provincia più vicina alla città in esame[3], e sono corredate da trascrizioni annotate; trascrizioni poi ripulite per servire da testo di riferimento.
Le **seconde registrazioni**, cinquanta frasi aventi come oggetto tipiche richieste appartenenti al dominio delle funzionalità di Bixby, sono state incise da un solo speaker dal moderato accento pugliese (io stesso) utilizzando un personal computer Macintosh (*Macbook 12 Early 2016*) in ambiente relativamente silenzioso a circa 22-25 dBA. La relativa trascrizione è stata effettuata seguendo i criteri descritti nel portale web di CLIPS[4].
In entrambi i casi il contesto diafasico è la lettura da testo scritto.
I due *test set* di registrazioni verranno riprodotti dai diffusori audio dello stesso personal computer utilizzato per incidere il secondo *test set*, ed analizzati dai sistemi ASR installati nello smartphone *Samsung Galaxy Note9*, posizionato di fronte ai diffusori alla distanza di circa 20cm.
È importante che entrambi i sistemi di ASR operino sullo stesso dispositivo, per eliminare variabili legate all'hardware che capterà l'audio.

3.2.2 Metrica

La qualità della trascrizione prodotta da un sistema di riconoscimento vocale automatico viene tipicamente calcolata tramite un caso particolare della distanza di Levenshtein, il Word Error Rate (WER), indicatore delle parole sbagliate nella trascrizione.
Gli sbagli possono essere di vari tipi:

- Parole sostituite (substitutions, **S**)
- Parole mancanti (deletions, **D**)
- Parole inserite erroneamente (insertions, **I**)

[3] Definizione delle caratteristiche generali del corpus: informatori, località. Alberto Sobrero (2007).
[4] Specifiche per la trascrizione ortografica annotata dei testi raccolti. Renata Savy (2007).

L'indicatore WER sintetizza gli errori in un unico valore percentuale, calcolato in questo modo:

$$WER = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{\mathrm{N}} * 100$$

dove **N** è il numero di parole totali contenute nella trascrizione corretta.
È bene chiarire che questa stima non entra nel merito della causa né della rilevanza dell'errore, costituendo soltanto un riferimento di massima per la valutazione approssimativa di un sistema ASR, e che potrebbe risultare in un valore che eccede il 100% a seconda dei parametri di dividendo e divisore.

3.2.3 Calcolo

Il calcolo del WER potrebbe essere fatto anche manualmente, tuttavia è più sicuro e rapido utilizzare il programma di riferimento SCLITE situato all'interno dello Speech Recognition Scoring Toolkit (SCTK), sviluppato dal National Institute of Standards and Technology (NIST).
Il programma riceve due testi in input: l'ipotesi di trascrizione dell'ASR (denominato HYP) e la trascrizione di riferimento (denominato REF). I formati accettati sono quattro: trn, ctm, stm, txt.
Per questa comparazione verrà utilizzato il formato trn, che separa con a capo (*newline*) ogni enunciazione e tiene traccia del parlante tramite identificativo, trattino basso e numero progressivo situati a fine enunciazione tra parentesi tonde (ad esempio «(a_1)»).
Il testo HYP viene allineato al testo REF tramite l'algoritmo Dynamic Programming (DP), che assegna un peso-valore ad ogni parola corretta, mancante, inserita o sostituita, rispettivamente 0, 3, 3 e 4.[5]
L'output del programma è volutamente composto da due documenti: il primo contiene una tabella riepilogativa del processo di allineamento, il secondo è strutturale e contiene le coppie di stringhe confrontate e la posizione degli errori, segnalati con le lettere S (substitution), D (deletion), I (insertion) sotto l'occorrenza.

[5] Oltre al Dynamic Programming, la scelta predefinita, è possibile utilizzare anche "diff", un algoritmo appartenente ai sistemi operativi GNU.

