Linguistica Computazionale – Lezione 4

Lessico, analisi morfologica e robustezza agli errori

Lunedì 13 Marzo 2007

Cristiano Chesi, chesi@media.unisi.it

Introduzione al NLP: Applicazioni e Problemi

- Indice
  - Lessico ed analisi morfologica
    - Lessico generativo
    - Morfologia a due livelli
    - Analisi morfologica con FSA e FST
    - Alcune applicazioni: lo stemming
  - Normalizzazione dell'input
    - Introduzione alla correzione ortografica
    - Classificazione errori
    - Metodologie di correzione automatica
    - Cellulari e T9

Letture, approfondimenti

- **Bibliografia essenziale**
  - Hutchins & Somers (1992) *Cap. 5*
  - Jurafsky & Martin (2000) *Speech & Language Processing*. Prentice Hall, NJ (Cap. 2)
- **Approfondimenti**
  - Miller & al. (1993) *Introduction to WordNet: An On-line Lexical Database*. ms.

Lessico "generativo"

(inizio ...)

- perché ogni entrata lessicale non può essere registrata **singolarmente**?
  - sarebbe **inefficiente**:

	mangi -	**sogn -**	**corr -**	**puff -**
-are/ere	**mangi-**are	**sogn-**are	**corr-**ere	**puff**-are
-o	**mangi-**o	**sogn-**o	**corr-**o	**puff-**o
-ato	**mangi-**ato	**sogn-**ato	***corr-**ato **(**corso**)**	**puff-**ato

  - in Turco (lingua agglutinante) ci sarebbero circa $600x10^6$ entrate lessicali da considerare. In Finlandese 10^7
  - sarebbe **non informativo**:
    - **nessuna relazione significativa** tra entrate lessicali (l'unica relazione possibile sarebbe l'ordine alfabetico, ma *casa* e *case* hanno intuitivamente una relazione più "intima" rispetto a quella tra *case* e *caso*)
    - non esisterebbe **nessun indizio per processare** in modo "particolare" ad esempio un verbo rispetto ad un nome

Lessico "generativo"

(... continua ...)

- classicamente un lessico computazionale era concepito in funzione del contesto in cui doveva essere usato
  - **lessico mentale** - modelli psicologicamente plausibili di relazioni tra unità minime di significato
  - **modelli computazionali** - usati per creare i database lessicali efficienti.
- regole di efficienza computazionale:
  - la rappresentazione lessicale deve essere **esplicita** ed **indipendente** dalle applicazioni che la utilizzeranno
  - la **struttura globale** delle entrate lessicali è importante almeno quanto la **struttura interna** delle singole parole: la sua organicità e significatività serve a rappresentare una complessa base di conoscenza (**ontologia**)
  - il lessico deve essere in grado di **coprire adeguatamente** il suo **dominio** (approssimativamente 400.000 entrate lessicali di cui 5.000 entrate verbali, 30.000 nominali, 5.000 aggettivali, un migliaio di avverbiali, altrettanti termini logici, 2.000 composti e 300.000 nomi propri + vari termini dominio-specifici)

Lessico "generativo"

(... continua ...)

- regole di efficienza computazionale (...continua):
  - i lessici computazionali devono essere valutabili almeno su tre scale:
    - **copertura** (sia in estensione, che in profondità, a livello di ricchezza dell'informazione)
    - **estensibilità** (deve poter essere formalmente possibile arricchire il vocabolario con termini nuovi)
    - **utilità** (stavolta valutata a livello delle singole applicazioni/elaborazioni)
- da ricordare:
  - la **completezza** non assicura la **correttezza** (psicolinguistica e computazionale)
  - la **plausibilità psicolinguistica** non garantisce l' **efficienza computazionale** e viceversa

Lessico "generativo"

(... continua ...)

