Università Ca' Foscari di Venezia

Linguistica Informatica Mod. 1

Anno Accademico 2010 – 2011

Parole nel computer

Rocco Tripodi

rocco@unive.it

Riassunto delle lezioni precedenti

Di cosa si occupa la LI

Perché viene elaborato il linguaggio

Applicazioni vecchie e nuove della LI

Inquadramento storico culturale della disciplina

Paradigmi di ricerca (bottom-up / top-down)

Analisi quantitativa (approfondimento nelle prossime lezioni)

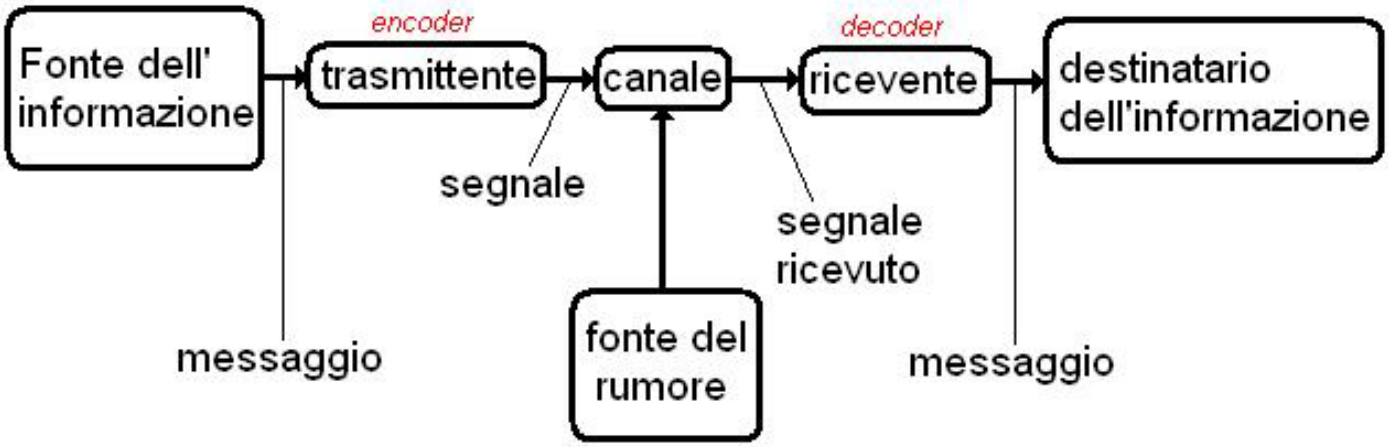
Il corpus (raccolta di testi in formato elettronico uniformemente trattati)

Tipi di corpus

Costruzione e rappresentatività

Teoria dell'informazione (modello di Shannon)

Si occupa esclusivamente dei problemi di trasmissione delle informazioni e ne considera solo gli aspetti tecnici e quantificabili (trasmissione ottimale)



La sorgente o emittente – il dispositivo che produce la variazione

Il ricevente o destinatario – il dispositivo che rileva la variazione

Il messaggio – l'insieme di variazioni emesse dalla sorgente

Il canale – il mezzi fisico su cui viene trasmesso

La codifica (processo di riduzione degli impulsi meccanici in una sequela di segnali sottoposti ad un insieme di regole che ne definiscono le procedure di organizzazione sintattica)

Il rumore – il possibile disturbo che può intervenire sul canale.

Informazione e digitale

L'informazione serve a far diminuire l'incertezza

Viene formulata tramite un codice

- Insieme dei segnali (opposizione reciproca), delle regole di combinazione (grammatica) e corrispondenza tra simbolo e significato

Nei calcolatori viene codificata in bit (*binary digit*)

- Informazione necessaria per scegliere tra due alternative
- **Digitale:** la realtà è descritta tramite un sistema costituito da unità discrete (individuazione degli elementi minimi)
- **Analogico:** gli elementi scelti per rappresentare mutano e cercano si adattano al *continuum* della realtà (termometro,orologio,dipinto)

Il codice binario può descrivere infiniti fenomeni

- Raggruppamento e posizione significativa del bit

Elaborazione

Algoritmo: lista di azioni che devono essere compiute per svolgere un lavoro preciso. Garantire che si arresti dopo un numero finito di passi. Per lavoro finito si intende un processo che non può essere ulteriormente scomposto

