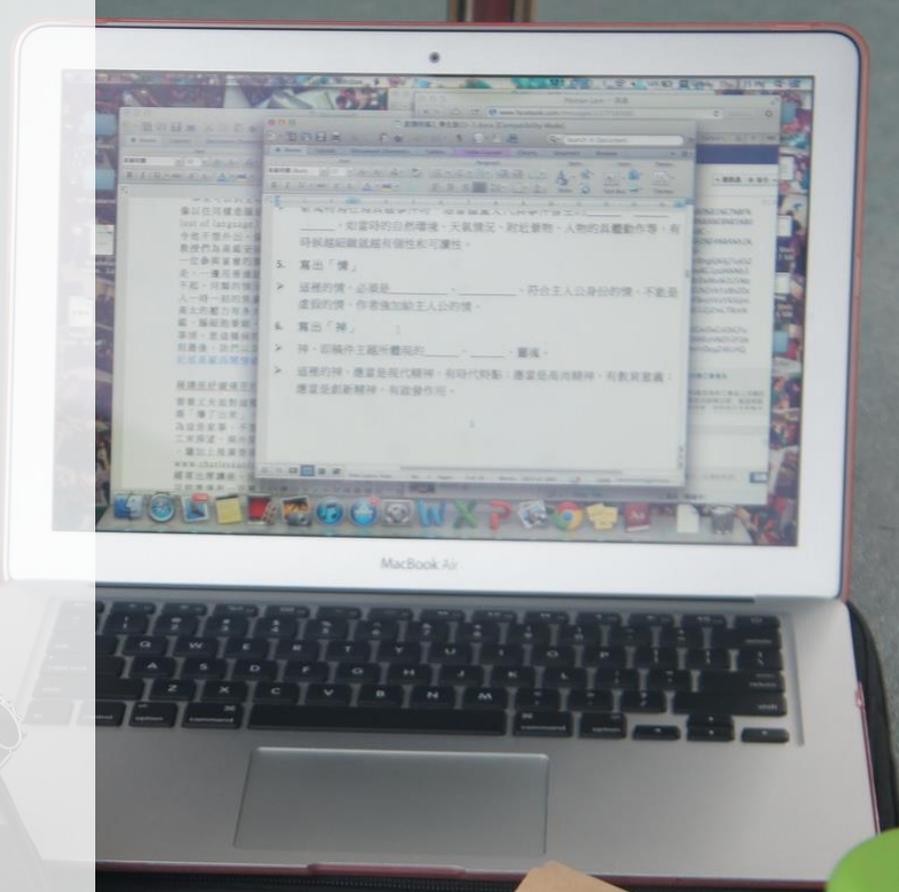


Modulo A

Valutare i sistemi automatici di interazione linguistica

6. Come controllare

20 ottobre 2016



Linguistica italiana II
Mirko Tavosanis
A. a. 2016-2017

Oggi

Come controllare sistematicamente?

Partiamo dalla dettatura, dove occorre:

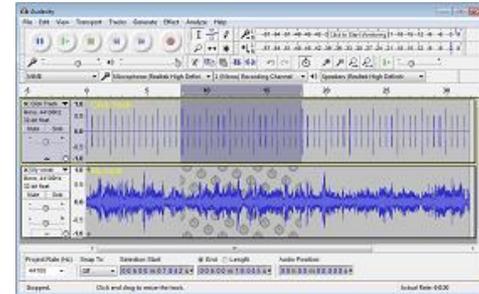
- Registrare le interazioni vocali
- Sottoporle a un sistema
- Controllare il risultato e calcolare la percentuale di errore (WER)

Alla base: verificabilità

- Il testo deve essere registrato, in modo che sia possibile controllare e confrontare:
 - Strumenti diversi (per esempio, Google e Dragon)
 - Lo stesso strumento nel corso del tempo (per esempio, Google in momenti diversi)
- Inoltre, per ogni testo deve essere disponibile una trascrizione ben controllata per permettere il calcolo WER
- Della selezione del testo – cosa fondamentale – parleremo più avanti