Struttura di una **singola entrata lessicale**:

- informazioni **ortografiche/fonetiche** (devono insomma codificare l'input nel modo più adeguato possibile)
- **morfologiche** (tratti inerenti, quali plurale/singolare, massa/contabile, animato/inanimato...)
- **sintattiche** (categoria grammaticale ed eventualmente la sottocategoria)
- **semantiche** (sia a livello di selezione semantica, che di significato ai fini della traduzione ad esempio)

CASA:
<C,A,S,A>
{N, singolare, femminile ...}
{N comune ...}
[house]

Es. di codifica in XML dell'entrata lessicale "casa":

```
<lex class="nome" subclass="comune" num="sg" gen="f" sem="c12">
    casa
</lex>
```


Lessico "generativo"

(... fine)

- Struttura **globale del lessico**
  - correlare la sottocategorizzazione con la **classe semantica** (Levin 93 propone una vasta serie di **classi di alternanza** cercando di dimostrare che certi comportamenti sintattici verbali, quali l'assegnazione di ruoli tematici, sono prevedibili sulla base di certi tratti semantici minimalmente distintivi, quali la modificazione di stato, la causatività, la relazione tra gli elementi in azione ecc.)
  - trarre immediate **inferenze** in base all'organizzazione gerarchica degli items (**part_of**, **member_of**...)

Lessico "generativo" - esempio di struttura globale di un lessico: le reti concettuali o semantiche

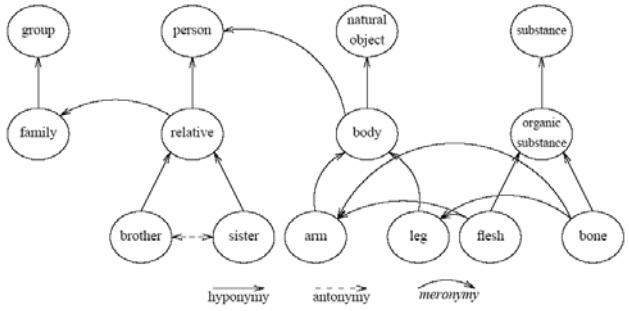
(inizio ...)

- **Wordnet** (Miller 90)
  - interessante esempio di rete semantica (scopo: organizzare il lessico sulla base del **significato** delle parole piuttosto che sulla base della loro **ortografia**) basata sui seguenti principi:
    - certe relazioni semantiche tra **nomi** (gerarchie ad eredità), **verbi** (implicazioni), **aggettivi** e **avverbi** (opposizioni) ma non tra **parole funzionali**, sono psicolinguisticamente significative
    - ogni concetto lessicale (**synset**) può essere **rappresentato dai suoi sinonimi** (altri synset)
    - es. di relazioni:
      - **iponimia** (relazione tra un concetto generale ed uno più specifico; ad esempio "pettirosso" è un iponimo di "uccello")
      - **iperonimia** (relazione inversa all'iponimia)
      - **meronimia** (parte_di)...
    - attraverso l'uso di synset distinti si affronta il problema della **polisemia** (*cane* = animale domestico e *cane* = parte metallica di una pistola saranno due nodi distinti di wordnet anche se si scrivono allo stesso modo)

Lessico "generativo" - esempio di struttura globale di un lessico: le reti concettuali o semantiche

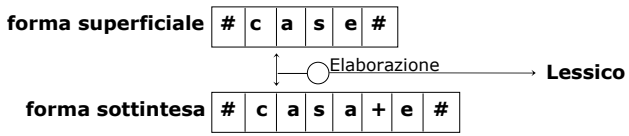
(... fine)

- **Esempio di relazioni semantiche** (Miller 1993)



Analisi morfologica – modello teorico

- **obiettivo**: riconoscere una stringa ben formata di caratteri e metterla in relazione con la struttura dei morfemi che la compongono; questo compito ci permette di introdurre tutti i problemi che si presenteranno nel parsing delle strutture frasali
- **modello teorico:**



Analisi morfologica – FSA

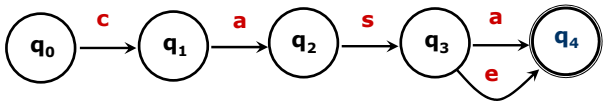
(inizio ...)