Automa (che si muove da sé):

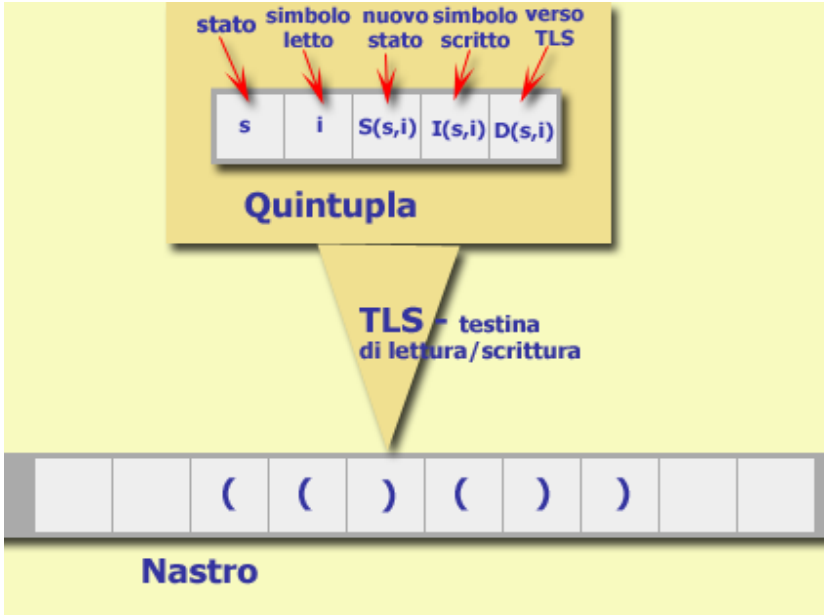
Segue un compito automaticamente
Stato iniziale → Svolgimento del compito → Stato iniziale
Lo svolgimento del compito comporta degli stati e delle regole di transizione. Si passa da uno stato all'altro al verificarsi di determinate condizioni. La lista delle azioni viene raccolta e coordinata dal *programma*.
Si chiama automa a stati finiti una macchina che compie un'operazione in un numero finito di passi.

La quantità di informazione di un messaggio è determinata dalla quantità di eventi possibili presenti nello stato iniziale

Macchina di Turing

Alan Turing (1912 – 1954)

Può un elaboratore replicare l'intelligenza umana? È dotato di un numero finito di condizioni che definiscono la sua configurazione.
La MdT è formata da un nastro di lunghezza potenzialmente infinita, suddiviso in sezioni che la MdT può far scorrere avanti e indietro e da una unità di lettura, scrittura e cancellazione di simboli. Inoltre la M ha una memoria interna che può assumere un numero finito di stati in base ai simboli letti. Ogni algoritmo e rappresentato da una MdT e viceversa.



Testo e computer

Il modello

Per compiere un'analisi tramite computer è necessario tradurre il testo in un linguaggio formale (codifica). È un processo di astrazione che mira ad estrarre la struttura e le caratteristiche dell'oggetto per renderle adatte all'elaborazione e agli scopi dell'analisi. Vengono mantenute le regole dell'isomorfismo cioè la trasformazione che conserva l'informazione, in modo che le due strutture astratte si possano applicare l'una sull'altra. Costruire un modello quindi significa creare una determinata rappresentazione dell'oggetto conservando e sfruttando le caratteristiche che interessano l'analisi.

La codifica del testo (livello zero)

L'elaboratore è in grado di manipolare solo sequenze di codici binari (*byte*). Le lettere dell'alfabeto devono essere rappresentate mediante cifre per poter essere riconosciute ed elaborate

I set di caratteri 1

Consiste nell'associare ad ogni carattere un codice numerico che lo identifica univocamente. Il carattere viene qua considerato come una entità astratta, distinta dalle sue possibili rappresentazioni grafiche (formattazione)

Un set di caratteri consiste in una tabella di associazione tra un repertorio di caratteri e i rispettivi codici numerici. Ogni codice numerico a sua volta è identificato da una sequenza di bit.

