L'analisi morfologica

a cura di Utzeri Irene

A writer is someone who writes, and a stinger is something that stings. But fingers don't fing, grocers don't groce, haberdashers don't haberdash, hammers don't ham, and humdingers don't humding.

Richard Lederer, *Crazy English*

Per computare correttamente le forme morfologiche di una parola bisogna conoscere:

- Spelling rules (regole ortografiche): ci dicono, ad esempio, che il plurale delle parole terminanti in –y in inglese si forma trasformando la –y in –i (se preceduta da consonante) e aggiungendo –es.

  lady → ladies ma day → days

e…

- Morphological rules (regole morfologiche): ci dicono, ad esempio, che *fish* al plurale rimane tale (one fish, two fish, red fish..) e che il plurale di *foot* si ottiene cambiando le vocali *(feet).*

Esistono due tipi di processi che possono intervenire nella computazione morfologica:

- Il parsing

Fare parsing significa riconoscere un input ed assegnargli una struttura adeguata.

Es. going (**surface o input form**)

VERB-go+GERUND-ing (**parsed form**)

- Lo stemming (da *stem, radice)*

Nell'ambito del recupero di informazione è quel processo che consiste nel ricondurre *(map)* una forma derivata/flessa alla rispettiva radice.

Es. Foxes ⟶ fox

L'utilità del parsing morfologico

- Nel recupero di informazioni, attraverso il riconoscimento della radice e di features morfologici che ne specificano la natura (+N,+SG,+Pl..)

Es. cities=city +N +Pl

- Nella traduzione automatica, per render conto della corrispondenza non univoca delle parole nel passaggio da una lingua all'altra.

L'utilità del parsing morfologico

Es. va e aller si traducono entrambe con go!

- Nello *spell checking*, perché sono le conoscenze morfologiche a dirci se una stringa di caratteri costituisce una parola in una certa lingua oppure no.

Limiti del parsing morfologico: il problema dell'ambiguità.

Se il parser riceve in input una parola ambigua restituirà più di un output ma, essendo una macchina,non sarà in grado di decidere qual è la parsed form adeguata.

Da ricordare: la disambiguazione richiede la conoscenza del contesto!

Cosa bisogna conoscere per costruire un parser?

- Lessico: l'insieme degli stem e degli affissi che compongono ciascuna parola (morphological features); ci danno le informazioni essenziali di ogni stem (nome, verbo, numero..)
- Regole morfotattiche: come si combinano più morfemi all'interno

…

…

di una parola. (es. F è sempre esterna a D).

- Regole ortografiche: (spelling rules) entrano in gioco quando due morfemi si combinano tra loro.

(es.in+ragionevole=irragionevole..)

Qualche esempio.

input	Morphological Parsed Output
monti	monte +N +PL
monte	monte +N +SG
noto	(notare +V +1SG) o (noto +A + SG)
noti	(notare +V +2SG) o (noto +A + PL)
amo	(amare +V +1SG) o (amo +N +SG)
ami	(amare +V +2SG +PRES) o (amare +V +1,2,3SG +CONG) o (amo +N +PL)

Ancora qualche esempio.

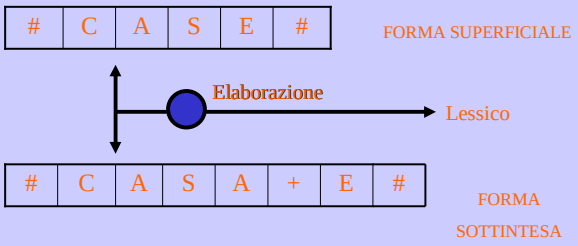
legge	(leggere +V +3SG) o (legge +N +SG)
leggi	(leggere +V +2SG PRES) o (leggere +V +2SG IMP) o (legge +N +PL)
letto	(leggere +V +PART PASS) o (letto +N +SG)
lucido	(lucidare +V +1SG) o (lucido +A +SG) o (lucido +N +PL)
presto	(prestare +V +1SG) o (presto +AVV)
parto	(partire +V +1SG) o (parto +N +SG)

Come fare l'analisi morfologica.

- **obiettivo**: riconoscere una stringa ben formata di caratteri e metterla in relazione con la struttura di morfemi che la compongono.
- **strumenti**:

1. Modello teorico
2. Finite-State Automata (FSA)
3. Finite-State Transducers (FST, trasduttori)

Modello teorico



Finite-State Automata (FSA)

A cosa servono gli automi a stati finiti?

Per verificare se una stringa di caratteri è una parola del lessico (di una lingua data, L) oppure no.