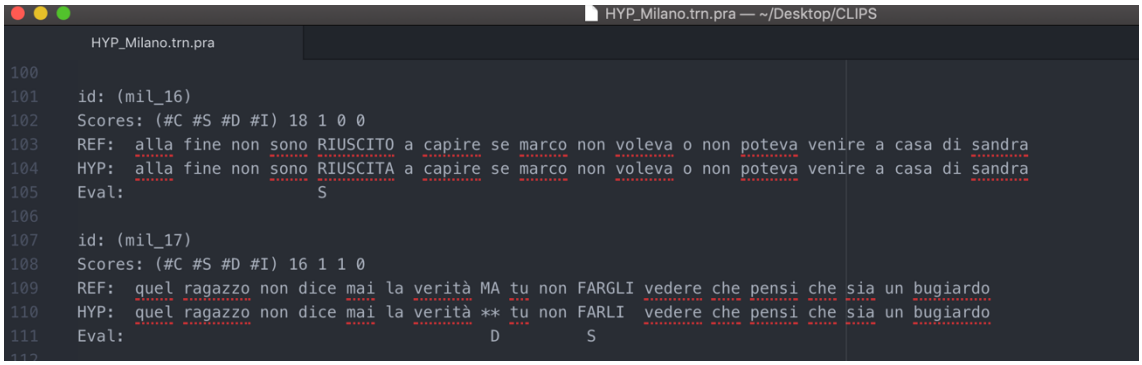

Figura 4 - Output strutturale di Sclite

Nel documento con tabella, la colonna denominata "Err" contiene il valore del WER, mentre la colonna "S.Err" contiene il *Sentence Error Rate* (SER), che indica il numero di enunciati non interamente riconosciuti (v. Figura 5).

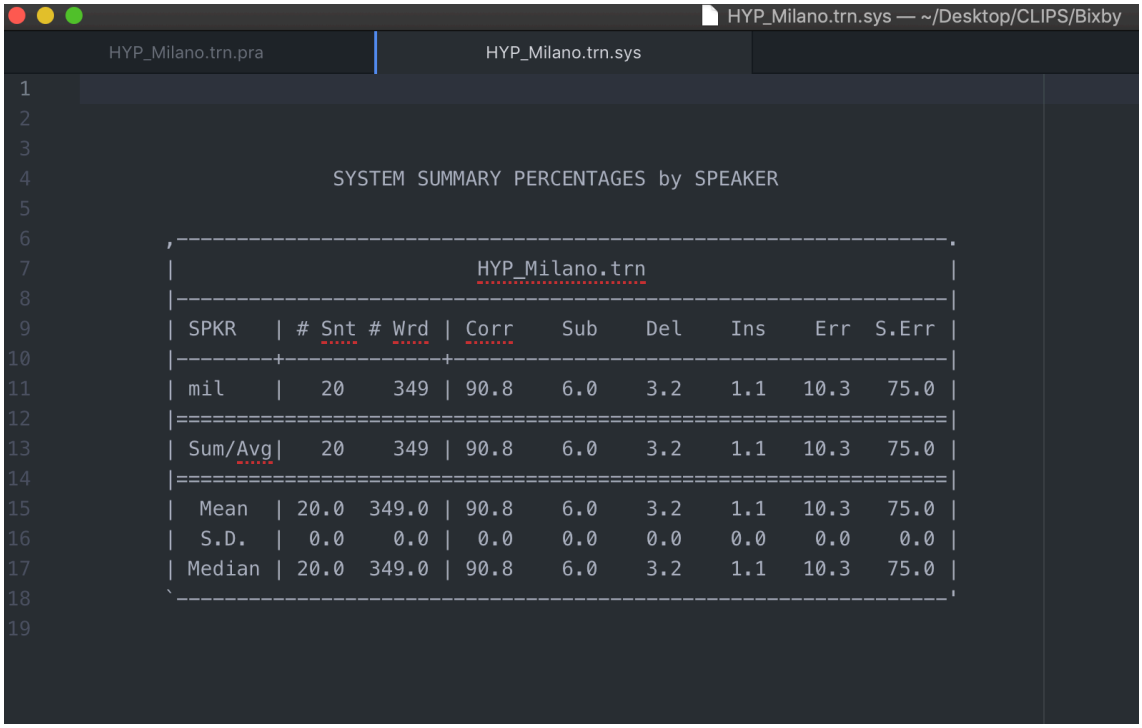

Figura 5 - Output tabellare di Sclite

3.3 Confronto mediante Sclite

3.3.1 Enunciati dal corpus CLIPS

Analizziamo le prestazioni di riconoscimento dei due sistemi di ASR sul *test set* di registrazioni appartenenti al corpus CLIPS.
Sebbene i tre gruppi di trascrizioni di riferimento, rispettivamente appartenenti agli speaker Milano, Roma, Palermo, dovrebbero essere tra loro uguali in quanto i parlanti pronunciano le stesse venti frasi, possiamo notare qualche differenza nella trascrizione dovuta ad errori nella lettura, come ad esempio nella frase #8 dello speaker Roma, che contiene un «(…) ti non ti (…)»:

Figura 6 - Trascrizione di riferimento dello speaker Roma, enunciato #8

Oppure il «(…) che che (…)» nella frase #17 sempre di Roma:

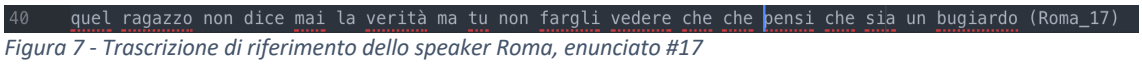

Figura 7 - Trascrizione di riferimento dello speaker Roma, enunciato #17

E ancora in frase #19 dove l'articolo femminile "la" viene ripetuto: «la la»

Figura 8 - Trascrizione di riferimento dello speaker Roma, enunciato #19

Oltre a queste sviste in lettura, mantenute nella trascrizione di riferimento, è bene tenere presente che gli speaker non hanno una dizione particolarmente accurata, mancando a volte di pronunciare parte finale o iniziale della parola, e che l'accento regionale è sempre riscontrabile in maniera abbastanza marcata.

Di seguito possiamo esaminare la tabella riepilogativa delle performance ASR di Google Assistant:

SYSTEM SUMMARY PERCENTAGES by SPEAKER

HYP_Google_CLIPS.trn

SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
mil	20	349	94.6	4.0	1.4	0.3	5.7	40.0
roma	20	353	91.5	5.9	2.5	0.8	9.3	65.0
pal	20	355	94.4	5.1	0.6	0.8	6.5	55.0
Sum/Avg	60	1057	93.5	5.0	1.5	0.7	7.2	53.3
Mean	20.0	352.3	93.5	5.0	1.5	0.7	7.2	53.3
S.D.	0.0	3.1	1.7	1.0	1.0	0.3	1.9	12.6
Median	20.0	353.0	94.4	5.1	1.4	0.8	6.5	55.0

Figura 9 - Tabella riepilogativa performance ASR di Google Assistant, minicorpus CLIPS

Su un totale di 1057 parole in 60 enunciati, Google Assistant riconosce e trascrive correttamente il 93.5% delle parole. Il valore WER medio si attesta sui 7.2 punti percentuali.
Un risultato eccellente viste le premesse, ma atteso: Google ha avviato lo sviluppo del suo assistente in lingua italiana nel 2016 e lo ha ufficializzato nel novembre del 2017[6].

[6] Fonte: Blog.Google.com (01/11/2017)

Vediamo ora la tabella riepilogativa delle performance ASR di Bixby:

SYSTEM SUMMARY PERCENTAGES by SPEAKER

HYP_BIXBY_CLIPS.trn							
SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
mil	20	349	91.4	5.4	3.2	1.1	9.7	75.0
roma	20	353	79.6	13.9	6.5	0.8	21.2	65.0
pal	20	355	91.0	7.3	1.7	0.8	9.9	70.0
Sum/Avg	60	1057	87.3	8.9	3.8	0.9	13.6	70.0
Mean	20.0	352.3	87.3	8.9	3.8	0.9	13.6	70.0
S.D.	0.0	3.1	6.7	4.4	2.5	0.2	6.6	5.0
Median	20.0	353.0	91.0	7.3	3.2	0.8	9.9	70.0

Figura 10 - Tabella riepilogativa performance ASR di Samsung Bixby, minicorpus CLIPS

Delle 1057 parole in 60 enunciati, 87.3% sono state correttamente riconosciute e trascritte. Il valore WER medio si attesta sui 13.6 punti percentuali.
Il 30% degli enunciati è stato completamente riconosciuto.
Considerando che ad un assistente vocale che opera negli ambiti descritti nel capitolo 3.1 basterebbere riconoscere solo alcune parole chiave per interpretare correttamente la richiesta dell'utente, siamo ben oltre la soglia dell'usabilità.

Incrociando i dati ottenuti possiamo notare come il valore WER calcolato per lo speaker Roma corrisponda in entrambi i sistemi a circa il doppio rispetto a quello degli altri speaker.
Dato che il valore è proporzionale e concorde in tutte e due le tabelle, possiamo ipotizzare che la qualità della pronuncia dello speaker Roma è inferiore a quella dei colleghi. Escludo invece un coinvolgimento della qualità di registrazione in sé, in quanto è stata premura del gruppo di lavoro CLIPS verificare la qualità delle

incisioni ed adottare uno standard di registrazione comune a tutti i contenuti raccolti.
Questa ipotesi non fa che avvalorare la bontà di entrambi i sistemi di riconoscimento che, in presenza di una pronuncia più chiara dovrebbero comportarsi ancora meglio.