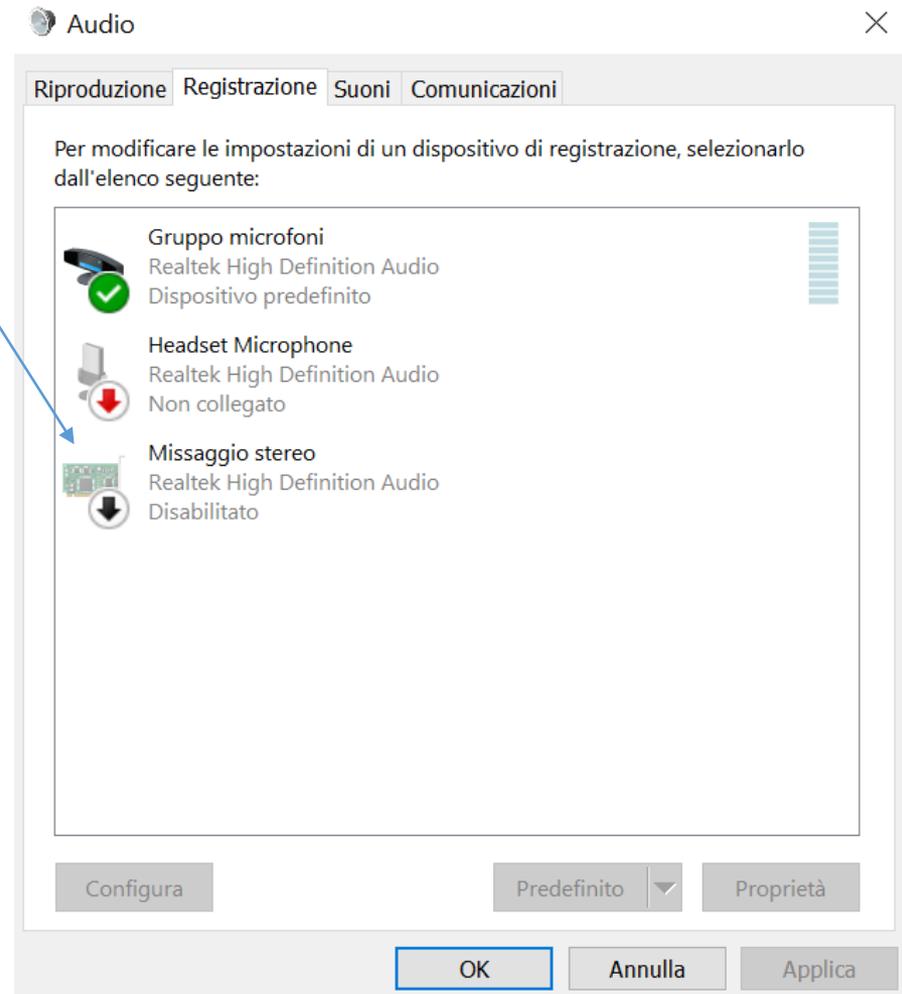
Registrazione

- Primo requisito: un sistema di microfoni che non distorca troppo il parlato (peraltro, vogliamo simulare circostanze reali)
 - Nella mia esperienza, anche i microfoni di un laptop sono adatti
 - Un po' meglio: un microfono dedicato
 - Non occorrono registrazioni in camera silente o simili
- Secondo requisito: un programma per fare la registrazione (anche qui non ci sono troppi problemi)
 - Andrebbe bene perfino il registratore di Windows
 - Vi consiglio il programma Audacity
<http://www.audacityteam.org/>
- La registrazione potrebbe essere fatta in vari formati (l'importante è che possa essere letta da strumenti comuni) ma per uniformità vi chiedo di:
 - Usare il formato .WAV
 - Salvare i file con nomi che permettano di ricostruire facilmente la situazione