Binario	Oct	Dec	Hex	Glifo
1000001	101	65	41	A
1100001	141	97	61	a

Il set di caratteri ASCII usa un byte per rappresentare 128 (2^7) caratteri: 33 di controllo (a capo,tabulazione,ecc) e i restanti 95 servono per rappresentare caratteri e segni di punteggiatura

Codifica di alto livello

I 128 caratteri della codifica ASCII sono pochi per rappresentare diversi alfabeti e diversi segni che possono occorrere nel testo (diacritici, matematici. Per ovviare a questo problema si sono sviluppati set di caratteri che impiegano 8 bit (256 caratteri) compatibili con ASCII ma che in base al sistema operativo possono differire nei punti da 128 a 255

ISO-Latin è la forma più diffusa per la codifica dei sistemi grafici e rappresenta una forma di standardizzazione anche se le varie famiglie di questo set sono mutuamente esclusivi e (caratteri diversi con la stessa cifra)

Unicode è lo standard più recente e permette di rappresentare testi scritti con sistemi grafici differenti. Codifica 96382 caratteri con una rappresentazione esadecimale.

UTF-8 utilizza da 1 a 4 byte. Il primo equivalente ad ASCII, il secondo (128 - 2047) per i caratteri non ideografici, il terzo per i caratteri ideografici, il quarto per repertori ulteriori

Codifica di alto livello

La codifica a livello zero trasforma il testo in un flusso continuo di simboli (*scripto continua*) i segni grafici sono privi di forma.

Per dare forma al testo l'elaboratore leggendo il flusso continuo di simboli deve riuscire a rintracciare le convenzioni tipografiche, testuali e linguistiche. In base agli intenti dell'analisi si possono includere informazioni che riguardano gli apparati testuali e extratestuali (filologia) oppure si potrà decidere di individuare la struttura sintattica (analisi)

Informazione = dati + struttura (modello)

Schema di codifica : insieme di categorie per la codifica (attributo → valore)

```
<TEI.2>
  <teiHeader> [informazioni dell'intestazione TEI] </teiHeader>
  <text> <front> [materiali del peritesto iniziale] </front>
    <body> [testo unitario] </body>
    <back> [materiali del peritesto finale] </back>
  </text>
<TEI.2>
```

Dal carattere alla parola

I computer sono macchine per la manipolazione di simboli. Hanno bisogno di istruzione chiare su come effettuare queste operazioni. Il testo nel computer non è altro che una serie di righe

L'identificazione delle parole (*tokenizzazione*) richiede competenze morfo-sintattiche e semantiche

Il **token** è l'unità base per la LC e comprende nomi, numeri, segni di punteggiatura, sigle, ecc. Sono estratti facendo ricorso alle informazioni presenti nella codifica di livello zero. Il token può quindi essere definito come una qualsiasi sequenza di caratteri delimitata da spazi. Oltre agli spazi vengono definiti ulteriori criteri per delimitare un token. Come i caratteri singoli

Normalizzazione: le varianti ortografiche di una parola (database, semilavorato,numeri,date) devono essere ricondotte ad una forma unica

Tokenizzazione 1

I fase: stabilire quali sono le unità linguistiche atomiche, per definire il tipo di token da estrarre (numeri,parole,nomi propri,ecc)

II fase: individuare la metodologia di individuazione. Token multipli (Gibbs,',s), accorpamento di elementi distinti (La Spezia), normalizzazione

III fase: esprimere i criteri di trasformazione tramite un linguaggio formale

[-+][0-9]+(\.[0-9]+)?([eE][0-9]+)	number
[:alpha:][:alnum:]+	word
[:space:]+	skipped
anything else	single-character

Tokenizzazione 2

Punteggiatura

i segni di punteggiatura devono essere trattati come segni indipendenti anche quando sono attaccati ad una parola. Sono difficilmente gestibili perché possono avere impieghi differenti.

Punto (fine di frase,abbreviazione,cifre,ecc).

Euristica: punto-spazio-maiuscola può trovare fino al 90% di delimitatori di frase ma deve considerare le eccezioni per le abbreviazioni (Sig. Rossi)

Case sensitive: l'elaboratore codifica le maiuscole e le minuscole in modo diverso per cui: casa, Casa e CASA sono tre token differenti che vanno normalizzati.