Passando ad analizzare i singoli errori, notiamo che il tipo di errore sostitutivo (colonna "Sub" nelle tabelle) è quello più presente, seguito dal tipo omissivo ("Del", circa un terzo in meno di "Sub") ed infine dal poco frequente inserimento (colonna "Ins").
Andando ad investigare nel file di output strutturale, possiamo catalogare gran parte degli errori in tre grandi gruppi:

1. Unità omofone
2. Agglutinazioni dovute perlopiù al principio di economia linguistica
3. Troncamenti

Un esempio del primo tipo è riscontrabile nell'enunciato #20 dello speaker palermitano:

```
gli pareva sempre troppo alto * EPPURE lavorava in un'azienda
gli pareva sempre troppo alto E PURE   lavorava in un'azienda
                             I S
```

Figura 11 - esempio di omofonia nell'enunciato #20 di Palermo.

Entrambi i sistemi di riconoscimento scambiano "eppure" con "e pure"; il motivo molto probabilmente risiede nel riconoscimento del raddoppiamento fonosintattico della consonante "P", che nella pronuncia palermitana non è pronunciato con intensità.

Un esempio del secondo tipo di errore invece, è ravvisabile nell'enunciato #19 di Milano, riconosciuto dal sistema di Google:

```
id: (mil_19)
Scores: (#C #S #D #I) 39 2 1 0
REF:  LUCIO ERA    CERCO che sarebbe diventato
HYP:  ***** LUCERA CERTO che sarebbe diventato
Eval: D     S      S
```

Figura 12 - Esempio di agglutinazione, speaker Milano, enunciato #19, Google ASR

Dove "Lucio era" diventa "Lucera", come il comune foggiano. In questo caso la causa dell'errore potrebbe risiedere nella pronuncia effettivamente strascicata dello speaker. Tuttavia l'ASR di Bixby riconosce correttamente le intenzioni del parlante, come mai?

Probabilmente dipende dal modello del linguaggio per la predizione adottato (v. 2.3). Il modello del linguaggio utilizzato da Google potrebbe essere stato addestrato con una quantità di dati sovrabbondante, contentente troppi esempi della parola Lucera, allo scopo di riconoscere efficacemente i nomi dei comuni italiani. Questo tipo di problema è chiamato *overfitting*: in statistica e in informatica si parla di *overfitting* (in italiano adattamento eccessivo, sovradattamento) quando un modello statistico molto complesso si adatta ai dati osservati (il campione) perché ha un numero eccessivo di parametri rispetto al numero di osservazioni.

Ovviamente questa è solo una supposizione personale nata dall'esperienza d'osservazione delle meccaniche microscopiche che regolano il funzionamento dei sistemi di riconoscimento, sarei felice di poter confermare o smentire l'ipotesi lavorando a contatto coi linguisti in Google.

Per quanto riguarda la terza tipologia di errori, possiamo constatare un esempio di troncamento nell'enunciato #14 di tutti gli speaker:

```
333     id: (pal_14)
334     Scores: (#C #S #D #I) 14 5 0 0
335     REF:  CHIAMAI il medico perchè avevo male
336     HYP:  CHIAMA  il medico perchè avevo male
337     Eval: S
```

Figura 13 - Esempio di troncamento, enunciato #19 per tutti gli speaker da entrambi i sistemi ASR

Sia l'ASR di Assistant sia quello di Bixby interpretano "chiamai" come "chiama". Non è un problema assimilabile al singolo speaker, in quanto tutte e tre le interpretazioni sono state affette da troncamento.
Causa del mancato riconoscimento potrebbe imputarsi alla debole pronuncia della "i" finale da parte di tutti gli speaker, anche alla luce della successiva "i" dell'articolo "il" che facilita la distinzione di "chiamai" e "il" all'ascoltatore umano.
Inoltre, un ruolo fondamentale potrebbe essere giocato dallo scopo per cui sono stati creati gli assistenti virtuali: per definizione essi devono comprendere ed eseguire dei comandi dettati dall'utente, che impartirà l'ordine quasi certamente utilizzando il tempo verbale imperativo (in questo caso "chiama").
Non un errore prodotto da cattiva interpretazione, bensì da eccessivo zelo nel voler riconoscere a tutti i costi le frasi tipiche che gli utenti pronunceranno. Se fosse questo il caso potremmo ancora parlare di *overfitting* nell'addestramento del modello del linguaggio (v. paragrafo precedente).