- **Finite-State Automata** (**FSA**)
  definiti come quintuple **<Q, Σ, q_0, F, δ>** dove:
  - **Q** = insieme finito e non nullo di stati
  - **Σ** = alfabeto finito e non nullo di caratteri accettabili in input
  - **q_0** = stato iniziale, con $q_0 \in Q$
  - **F** = insieme di stati finali, con $F \in Q$
  - **δ** = insieme delle regole di transizione definite in QxΣ su Q

Analisi morfologica – FSA

(... fine)

- un insieme di FSA non è solo un insieme di macchine che permettono di **riconoscere** o **rifiutare** un elemento lessicale, ma anche di **rappresentare** l'intero lessico.
- FSA che riconosce la parola *casa* ed il suo plurale:



Q = {q₀, q₁, q₂, q₃, q₄},
Σ = {c,a,s,e,#},
Q₀={q₀},
F ={q₄},
δ =

	q0	q1	q2	q3	q4
c	q1				
a		q2		q4	
s			q3		
e				q4	

Analisi morfologica – FSA e two-level morphology

- **Limiti degli FSA**
  per **associare** una **descrizione strutturale** ad un elemento riconosciuto come appartenente al lessico, i semplici FSA non sono più sufficienti (non esiste una memoria esterna, se non la memoria implicita data dallo stato in cui si trova l'automa, in cui "conservare" il percorso e la struttura esaminata).
- Koskenniemi (83) propone un modello di morfologia a due livelli (**two-level morphology**): un **livello lessicale** ed uno **superficiale** che devono essere messi in una qualche relazione significativa dal punto di vista morfologico.
- Tale modello è implementabile utilizzando i **Finite-State Transducers** (**FST**, o **Trasduttori**)

Koskenniemi, K. (1983) *Two-level morphology: A general computational model for word-form recognition and production*. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Analisi morfologica – FST

(inizio ...)

- **Finite-State Transducers** (**FST**, o **Trasduttori**) definiti come quintuple **<Q, Σ, q_0, F, δ>**, dove però sussistono alcune sostanziali differenze rispetto agli FSA:
  - **Σ** = alfabeto finito e non nullo di *caratteri complessi* accettabili in input della forma i:o dove i sono i simboli dell'alfabeto I di input e o simboli dell'alfabeto O di output. $\Sigma \in I \times O$. ε (l'elemento nullo) può essere incluso sia in I che in O
  - **δ** = è definita come (q , i : o) e rappresenta la matrice di transizione che mette in relazione uno stato q di partenza e uno stato q′ di arrivo se la relazione i : o è definita. δ è quindi una relazione da Q x Σ su Q
- i trasduttori hanno funzioni più generali degli SFA: questi ultimi descrivono un linguaggio formale definendo un insieme di stringhe ben formate, gli FST definiscono invece **relazioni** tra insiemi diversi di stringhe.

Analisi morfologica – FST

(... fine)

- In particolare gli FST possono essere usati come **riconoscitori**, **generatori**, **traduttori**, **correlatori tra insiemi**.
- alcune proprietà di cui gli FST godono sono:
  - l'**inversione**, definita come T^{-1}, scambia le etichette di input con quelle di output
  - la **composizione**, se T_1 mappa I_1 su O_1 e T_2 è un trasduttore da I_2 ad O_2, $T_1 \circ T_2$ mappa I_1 in O_2.

Analisi morfologica – esempi di FST

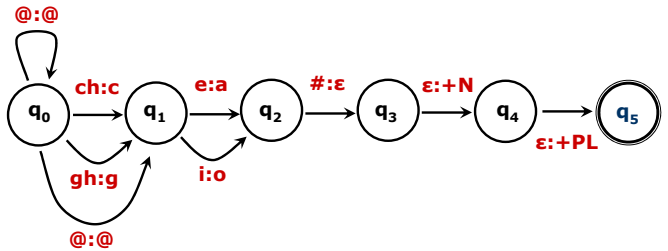
(inizio ...)

- problema di **morfologia flessiva**: definire un FST che descriva il fenomeno dei plurali in italiano.
  - **rappresentazione del problema**
    esempi: casa > case; donna > donne; gatto> gatti; ago > aghi; sacco > sacchi ...
  - **intuizioni e generalizzazioni**
    i nomi femminili prendono il plurale in *e*, i maschili in *i*. *c* e *g* diventano rispettivamente *ch* e *gh* al plurale.
  - **formalizzazione**
    caso regolare: nome maschile > @:@ c|g|@:ch|gh|@ o:i
    nome femminile > @:@ c|g|@:ch|gh|@ a:e
    caso irregolare: uomo > @:@ o:i #:n #:i
  - **implementazione**
    nome femminile > @:@ c|g|@:ch|gh|@ e:a #:ε #:+N #:+PL
    es. case > casa +N +PL (c:c a:a s:s e:a #:ε #:+N #:+PL)

Analisi morfologica – esempi di FST

(... fine)

- **FST** (approssimativo) per descrivere i plurali regolari in italiano:



Analisi morfologica – (in)adeguatezza di FSA e FST

(inizio ...)