Euristica: convertire in minuscolo solo le parole all'inizio della frase

Tokenizzazione 3

Acronimi e abbreviazioni

difficili da individuare perché in continua espansione e riportati in maniera differente diversa (con o senza punti).
Euristica: consultazione di elenchi combinata ad euristiche di individuazione (lettere puntate, maiuscoletto)

Token complessi (entità unitarie)

nomi propri: Los Angeles
espressioni multilessicali: fuori servizio, ad hoc
strutture alfanumeriche: espressioni monetarie, unità di misura, targhe
Gli scopi dell'analisi molto spesso guidano il trattamento dei token complessi. In alcuni casi anche la sequenza nome cognome può essere considerata come un token unico

Espressioni regolari 1

La tokenizzazione si occupa della ricerca di stringhe che soddisfano particolari criteri. Le espressioni regolari consentono di definire in maniera formale questi criteri e di individuare tali stringhe (*pattern matching*)
Sono state ideate dal logico matematico Steven Kleene nel 1956 come notazione algebrica per definire insiemi di stringhe di simboli.

Sintassi

Ricercare un carattere: **/carattere/** (*case sensitive*)
Classi di caratteri: **/[zL]/** (tutte le occorrenze di z e L)
Intervallo: **/[a-z]/** (tutti i caratteri dalla a alla z in ASCII)
/[a-d]/ è uguale a /[a,b,c,d]/
Negazione della classe **/[^wg]/** (tutti tranne w e g)
Speciali: preceduti da "\":
/\d/ è uguale a /[0-9]/
/\t/ tabulazione
/\s/ spaziatura, uguale a /[\r\n\t]/

Espressioni regolari 2

Sintassi

Sequenze di caratteri **/\s[a-z]/** è uguale a spaziatura + carattere
Disgiunzione: **/il|la/** tutte le occorrenze di *il* o *la*

Moltiplicatori: operatori che permettono di specificare quante volte un elemento può occorrere
? zero o una volta **/uno?/** unifica con un e uno
* zero o più volte /RA\d*/ RA seguito da zero o più numeri
+ uno o più volte **/\d+/** unifica con 4, 23, 34544, ecc

Ancore: caratteri speciali che indicano la posizione precisa nella riga in cui devo occorrere la stringa
Inizio di riga: ^ **/^Il/**
Fine di riga: $
Confine di token: \b / **\be\b /**
Qualsiasi tranne il confine di token: \B

Espressioni regolari 3

Raggruppamento

- Moltiplicare intere stringhe **/(la)+/**
- Alternative: **/trov(o|iamo)/**

Ordine di precedenza:

- parentesi, moltiplicatori, sequenze e ancore, alternativa

Sostituzione di testo

- s/espressione_regolare/nuova_stringa

Analizzare il linguaggio

- Punto confine di frase **/\b[a-z]+\.\s+[A-Z]/**
- Non tiene conto che il punto potrebbe essere preceduto da altri caratteri oltre che a-z
- Analisi morfologica : raggruppare le radici e le terminazioni
- Estrazione delle frasi nominali: **/Art?\sAgg*\sN\sAgg*/**

Operatori di ricerca

Operatore [+]: costringe il motore di ricerca a includere la parola che precede nella *query* anche se si tratta di una *stop word*. Es: Star Wars +I

Operatore [-]: esclude la parola che precede dalla ricerca

Operatore [~]: cerca i sinonimi della parola che precede

Operatore [OR] o [|]: OR logico

Operatore [..]: range numerico di risultati. Es chitarra € 250..1000

Operatore [*]: unifica con una o più parole

Operatore [site:]: consente di ricercare all'interno di un sito specifico

Operatore [filetype:]: consente di ottenere i risultati nel formato richiesto

Operatore [cache:]: restituisce l'ultima versione indicizzata del sito

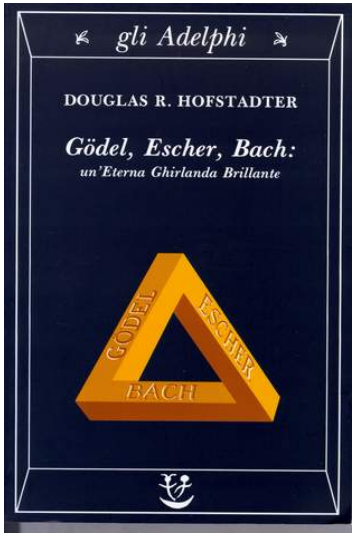
Letture consigliate

Giuseppe Gigliozzi

Introduzione all'uso del computer negli studi letterari

Douglas R. Hofstadter

Godel, Escher, Bach



Progetti di ricerca

Read the Web - Link

Data Journalism – Link

Google Prediction API - Link