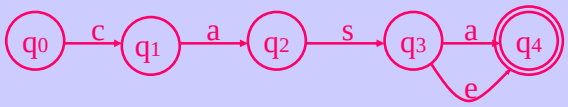
Il comportamento dell'automa è determinato da:

1. Lo stato in cui si trova
2. L'input che riceve

Formalmente un FSA è definito come una quintupla <Q,Σ,q0,F,δ> dove:

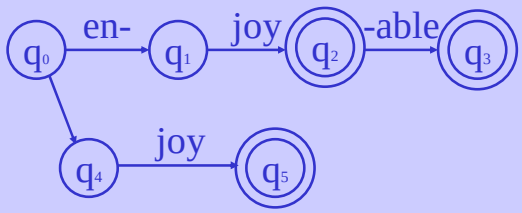
- Q =insieme finito e non nullo di stati
- Σ =alfabeto finito e non nullo di caratteri accettabili in input
- q_0 =stato iniziale,con q0 Є Q
- F =insieme di stati finali, con F Є Q
- δ =insieme delle regole di transizione definite in Q× Σ su Q

Ecco un FSA che riconosce la parola *casa* ed il suo plurale:



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.

Adeguatezza dei FSA nel rappresentare certe proprietà morfologiche (Sproat 93)



Limiti di FSA

FSA non ha memoria. Ciò significa che tale macchina non ricorda le transizioni avvenute, ma soltanto l'ultimo input ricevuto in base al quale si comporta.

E' come se la stringa "consumasse" i caratteri man mano che la macchina procede.

Conseguenze dell'amnesia di FSA

1. FSA non può descrivere un linguaggio naturale nella sua complessità, ma solo alcuni fenomeni che lo caratterizzano.
2. L'unica grammatica che gli FSA sono in grado di rappresentare è quella che Chomsky ha definito di tipo 3, ovvero quella formata da espressioni regolari.

L'inglese non è una lingua a stati finiti (regolare) (Chomsky1956,57,59)

E' impossibile costruire una macchina a stati finiti che produca tutte e solo le frasi grammaticali dell'inglese (pag.26, SS).

Infatti esistono strutture del tipo:

a. If S1 then S2
b. Either S3 or S4
c. The man who said S5 is arriving today

Perché il linguaggio regolare è inadeguato per rappresentare il linguaggio naturale?

- Non cattura le espressioni speculari (se…allora…) (né…né…)
- Non riesce a descrivere le strutture ad incassamento centrale:

Es. Al topo, che il gatto cacciò, piace il formaggio.

→

- Ineleganza e implausibilità psicolinguistica (vedi l' accordo a lunga distanza, Pullum & Gazdar 82)

Es.Qual(i/e) problem(i/a) dice il tuo professore (è/sono) irrisolvibil(i/e)?

Alcuni esempi di linguaggi non regolari.

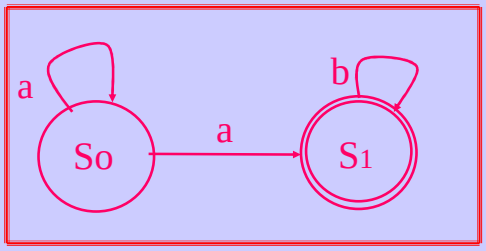
- ab, aabb, aaabbb,…,tutte e solo le frasi consistenti di *n* occorrenze di a seguite da *n* occorenze di b;
- aa, bb, abba, baab, aaaa, bbbb, …,tutte e solo le frasi costituite da una stringa x seguita dall' immagine speculare di x;
- aa, bb, abab, baba, aaaa, bbbb,…,tutte e solo le frasi costituite da una stringa x di a e di b seguita da un' identica stringa x.

Questi linguaggi non possono essere descritti da un FSA perché una volta generata la stringa di a la macchina non ha modo di "ricordarsi"quante occorrenze di a ha prodotto per riprodurle con b.

Esempio di linguaggio regolare…

aabbb, abbbb, aaaaaabbbbbbbbbb, ….,tutte le frasi costituite da *n* occorrenze di a seguite da m occorrenze di b.

Questo è un linguaggio regolare: la macchina che lo computa, una volta passata dalla generazione di a alla generazione di b, non ha il problema di "ricordarsi"il numero delle occorrenze.



…e dell'automa che lo descrive.

Proprietà utili di FSA

Gli FSA permettono di gestire adeguatamente la relazione di precedenza lineare.

Finite-State Transducers (FST, o Trasduttori)

- Un FST mette in relazione due FSA.
- Esso costituisce un sistema economico utile per rappresentare l'analisi morfologica.
- Associa una descrizione strutturale ad una stringa di caratteri riconosciuta come appartenente al lessico.

Per non confondersi..