- certe lingue mostrano fenomeni più problematici di quelli appena descritti. Tali fenomeni sono detti di **morfologia non-concatenativa**
- **Tagalog** (un dialetto parlato nelle Filippine), **infissi** nel mezzo della parola:
  um (marca l'agente dell'azione) + **hingi** (prestare) = **h-um-ingi**
- **Lingue semitiche**, morfologia a modelli (**templatic morphology**):
  radici verbali composte da consonanti (CCC) **lmd** (apprendere) + flessioni in schemi vocalici (CVCVC) =
  lamad (studiò)
  lumad (fu insegnato)

Analisi morfologica – (in)adeguatezza di FSA e FST

(... fine)

- **Problemi** incontrati:
  - **non-determinismo** (due o più percorsi possono essere innescati dallo stesso carattere allo stato q; transizioni ε)
  - **inadeguatezza** del modello per trattare fenomeni morfologici complessi
  - **ordine** di applicazione degli FSA (o delle regole a seconda dei punti di vista)

Analisi morfologica – Alcune applicazioni

(inizio ...)

- **Ricerca di informazioni**
  (web, archivio digitale strutturato e non)
  - **Keywords** combinate con operatori booleani
    (alberghi & Firenze)
  - **Stemming**
    si cerca di ricavare la radice (*stem*) delle parole da cercare in modo da effettuare ricerche più complete e tolleranti (es. da "alberghi & Firenze" si può generare una query (alberghi AND Firenze) OR (albergo AND Firenze)).

Analisi morfologica – Alcune applicazioni

(... fine)

- L'algoritmo di **Porter Stemming**
  (Porter dal nome del suo ideatore)

  semplice serie di FST a cascata per l'inglese del tipo:
  - ATIONAL -> ATE (es. relational -> relate)
  - ING -> ε (talking -> talk)
- pro e contro:
  - **ipergeneralizzazione** (Krovetz 93)
    es. organization > organ, generalization > generic,
  - **non cattura generalizzazioni** corrette:
    matrices > matrix o European > Europe.
  - vantaggio nell'uso dello stemming solo quando la ricerca è espansiva

Analisi morfologica – Plausibilità psicolinguistica

(inizio ...)

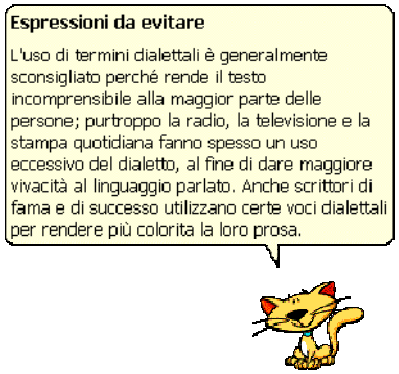
- **Come è strutturato il lessico mentale?**
  - **full listing hypothesis -** *correre*, *corre* e *ha corso*, sono entrate distinte nel lessico mentale (nessuna struttura morfologica interna)
  - **minimum redundancy** - solo i morfemi costituenti sono compresi nel lessico umano; quando si ha accesso ad una parola come *corre* in realtà si ha accesso a due morfemi (*corr-* radice ed *-e* terza persona sing. presente) che poi vengono combinati tra di loro

Analisi morfologica – Plausibilità psicolinguistica

(... fine)

- **Evidenze sulla struttura del lessico mentale**
  - **Effetti di priming** (Stanners ad al. 79)
    flessioni irregolari: *happiness*, *happily* no priming con la radice *happy* Vs. flessioni regolari *pouring* > *pour*
  - **Affinità semantica** (Marslen-Wilson 94)
    government > *govern*
  - **Errori di pronuncia** (Fromkin e Ratner 98)
    easy enoughly*** invece di √*easi**ly** enough*
- Questo sembra suggerire che il lessico mentale debba contenere alcune informazioni sulla struttura morfologica delle parole rappresentate.

La correzione ortografica



La correzione ortografica

- **correzione ortografica** è diversa dal **controllo ortografico**: mentre il controllo può limitarsi semplicemente ad accettare/rifiutare una stringa di testo, la correzione deve proporre una forma corretta in alternativa.