Microfoni

- Dipende dal vostro sistema, però su Windows 10 spesso dovrete abilitare il «Missaggio stereo»
- L'importante è che il suono arrivi alla registrazione senza passare da altoparlanti e microfoni esterni (a volte è accettabile anche questo, ma di regola produce una distorsione che rende le registrazioni poco decifrabili)



Esempio pratico

- Registro un breve messaggio: «Buongiorno, vorrei essere avvisato quando il mio invio va in consegna»
- Salvo il file in formato .WAV
- Lo faccio trascrivere da Google e da Dragon, inviandolo direttamente al microfono
- Salvo poi i risultati – e qui le cose si fanno un po' più complicate; comunque va bene intanto un output in formato «solo testo»

Valutazione

- Un indicatore comune è la percentuale di parole sbagliate nella trascrizione – Word Error Rate o WER
- Tuttavia, gli sbagli possono essere di vari tipi:
 - Parole omesse
 - Parole inserite (quando non ci sono nell'originale)
 - Parole sostituite
- L'indicatore WER sintetizza gli errori in un unico valore numerico, calcolato in questo modo:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of the corrects,
- N is the number of words in the reference ($N=S+D+C$)

V. voce su Wikipedia
in lingua inglese:

https://en.wikipedia.org/wiki/Word_error_rate

Perché è complesso?

- Il WER è un caso particolare di misurazione della distanza di Levenshtein, che misura la differenza tra due stringhe
- Gli algoritmi per il calcolo sono piuttosto complessi
- Pensiamo alla sostituzione: che cosa viene effettivamente rimpiazzato?
- Facile:
 - A. Ho visto il **cane**
 - B. Ho visto il **pane**
- Difficile:
 - A. Ho visto che l'algoritmo a volte produce risultati poco chiari
 - B. Ho visto che gli a volte algoritmo a volte risultati poco chiari
- Occorre calcolare il numero **minimo** di modifiche necessario a trasformare la frase A nella frase B, usando le parole

Calcolo

- Il calcolo del WER può essere fatto a mano, ovviamente
- Tuttavia, è più sicuro e più rapido farlo con programmi dedicati... e oltre una certa dimensione, questo è l'unico modo pratico
- Il riferimento oggi è il programma SCLITE all'interno dello Speech Recognition Scoring Toolkit (SCTK): un vecchio programma utilizzabile su Linux (e forse con un emulatore su Windows)
<https://www.nist.gov/itl/iad/mig/tools>
- Il programma funziona da riga di comando
- Per funzionare, deve confrontare un file di controllo con l'output del riconoscimento vocale
- Un impedimento è dato dal formato: il testo semplice non va bene, perché spesso le registrazioni devono confrontare il parlante, ecc.
- Occorre quindi fornire un identificativo per ogni enunciazione

Formati dei file

SCLITE accetta input in quattro formati:

- trn
- ctm
- stm
- txt

Salvo sorprese, noi useremo solo il formato trn, semplice e adeguato

In sostanza, basta:

- separare con a capo (newline) ogni enunciazione – tipicamente ogni frase
- Inserire a fondo enunciazione un codice tra parentesi tonde con identificativo del parlante, trattino basso e numero progressivo: «(a_1)»

trn - Definition of a transcript input file

The transcript format is a file of word sequence records separated by newlines. Each record contains a word sequence, followed by the utterance ID enclosed in parenthesis. See the '-i' option for a list of accepted utterance id types.

example.

she had your dark suit in greasy wash water all year (cmh_sa01)

Transcript alternations, described above, can be used in the word sequence by using this BNF format:

ALTERNATE ::= "{" TEXT ALT+ "}"

ALT ::= "/" TEXT

TEXT ::= 1 or more whitespace separated words | "@" | ALTERNATE

The "@" represents a NULL word in the transcript. For scoring purposes, an error is not counted if the "@" is aligned as an insertion.