3.3.2 Enunciati dal corpus Custom

Il corpus Custom è un insieme di cinquanta enunciati che contengono alcuni dei casi particolari descritti in 2.4. Andiamo a vedere alcune trascrizioni di riferimento degli enunciati registrati:

```
che ore sono a san jose quando qui sono le sedici?
dammi la posizione di alitalia 1667
qual è la destinazione del kl 1421
fammi vedere lo stato dei voli da zurigo ad amsterdam di oggi pomeriggio
usa gmail per mandare un'email a tommaso con il titolo ciao e con messaggio non ci vediamo da un sacco
mostrami la fotocamera nel manuale dell'utente
quali sono i distributori di gpl che costano meno a milano?
aggiungi un nuovo contatto con nome ottavia e numero di telefono 1234567890
condividi il contatto vcf di liberato su whatsapp
dimmi qualcosa sul concerto di ed sheeran di sabato a roma
parlami della rapina avvenuta a tuscolano
il treno 9643 per roma è in orario?
inizia una corsa di 5km su samsung health
dammi la posizione di ryanair 8775
accendi la torcia
svuota la ram
apri le impostazioni dei bordi edge
cerca un paio di pantaloni maison margiela
```

Figura 14 - Alcune trascrizione di riferimento per il corpus Custom

S può immediatamente notare che il contesto è totalmente diverso dal corpus precedente. In questo caso ci troviamo davanti a cinquanta frasi che ricadono nel dominio specifico di competenza di entrambi gli assistenti; sulla carta le performance di Bixby dovrebbero migliorare se rapportate ai risultati avuti nel corpus CLIPS, dato che ho personalmente lavorato durante il tirocinio al riconoscimento di *questa* tipologia di richieste con le attività descritte nel capitolo 2.
Verifichiamo se ciò corrisponde al vero confrontando i risultati del riconoscimento.

Di seguito la tabella riepilogativa delle performance ASR di Google Assistant:

SYSTEM SUMMARY PERCENTAGES by SPEAKER

HYP_GOOGLE_Custom.trn								
SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
col	50	348	97.1	1.7	1.1	0.6	3.4	14.0
Sum/Avg	50	348	97.1	1.7	1.1	0.6	3.4	14.0
Mean	50.0	348.0	97.1	1.7	1.1	0.6	3.4	14.0
S.D.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Median	50.0	348.0	97.1	1.7	1.1	0.6	3.4	14.0

Figura 15 - Tabella riepilogativa prestazioni ASR Assistente Google su corpus Custom

Su 348 parole in 50 enunciati, Google Assistant riconosce e trascrive correttamente il 97.1% delle parole. Il valore WER è di soli 3.4 punti percentuali, un risultato ottimo ed atteso.
La vera sorpresa si ha andando ad analizzare i risultati di Bixby:

SYSTEM SUMMARY PERCENTAGES by SPEAKER

HYP_BIXBY_Custom.trn								
SPKR	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
col	50	348	99.7	0.3	0.0	0.0	0.3	2.0
Sum/Avg	50	348	99.7	0.3	0.0	0.0	0.3	2.0
Mean	50.0	348.0	99.7	0.3	0.0	0.0	0.3	2.0
S.D.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Median	50.0	348.0	99.7	0.3	0.0	0.0	0.3	2.0

Figura 16 - Tabella riepilogativa prestazioni ASR di Bixby, corpus Custom

Delle 348 parole in 50 enunciati, il 99.7% è stata riconosciuta correttamente. Il punteggio WER è di 0.3% punti percentuali. In pratica il sistema di trascrizione automatica ha sbagliato una sola parola:

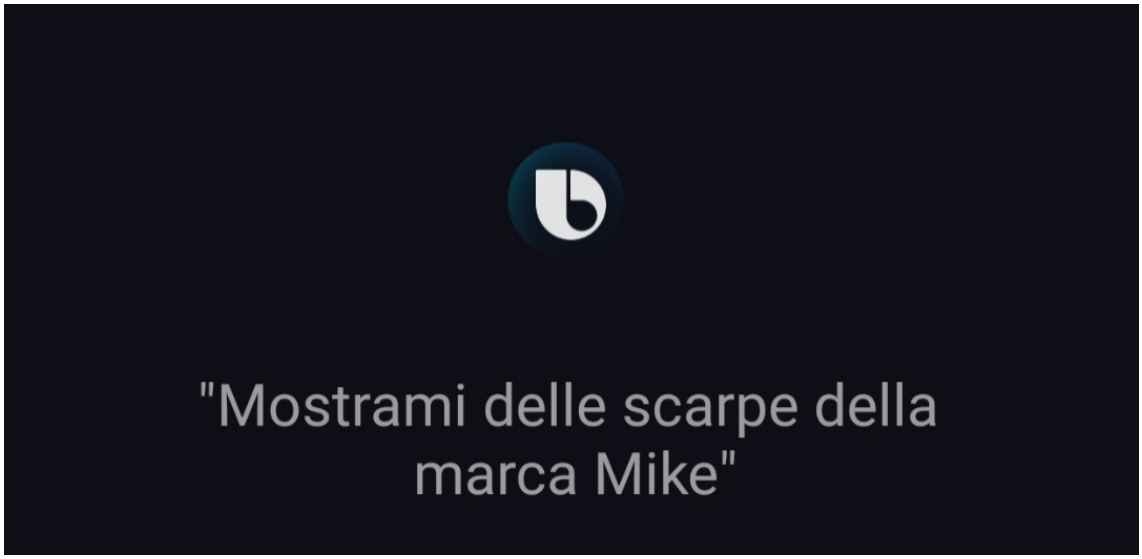

Figura 17 - Samsung Bixby, dettaglio della schermata di trascrizione in tempo reale

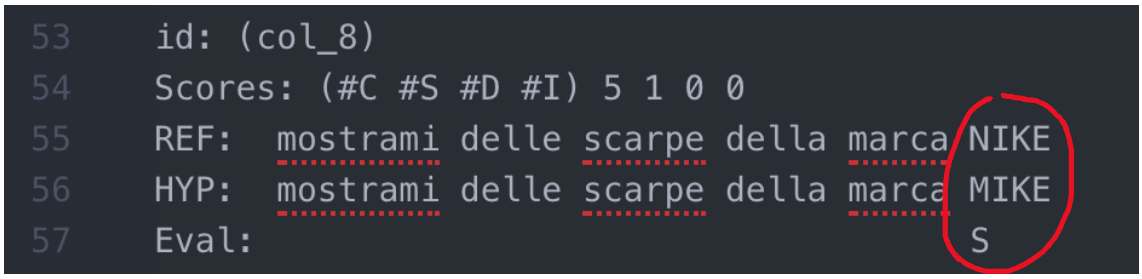

Figura 18 - Errore sostitutivo, ASR di Bixby, corpus Custom, enunciato #8

L'errore riguarda una sostituzione. La nota marca di prodotti sportivi "Nike" è stata trascritta come "Mike". L'errore non si verifica nell'assistente Google, possiamo dedurre da ciò che la registrazione sia abbastanza accurata da poter essere riconosciuta. Anche in questo caso ci sono i presupposti per ipotizzare che a monte dello sbaglio ci sia un problema di *overfitting* nei corpora di addestramento del modello del linguaggio che riguarda i nomi di persona stranieri. È sorprendente anche per me notare come questo sia l'unico enunciato non completamente riconosciuto da Bixby nel corpus Custom.
La mia opinione è che l'addestramento di Bixby si sia focalizzato su tipologie di enunciati, come quelli del corpus Custom, che dovevano *assolutamente* essere riconosciuti, tramite le attività di rinforzo viste in 2.4. Da qui deriverebbe il divario nelle prestazioni tra Bixby e Assistente Google nei due *test set*.

Andando ad analizzare gli errori che ha commesso Assistente Google, si possono scorgere alcuni dei problemi menzionati in 2.4, in particolare:

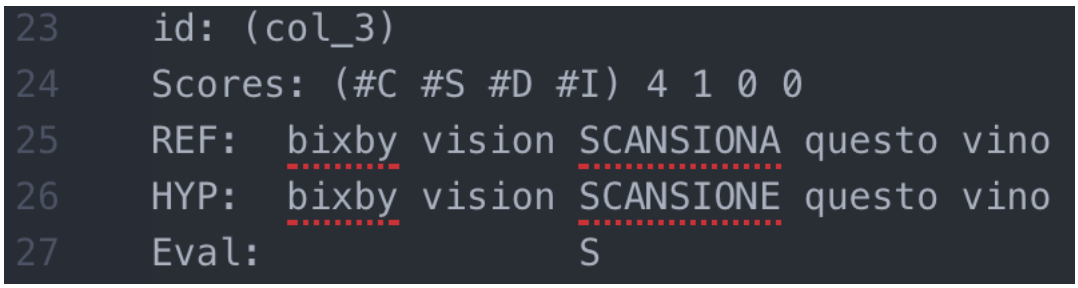

Figura 19 - Errore nel riconoscimento di un servizio Bixby, ASR di Assistente Google, enunciato 3 corpus Custom