  Esempio di approccio **ingegneristico**:

  1. **definizione** precisa del **problema**
  2. **raccolta dati** rilevanti
  3. **classificazione** degli errori
  4. **ricerca di soluzioni adeguate** ed **efficienti** (relativamente alle classi di errori)

Classificazione errori

(inizio ...)

- all'identificazione degli errori tipici segue solitamente una **categorizzazione** su quattro livelli:
  - **lessicale**
  - **sintattico**
  - **semantico**
  - **pragmatico**
- Va ricordato che ogni errore può essere riconosciuto come tale sia perché è un vero errore (**malformatezza assoluta**), sia perché il sistema non è in grado di trattare, per la limitatezza delle risorse linguistiche utilizzate, la forma che in realtà sarebbe corretta (**malformatezza relativa**)

Classificazione errori

(... continua ...)

- **malformatezze lessicali**
  - **Relative**
    - parole non presenti nel lessico del sistema
  - **Assolute**
    - tipografiche (omissioni, sostituzioni, inserzioni involontarie di lettere)
    - cognitive (errata credenza sull'ortografia della parola)
    - fonetiche (errata credenza sull'ortografia della parola in base alla sua pronuncia)

Classificazione errori

(... continua ...)

- **malformatezze sintattiche**
  - **Relative**
    - inadeguatezza della teoria sintattica implementata (poche regole > ipergeneralizzazione; troppe regole > inconsistenza, esclusione di strutture in realtà corrette)
    - forme colloquiali o dialettali (espressioni idiomatiche, indicativo al posto del congiuntivo…)
    - pronomi di ripresa (pro-sintagmi ripetuti impropriamente)
  - **Assolute**
    - pronome sbagliato (es. me sono andato)
    - mancanza di accordo tra:
      soggetto - verbo (es. Loro è andati...)
      modi - tempi (es. Voglio vado; vorrei andato)
      determinanti - nomi (es. Lo casa)
      aggettivi - nomi (es. Il mare verdi)
    - omissioni di argomenti obbligatori (es. ho messo sul tavolo _)

Classificazione errori

(... continua ...)

- **malformatezze semantiche**
  - **Relative**
    - relazione non presente (relazioni tra gli oggetti non disponibili nella base di conoscenze del sistema)
    - violazione delle restrizioni di selezione (uso di espressioni che violano le restrizioni della base di conoscenze del sistema)
    - sinonimia (mancanza del collegamento semantico tra due sinonimi)
    - polisemia (significati alternativi non presi in considerazione dal lessico di macchina)
  - **Assolute**
    - violazione delle restrizioni di selezione (es. il telescopio nuotò)
    - logica spaziale (es. vieni là)
    - logica temporale (es. domani sono andato a ballare)

Classificazione errori

(... fine)

- **usi figurativi**
  - metafora (es. "con un filo di voce" per "con voce flebile")
  - metonimia (es. "quel ferro vecchio va rottamato" per "quella macchina")
  - sineddoche (es. "il mondo ci è nemico" per "si percepisce una certa ostilità")
  - antonomasia (es. "il divino poeta" per "Dante")
  - perifrasi (es. "quel coso per asciugare i capelli" per "asciugacapelli")
  - eufemismo (es. "passare a miglior vita" per "morire")
  - litote (es. "non è certo un'aquila" per "non è molto intelligente")
  - iperbole (es. "l'ho detto mille volte" per "l'ho già detto molte volte")
  - idioma (es. "il dado è tratto" per "ormai la decisione è stata presa")

Metodologie di correzione automatica

(inizio ...)

- Le varie tecniche che permettono di gestire le malformatezze si basano principalmente su un sistema di **pattern matching** con le forme archiviate nel **lessico** di cui dispone il sistema e su una serie di **euristiche** per decidere le correzioni possibili alle forme errate
  - metodi **simbolici**
    (buona rappresentazione del problema)
  - metodi **subsimbolici**
    (rappresentazione del problema insufficiente)

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Distanza minima**

  Il sistema inventato da Damerau (Damerau 64) e perfezionato da Wagner (Wagner 74) tratta l'errore come una forma che si differenzia da quella corretta per un numero minimo di operazioni di **inserimento**, **cancellazione**, **sostituzione** e **scambio** di caratteri.

