

Interazione Uomo-Macchina II:

Laboratorio di Interfacce Intelligenti

Fiorella de Rosis

Esercitazione 5

NLU: parsing probabilistico

Tutor esercitazione: Nicole Novielli

Prerequisiti

Conoscenze di base sui concetti di Natural Language Understanding presentati a lezione

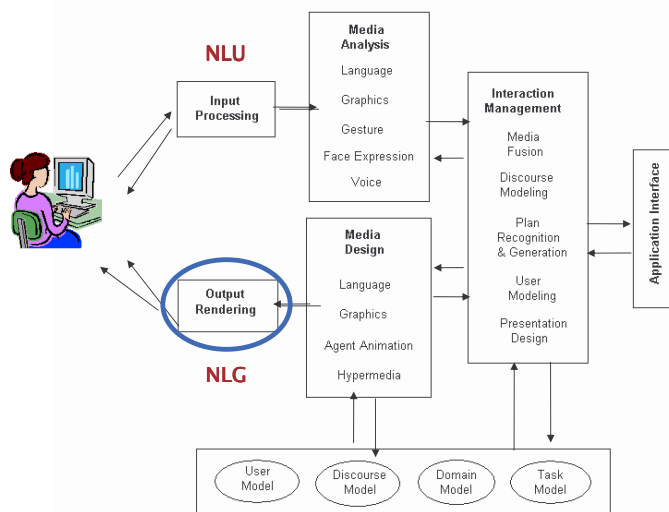
Materiale

- Dispense del corso, Unità 7 (reperibili sul sito web)
- Slide dell'esercitazione (online dopo l'esercitazione in laboratorio)

Obiettivi

- Sviluppo di una grammatica probabilistica per il riconoscimento di atti comunicativi

Architettura di una interfaccia intelligente (da Maybury e Wahlster)



Semantica

L'**obiettivo** del riconoscimento in un sistema di dialogo è quello di **formalizzare le frasi in linguaggio naturale tramite un linguaggio ad hoc** (ad es. linguaggio logico), allo scopo di 'ragionare' su di esse.

The dog ate.
Ate(dog)

A man saw a cat with a telescope

Saw(man, CatWithTelescope) gatto dotato di telescopio ☺

Oppure: Saw(man, Cat, UsingTelescope) usando un telescopio nelle due interpretazioni della frase che abbiamo visto.

Jack downloaded the computer game

Downloaded(Jack, ComputerGame)

Con lo speech (spontaneo!) questo compito diventa *molto* complesso...

S: A at American Express may I help you?
 U: yeah this is B. I need to make some travel arrangements
 S: ok and what do you need to do?
 U: ok on June sixth from San Jose to Denver, United
 S: leaving at what time?
 U: I believe there's one leaving at eleven o'clock in the morning
 S: yeah that's United flight four seventy
 U: ok
 S: leaves at eleven a.m. and arrives Denver at two twenty p.m. out of San Jose
 U: that's the one
 S: and a return?
 U: return June seventh
 S: leaving at what time?
 U: uh I believe there's one at five sixteen
 S: yeah that's United flight six fifty nine it arrives back into San Jose at six forty two
 U: ok

Frase incomplete
 manca la formula di richiesta ('I need a flight', 'I'm leaving', etc...)

Casi in cui la semantica della frase può essere interpretata solo alla luce del contesto (move precedente del sistema e storia del dialogo)

o qualsiasi forma verbale

Concentriamoci sui problemi relativi alle frasi incomplete...

5

Per risolvere problemi dovuti a *frasi parziali* o a *modi diversi* di esprimere lo stesso concetto posso utilizzare le **grammatiche probabilistiche**

Riprendiamo alcuni concetti teorici

Si tratta di grammatiche che ci consentono di assegnare **un valore di probabilità** per ogni **frase** appartenente al linguaggio ad esse associato

Formalmente, differiscono dalle grammatiche tradizionali semplicemente perché **ad ogni produzione viene assegnato un valore di probabilità**

Nota: la somma dei valori di probabilità associati a produzioni con parte sinistra identica deve essere uguale a 1

Esempio: **Assioma**

S → NP VP [.80]
 S → AuxNP VP [.15]
 S → VP [.05]
 NP → Det Nome [.20]
 NP → NomeProprio [.35]
 NP → Nome [.05]
 NP → Pronome [.40]

Nome → Noun [.75]
 Nome → Noun Nome [.20]
 Nome → NomeProprio Nome [.05]
VP → Verbo [.55]
 VP → Verbo NP [.40]
 VP → Verbo NP NP [.05]
 Det → that [.05] | the [.80] | a [.15]
 ...