example

i've { um / uh / @ } as far as i'm concerned

Comandi

Per eseguire il confronto, occorre dare un comando:

```
./sclite -r [nome file originale] -h [nomefile] -i spu_id
```

Per esempio:

```
./sclite -r toner-originale.trn -h toner_elaborato.trn -i  
spu_id
```

<http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/options.htm>

Opzioni

Le opzioni di SCLITE ricadono in quattro categorie:

Input File Options:

- -e, -h, -i, -P, -r, -R

Alignment Options:

- -c, -d, -F, -L -m, -s, -S, -T -w

Output Options:

- -f, -l, -O, -p

Scoring Report Options:

- -C, -n, -o

Le vedremo meglio più avanti, anche in relazione ai vostri lavori

Che cosa ci fa?

L'impostazione di base prevede l'uso dell'algoritmo Utterance ID Matching:
Input reference and hypothesis files in "trn" transcript format can be aligned by either dynamic programming (DP) or GNU's "diff".

L'impostazione normale è con DP:

When alignments are performed via DP, corresponding REF and HYP records with the same utterance id's are located in the REF and HYP files. DP Alignment and scoring are then performed on each pair of records. Only the utterance ID's present in the HYP file are aligned and scored. This means the REF file may contain more utterance records than the HYP.

Dynamic Programming string alignment

The DP string alignment algorithm performs a global minimization of a Levenshtein distance function which weights the cost of correct words, insertions, deletions and substitutions as 0, 3, 3 and 4 respectively. The computational complexity of DP is $O(NN)$.

When evaluating the output of speech recognition systems, the precision of generated statistics is directly correlated to the reference text accuracy. But uttered words can be coarticulated or mumbled to where they have ambiguous transcriptions, (e.i., "what are" or "what're"). In order to more accurately represent ambiguous transcriptions, and not penalize recognition systems, the ARPA community agreed upon a format for specifying alternative reference transcriptions. The convention, when used on the case above, allows the recognition system to output either transcripts, "what are" or "what're", and still be correct.

(...) For a detailed explanation of DP alignment, see *TIME WARPS, STRING EDITS, AND MACROMOLECULES: THE THEORY AND PRACTICE OF SEQUENCE COMPARISON*, by Sankoff and Kruskal, ISBN 0-201-07809-0.

As noted above, DP alignment minimizes a distance function that is applied to word pairs. In addition to the "word" alignments which uses a distance function defined by static weights, the scilite DP alignment module can use two other distance functions. The first, called Time-Mediated alignment and the second called Word-Weight-Mediated alignment.

Lo stato dell'arte

Richard Eckel, post sul blog Microsoft, 13 settembre 2016: *Microsoft researchers achieve speech recognition milestone*

<http://blogs.microsoft.com/next/2016/09/13/microsoft-researchers-achieve-speech-recognition-milestone/>

“Xuedong Huang, the company’s chief speech scientist, reports that in a recent benchmark evaluation against the industry standard Switchboard speech recognition task, Microsoft researchers achieved a word error rate (WER) of **6.3** percent, the lowest in the industry.

In a research paper published Tuesday, the scientists said: “Our best single system achieves an error rate of 6.9% on the NIST 2000 Switchboard set. We believe this is the best performance reported to date for a recognition system not based on system combination. An ensemble of acoustic models advances the state of the art to 6.3% on the Switchboard test data.”

This past weekend, at Interspeech, an international conference on speech communication and technology held in San Francisco, IBM said it has achieved a WER of 6.6 percent. Twenty years ago, the error rate of the best published research system had a WER of greater than 43 percent.

Quasi tutto dipende dal corpus

Figure 1. Historical progress of speech recognition word error rate on more and more difficult tasks.²⁸ The latest system for the switchboard task is marked with the green dot.