  Il metodo consiste nel calcolare attraverso una funzione diversa da sistema a sistema, la minima distanza di correzione tra le stringhe ortograficamente scorrette e le parole presenti nel vocabolario. Se questa distanza è considerata accettabile il vocabolo è considerato come possibile correzione della forma non standard.

  Il grave difetto di questo approccio è l'**inefficienza**: l'elaborazione richiede un numero *n* di confronti, con *n* uguale al numero delle parole del vocabolario.

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Chiave di somiglianza** (algoritmo SOUNDEX, Odell e Russel 1918, correzione di errori fonetici, migliorato ed esteso da Davidson 1962)

  Questa tecnica associa ad ogni stringa una chiave costruita in modo che tutte le parole scritte o pronunciate in un modo simile abbiano una chiave uguale o molto somigliante.

  Confrontando, non le parole, ma solo le chiavi si ottengono le candidate alla correzione della parola scorretta.

  chiave = prima lettera della parola + sequenza di numeri associati secondo certe regole e statistiche di frequenza
  Gli zero e i numeri ripetuti vengono eliminati

  Esempio:

vocali	b, f, p, v	altre consonanti
0	1	2

  casa = c020 > c2; *csa* = c20 > c2

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Chiave di somiglianza** – migliorata
  (Pollock e Zamorra, SPEEDCOP, 84)

  migliorano il metodo della chiave di somiglianza attribuendo due tipi di chiavi ad ogni parola del vocabolario, basandosi sulle seguenti osservazioni riguardo alla distribuzione degli errori:

  1. l'ordine delle vocali è spesso mantenuto invariato
  2. raramente viene sbagliata la prima lettera, statisticamente gli errori si situano verso la fine della parola

  - **skeleton key** = prima lettera della parola + consonanti nell'ordine in cui si presentano nella parola senza ripetizioni + vocali (sempre nell'ordine e sempre senza ripetizioni) (es. gambero = gmbraeo);
  - **omission key** = consonanti, senza ripetizione in un ordine di frequenza (determinato staticamente) e poi dalle vocali, senza ripetizioni, nell'ordine in cui si presentano nella parola.

  Gestiti il 94% degli errori singoli e tra il 74% e l'88% degli errori complessivi presenti nel testo

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Regole**

  La tecnica basata su regole utilizza algoritmi ed euristiche per rappresentare la conoscenza necessaria per determinare quali sono le regole che il termine sbagliato ha violato e le correzioni necessarie per correggerlo (es. restrizioni fonotattiche + informazioni sull'ordine delle lettere sulla tastiera).

  Una volta applicate tutte le regole a disposizione, i risultati vengono presentati all'utente secondo una stima di probabilità.

  Il sistema realizzato da Yannakoudakis e Fawthrop (83) permette una precisione intorno al 76% di errori rilevati. Means (Means 88) affina la tecnica inserendo nel suo correttore oltre alle regole della morfologia inglese altre regole di abbreviazione e flessione non standard migliorando in parte i risultati del primo prototipo.

Metodologie di correzione automatica

(... continua ...)

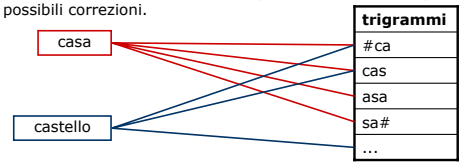
Normalizzazione dell'input

- **N-grammi** (Kohonen 80; DeHer 82; Angell et al. 83; DeSmedt e VanBerkel 88)

  parola = insieme di sottostringhe (n-grammi) che si sovrappongono

  esempio:
  casa = #c + ca + as + sa + a# (bi-grammi)
  strumento = #st str tru rum ume men ent nto to# (tri-grammi)

  vocabolario = tabella di n-grammi indicizzati; ogni indice rinvia ad un determinato termine nel vocabolario di macchina.
  L'insieme dei rinvii determina il campo d'attivazione delle parole e seleziona le possibili correzioni.



Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **N-grammi** - la procedura di correzione degli errori

  1. ogni parola non corretta viene scomposta nei suoi n-grammi
  2. tali n-grammi vengono utilizzati come indici nella tabella per individuare le possibili parole candidate alla correzione
  3. i vocaboli candidati alla correzione saranno tutti quelli che presentano un livello soglia di n-grammi in comune con il termine sbagliato.