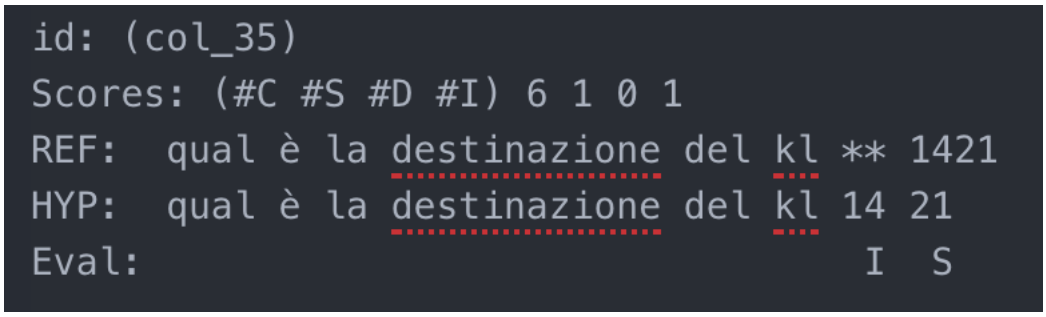

Figura 20 - errore nel riconoscimento di un codice volo, Assistente Google, enunciato #35 corpus Custom

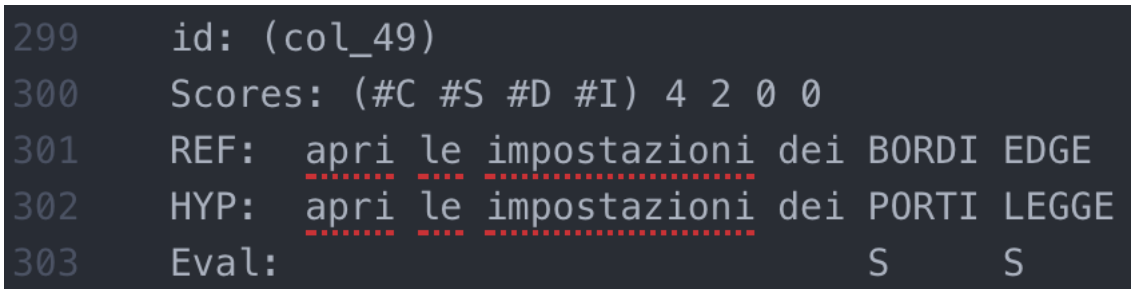

Figura 21 - errore nel riconoscimento di una funzionalità presente in alcuni smartphone Samsung, Assistente Google, enunciato #49 corpus Custom

I pochi errori commessi dal sistema ASR di Assistente Google riguardano ambiti non presi in considerazione dai linguisti durante lo sviluppo dell'ASR.
In figura 19 l'ASR non riconosce correttamente la sequenza di parole che invoca il servizio Bixby Vision, utilizzato per scansionare oggetti e cercarli nel web.
In figura 20, l'errore è dovuto al mancato riconoscimento di un codice volo, un servizio particolarmente pubblicizzato da Bixby che infatti non fallisce nell'individuarlo in qualsiasi forma lo si pronunci, sia che il codice venga scandito cifra per cifra, sia che venga letto come coppia di numeri, sia come singolo numero.
In figura 21, l'errore riguarda la trascrizione del nome di una funzionalità propria di alcuni smartphone Samsung, ed è abbastanza comprensibile che la predizione decreti che "porti legge" sia una combinazione più probabile di "bordi edge".

Conclusioni

Samsung Bixby e Assistente Google contengono dei sistemi ASR di ultima generazione, progettati per strizzare l'occhio alle prestazioni di riconoscimento in ambito generalista.
Al momento i due assistenti vocali si "accontentano" di eccellere nel settore che li riguarda, e tuttavia continuano a migliorare giorno dopo giorno grazie

all'enorme quantità di dati da cui apprendono. Il sistema di Automatic Speech Recognition di Bixby, in particolare, è addestrato a riconoscere in maniera ineccepibile frasi relative a servizi proprietari (*Bixby Vision*, funzionalità *Edge*, gestione della rubrica telefonica etc.) e servizi di partner commerciali (ricette per GialloZafferano, codici voli per FlightStats, stato ferroviario per TrenUp, e così via…), come evidenziato nei test su corpus Custom costruito allo scopo.
Il sistema di ASR creato da Google invece, offre prestazioni migliori nei test sul corpus selezionato da CLIPS, esibendo una affidabilità generale migliore, frutto di vantaggio nelle tempistiche di sviluppo e nelle risorse impiegate per la localizzazione in italiano.