FST hanno funzioni più generali degli FSA: gli FSA descrivono un linguaggio formale definendo un insieme di stringhe ben formate, mentre gli FST definiscono relazioni tra insiemi diversi di stringhe.

Gli FST possono essere usati come:

- riconoscitori
- generatori
- traduttori
- correlatori tra insiemi

Formalmente un FST è definito come una quintupla <Q,Σ,q0,F,δ> dove :

- 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. Σ è sottinsieme di IxO. ε 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' se la relazione i:o è definita. δ è quindi una relazione da Q x Σ su Q.

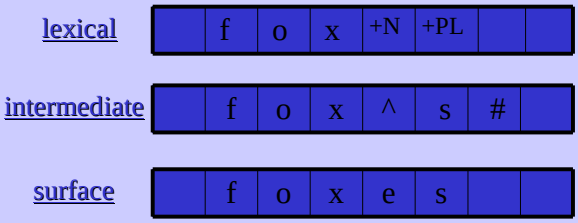
Koskenniemi('83) propone un modello di morfologia a due livelli.
Two-level morphology rappresenta una parola come una corrispondenza tra un livello lessicale ed uno superficiale (simile al modello teorico).
Questi due livelli devono essere messi in una qualche relazione significativa dal punto di vista morfologico. Tale modello è implementabile con l' uso di FST.

Esempio

Lexical		c	a	t	+N	+PL	
Surface		c	a	t	s		

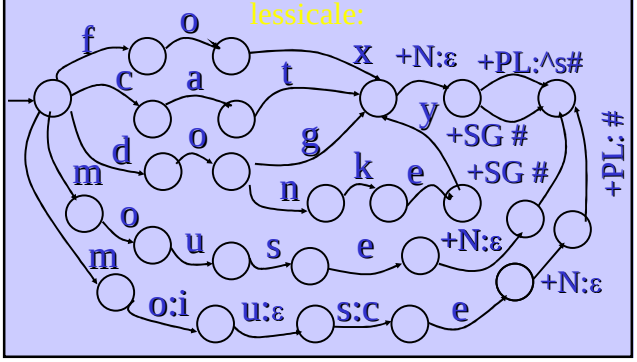
Un trasduttore utilizza FSA per abbinare stringhe di input a stringhe di output.

Teoricamente le relazioni tra stringhe possono essere definite anche su più livelli utilizzando output intermedi.

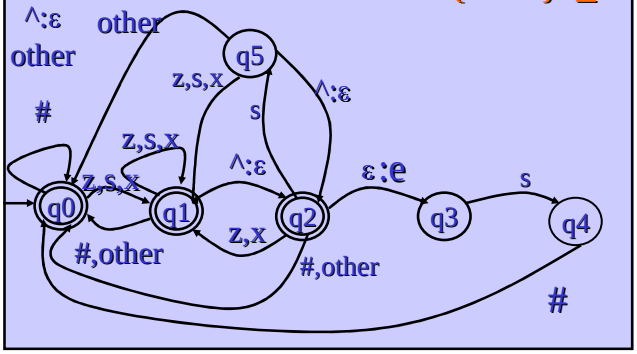


Tra ogni coppia di livelli c'è un *two-level transducer*.

Il livello lessicale è messo in relazione con il livello intermedio dal trasduttore lessicale:



Tra il livello intermedio e il livello superficiale opera la regola ortografica dell' inserzione della e: ε ⟶ e/{z/x/s}^_s#



Inadeguatezza del modello per trattare fenomeni morfologici complessi.

Alcune proprietà morfologiche non possono essere gestite da FST.

ES. il fenomeno dei plurali:

banco > banchi ma amico > amici

E non dimentichiamo i casi irregolari: uomo > uomini

Due tipi di lingue rispetto a M:

- A M concatenativa: lingue in cui i morfemi si uniscono tra loro per formare le parole.

Aggiungendo affissi diversi ad una base si ottengono parole differenti.

...

- A M non concatenativa: per flettere o derivare una forma si aggiungono vocali o si rafforzano consonanti (*templatic morphology*, morfologia a modelli). E'questo il caso delle lingue semitiche.

In altre lingue invece è possibile inserire infissi in mezzo alla parola, come succede in Tagalog.

ESEMPIO

- In Ebraico:

lmd=apprendere
lamad=studiò
lumad=fu insegnato

ESEMPIO

- In Tagalog: hingi=prestare
um=colui/colei che V

h-um-ingi=colui/colei che presta

Concludendo…

Come si vede dagli esempi, nelle lingue naturali possono essere presenti fenomeni morfologici molto complessi dal punto di vista computazionale.
In tutti questi casi i FST risultano inadeguati.