  - Un esempio d'implementazione di questo metodo è il programma ACUTE realizzato da Angell e al. (83). Il sistema utilizza una tabella a tri-grammi
  - DeSmedt e VanBerkel (88) propongono una diversa analisi chiamata triphone analysis che permette di correggere errori nel riconoscimento del parlato.
  - Le prestazioni di questo sistema variano a seconda dei vocabolari utilizzati e nessun test standardizzato ha paragonato questo approccio agli altri presentati.

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Analisi probabilistica**
  - utilizzato per migliorare le prestazioni del precedente metodo con n-grammi.
  - due indici che vengono solitamente assegnati alle possibili parole di correzione:
    - **probabilità di transizione** (la probabilità che ha una determinata lettera di seguire una sequenza di caratteri)
    - **probabilità di confusione** (stima della probabilità di sostituzione tra una lettera e l'altra)
  - I primi studi fatti hanno evidenziato come questa sola tecnica non sia sufficiente per ottenere risultati soddisfacenti. Kashyap e Oommen (84) hanno utilizzato questo metodo probabilistico per correggere parole con meno di sei caratteri (svantaggiate dal precedente approccio per n-grammi). Church e Gale (91) propongono con il loro sistema, CORRECT, un approccio ancora più complesso utilizzando quattro matrici di confusione contenenti 44 milioni di parole errate tratte da vari testi.

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Reti neurali**
  - L'applicazione delle reti neurali a questo campo cerca di sfruttare la versatilità che caratterizza questi sistemi per approssimare funzioni euristiche implicite: vista l'intrinseca difficoltà nel definire "regole di violazione", si cerca di far apprendere alla rete ad associare forme errate con forme presenti nel lessico attraverso cicli di addestramento in cui si mostrano "associazioni cognitivamente plausibili".
  - Rumelhart, Burr, Matan (Rumelhart 86; Burr 87; Matan 92) hanno adottato questo approccio in sistemi di correzione che, secondo una stima di Kukich (Kukich 92), possono raggiungere una capacità di correzione che si aggira intorno al 75% dei termini errati.
  - l'efficacia dell'approccio è strettamente dipendente dal tipo di input che si sceglie di dare in pasto alla rete (stringhe di caratteri semplici, n-grammi, sequenze fonetiche…); il problema di una correzione efficiente viene perciò semplicemente spostato, ma non risolto e una riflessione "simbolica" sulla natura del problema sembra sempre comunque fondamentale per il trattamento del problema.

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Espressioni dipendenti dal contesto**
  - Vari autori (Thompson 80, Eastman e McLean 81; Young 91) hanno messo in evidenza che gli errori prodotti, dipendenti dal contesto, sono tra il 25% e il 50% degli errori totali, e di questi circa il 75% è di ordine sintattico.
  - Esistono due principali tipi di approccio:
    - **simbolico** – necessita di un robusto parser e degli analizzatori morfologici e sintattici (richiede una solida teoria linguistica e una efficiente implementazione software)
    - **probabilistico** – utilizza delle tabelle di probabilità per determinare le sequenze di termini consentite (richiede una mole consistente di dati)

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Espressioni dipendenti dal contesto** (Microsoft Word XP)
  - Regole grammaticali:
    - **Punteggiatura** (dopo aver mangiato, decise di lasciare la tavola)
    - **Maiuscole** (le scarpe di paola sono molto costose)
    - **Genere-Numero** (Franco ha comprato dei pantaloni nuove)
    - **Concordanza Soggetto-Verbo** (Il cane e il gatto ha mangiato i resti del pranzo; Io speriamo di vincere un premio. Gli scolari sono uscito alcuni minuti prima del solito)
    - **Frasi** (segnala i più comuni errori relativi alla frase e alla sua costruzione. Esempi di errori rilevati: La donna disse sarebbe andata in città)
    - **Verbi** (segnala gli errori relativi all'uso di un verbo con l'ausiliare sbagliato; L'aereo ha arrivato con parecchi minuti di ritardo sull'orario previsto. Io ho potuto partire per la Francia grazie all'aiuto di mio padre)
    - **Aggettivi** (segnala gli usi impropri degli aggettivi. Esempi di errori rilevati: lavoro molto poco in primavera; corregge in "pochissimo")
    - ...