Questo progetto si propone di dimostrare come l'attività svolta durante il tirocinio in Harman-Samsung abbia permesso di migliorare il sistema di trascrizione automatica.
Nella prima parte è stato descritto il contesto in cui tale sistema si colloca, presentando l'obiettivo per cui nasce Bixby ed il panorama attuale degli assistenti virtuali.
Nella seconda parte è stata delineata la *pipeline* di un sistema di ASR, andando ad evidenziare nel particolare alcune scelte progettuali del motore di riconoscimento vocale e descrivendo le attività di sviluppo che mi sono state assegnate.
Nell'ultima parte vengono analizzate le performance del sistema in due scenari opposti: da un lato nell'estratto di letture dal corpus generalista CLIPS, dall'altro nel corpus Custom che rappresenta le richieste tipiche che vengono effettuate all'assistente vocale.
Si è constatato, soprattutto con quest'ultima analisi, come le attività svolte nel tirocinio abbiano ridotto al minimo il Word Error Rate del sistema ASR di Bixby, andando ad appianare alcuni ostacoli che si frappongono tra il trattamento automatico della comunicazione verbale e un'entità complessa come la lingua italiana, raggiungendo così lo scopo prefissato del progetto.

Lo sviluppo del settore è stato accelerato negli ultimi anni dall'uso di tecniche derivanti dalla ricerca nel campo dell'Intelligenza Artificiale, ed in particolare dal settore dell'apprendimento automatico: nell'ambito del motore di ricerca ipotesi, nuove reti neurali reclamano lo scettro di stato dell'arte ogni poche settimane. Nel momento in cui scrivo, è stato pubblicato un articolo sulla piattaforma web Arxiv[7] gestita dalla Cornell University, ad opera del team di ricerca *Google Brain* che descrive come sia possibile migliorare le prestazioni di un sistema di Automatic Speech Recognition andando a trattare il segnale audio come se fosse

[7] https://arxiv.org

un segnale visivo, ed applicando poi la tecnica dell'*Augmentation* per ricavarne più informazione[8].

Nell'ambito degli assistenti vocali in italiano, c'è ancora parecchio margine di miglioramento soprattutto nel trattamento di dialoghi spontanei, caratterizzati da parole pronunciate assieme, parole interrotte, prosodia accentuata, dizionari non chiusi etc.
Possibili migliorie su cui lavorare potrebbero essere l'arricchimento del vocabolario, con relative varianti fonetiche possibili; un miglioramento qualitativo dei dati utilizzati per l'addestramento del modello del linguaggio; nuovi set di registrazioni annotati fedelmente per migliorare il modello acustico. Oltre che l'adozione delle nuove reti neurali più efficienti per la predizione delle ipotesi di riconoscimento.

[8] "SpecAugment" (Park, Chan, Zhang, Chiu, Zoph, Cubuk, Le. 2019)

Bibliografia

[1] CounterpointResearch.com. *Global Smartphone Market Share: by Quarter*, gen. 2019. https://www.counterpointresearch.com/global-smartphone-share (visitato il 4 maggio 2019);

[2] R. Fish, Hu, Boykin, 2006. *Using audio quality to predict word error rate in an automatic speech recognition system*, The MITRE Corporation, 2006;

[3] T. K. Ho. 1995. *Random Decision Forests*. In "Proceedings of 3rd International Conference on Document Analysis and Recognition" vol. 1, pp. 278-282;

[4] D. Jurafsky, Martin. Speech and Language Processing. *An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Pearson, 2009;

[5] A. Lenci, Montemagni, Pirrelli. *Testo e Computer, Elementi di Linguistica Computazionale,* Carrocci, edizione 2016;

[6] M. Palmerini, Savy. *Gli errori di un sistema di riconoscimento automatico del parlato. Analisi linguistica e primi risultati di una ricerca interdisciplinare.* In "Proceedings of the First Italian Conference on Computational Linguistics CLiC-it Pisa" pp. 281-285, Pisa University Press Vol.1, 2014;

[7] D. S. Park, Chan, Zhang, Chiu, Zoph, Cubuk, Le. *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.* Cornell University ArXiv Press, 2019;

[8] R. Pieraccini. *The Voice in the Machine: Buildings Computers That Understand Speech*, MIT Press, 2013;

[9] R. W. Smith, 1997. *Performance measures for the next generation of spoken natural language dialog systems.* In "Interactive Spoken Dialogue Systems: Bringing Speech and NLP Together in Real Applications" pp. 37-40, Eds. ACL, 1997;

[10] Richard S. Sutton, Barto, Andrew. *Reinforcement Learning: An Introduction*, MIT Press, 1998;

[11] M. Tavosanis. *Lingue e Intelligenza Artificiale*, Carrocci, 2018;