Metodologie di correzione automatica

(... continua ...)

Normalizzazione dell'input

- **Espressioni dipendenti dal contesto** (Microsoft Word XP)
  - Regole grammaticali:
    - **...**
    - **Articoli** (Il yogurt è un alimento molto indicato per i bambini)
    - **Elementi della frase** (segnala un insieme di errori commessi con una certa frequenza e che coinvolgono diversi elementi della frase. Esempi di errori rilevati:
      La torre di Pisa è tanto alta come bella.
      ma anche: ho mangiato tanto cioccolato come quando ero bambino > sostituire come con quanto)
    - **Preposizioni** (segnala l'esattezza nell'uso delle preposizioni insieme con sostantivi, aggettivi, pronomi, verbi ed avverbi, e segnala alcune tra le più comuni forme del parlato che sono errate nei testi scritti. Esempi di errori rilevati: Il nonno si è addormentato come al solito. La nuova macchina stampa 100 copie all'ora. Con domani inizieremo la costruzione della seconda ala dell'edificio

Metodologie di correzione automatica

(... fine)

Normalizzazione dell'input

- **Espressioni dipendenti dal contesto** (Microsoft Word XP)
  - Regole di stile:
    - **Espressioni da evitare / parole ridondanti** (Ed è per questo che abbiamo deciso di modificare i piani di produzione, Per potere avere una promozione, bisogna meritarsela. Quella maionese è lievemente acidula. Le domande devono essere presentate entro e non oltre le ore 17 del 12 ottobre)
    - **Leggibilità** (l'arciere non sapeva scegliere fra frecce rosse e frecce verdi. Il treno arrivò a Ascoli con due ore di ritardo. Il di lui cane è molto affettuoso)
    - **Termini ripetuti** (La casa vicina al ponte è più bella della casa di tuo padre. Per eliminare un problema, abbiamo eliminato anche molte cose utili)
    - **Uso errato** (Questi ragazzi hanno un gran spirito d'iniziativa. Abbiamo deciso di comprarlo sia lui che io. Malgrado tutto, siete riusciti ad arrivare in tempo a scuola)

Vincoli sull'input: cellulari e T9

(inizio ...)

- **Input nei dispositivi mobili: la scrittura degli SMS**

1. **definizione precisa del problema**
   composizione il più veloce possibile dei messaggi di testo tenendo conto dei vincoli della tastiera
2. **raccolta dati**
   esempi di messaggi, parole utilizzate, struttura delle parole
3. **classificazione**
   problemi probabilistici, semplicemente combinatori, morfologici
4. **ricerca di soluzioni adeguate ed efficienti**
   modelli di selezione per numero minimo di pressioni, modelli probabilistici

Vincoli sull'input: cellulari e T9

(... continua ...)

- **Vincoli della tastiera e metodi di composizione di SMS**
  (Silfverberg e al. 1999)
- **Alcune soluzioni possibili:**

	C	A	S	A	tot
Multi-press	2-2-2	2	7-7-7-7	2	**8**
two-key	2-3	2-1	7-4	2-1	**8**
T9	2	2	7	2	**4**

Vincoli sull'input: cellulari e T9

(... fine)

	abc	abc	pqrs	abc
T9	2	2	7	2

- **Risorse linguistiche necessarie per il T9**
  - Vocabolario
  - Indici di frequenza (es. premendo 6-6 in inglese "ON" viene selezionata prima di "NO" sulla base di osservazioni statistiche basate su corpora, in questo caso il British National Corpus, si calcola che il lavoro di disambiguazione non superi il 5% delle produzioni)
- **Risorse non linguistiche per valutare i modelli**
  - Legge di Fitts (modello quantitativo di valutazione dei movimenti rapidi diretti ad un fine)
- **Risultati** (in Words Per Minutes, wpm)
  - Multi-press: 25-27 wpm
  - Two-key: 22-25 wpm
  - T9: 41-46 wpm

Prossima lezione

(Mercoledì, 14 Marzo, ore 16:30-18:30, **Aula G, Via Roma 47**)

- **Laboratorio!**
  - Childes
    - esplorazione struttura del corpus
    - uso di espressioni regolari per estrarre informazioni linguistiche
  - PCKimmo
    - morfologia a due livelli
    - implementazione di macchine a stati finiti per l'analisi lessicale