Non Terminali

Terminali (regola lessicale)

6

Come usare l'incertezza nelle produzioni

La probabilità di un albero di parsing T per una frase S si può stimare come

$$P(T,S) = \prod_{n \in T} p(r(n))$$

Se diverse interpretazioni sono possibili, l'algoritmo di parsing sceglierà, tra di esse, l'albero con probabilità massima.

7

Un Parser Probabilistico in Java¹

Parser probabilistico open source che permette di definire grammatiche probabilistiche

Riceve in input una stringa in linguaggio naturale e fornisce in output la rappresentazione grafica di tutti i possibili alberi di parsing della stessa

Consente di definire grammatiche in CNF (Chomsky Normal Form)

Quindi regole del tipo

$S \Rightarrow NP VP$
 $NP \Rightarrow John$

Ma consente di ampliarle con l'uso di **Null Rules** per agevolare la definizione di grammatiche a partire da automi...

Null rules: $S \Rightarrow$

... oltre che di **Unary Rules**, per consentire la definizione del lessico

Unary rules: $S \Rightarrow VP$

8

¹ by Bob Carpenter, <http://www.colloquial.com/carp/Projects/index.html>

Un primo esempio

$S := \text{NEED_HOTEL} / \text{LEAVING_FROM}$
 $\text{NEED_HOTEL} := \text{WANT HOTEL WHERE}$
 $\text{LEAVING_FROM} := \text{LEAVING WHERE WHEN}$
 $\text{WANT} := \text{PRON VERB}$
 $\text{LEAVING} := \text{PRON VERB}$
 $\text{WHEN} := \text{TEMP_EXPR} / \text{PREP TEMP_EXPR}$
 $\text{TEMP_EXPR} := \text{TE} / \text{DET TE}$
 $\text{WHERE} := \text{PREP CITY}$
 $\text{DET} := \text{the}$
 $\text{TE} := \text{morning} / \text{afternoon} / \text{today} / \text{first of May} / \text{Monday} \dots$
 $\text{PRON} := \text{i} / \text{we} / \dots$
 $\text{VERB} := \text{want} / \text{need} / \text{would like} / \text{am leaving} / \dots$
 $\text{HOTEL} := \text{hotel} / \text{motel} / \text{an accomodation} / \text{a place to stay} / \dots$
 $\text{PREP} := \text{in} / \text{from} / \text{on} / \dots$
 $\text{CITY} := \text{London} / \text{Bari} / \text{Milano} \dots$

Le regole di riscrittura dell'assioma specificano i possibili tipi di move utente da accettare in input...

... di cui definisco la struttura globale...

... e di ogni loro parte

Nell'ambito del software che utilizzeremo, le regole che definiscono l'uso di simboli terminali prendono il nome di 'Lexical Rules'

Oltre alle Lexical Rules, il parser ammette la definizione di regole unarie (Unary Rules) binarie (Binary Rules) e nulle (Null Rules, non in questo esempio)

Regole di questo tipo (più di due simboli nella parte destra) non sono consentite -> necessario introdurre nuovi non terminali per rendere questo tipo di regole

Versione CNF della grammatica

$S \Rightarrow NP VP$ Binary Rules
 $S \Rightarrow VP$ Unary Rules
 $NP \Rightarrow \text{John}$ Lexical Rules

$S := \text{NEED_HOTEL}$
 $S := \text{LEAVING_FROM}$

$\text{NEED_HOTEL} := \text{WANT HOTEL WHERE}$
 $\text{LEAVING_FROM} := \text{LEAVING WHERE WHEN}$

$\text{WANT} := \text{PRON VERB}$
 $\text{LEAVING} := \text{PRON VERB}$

$\text{WHEN} := \text{TEMP_EXPR}$

$\text{WHEN} := \text{PREP TEMP_EXPR}$

$\text{TEMP_EXPR} := \text{TE}$

$\text{TEMP_EXPR} := \text{DET TE}$
 $\text{WHERE} := \text{PREP CITY}$

$\text{DET} := \text{the}$
 $\text{TE} := \text{morning} / \text{afternoon} / \dots$

Rendo le produzioni in forma binaria, inserendo dei non terminali ad hoc

$\text{NEED_HOTEL} := \text{WANT HOTEL}$
 $\text{HOTEL} := \text{ACC_TYPE WHERE}$
 $\text{LEAVING_FROM} := \text{LEAVING DEPARTURE}$
 $\text{DEPARTURE} := \text{WHEN WHERE}$

10

La grammatica corrispondente per il parser

Organizzo le regole in quattro file, a seconda del tipo, con la sintassi seguente

Null Rules: (non ho regole di questo tipo)

Unary Rules:

$s \rightarrow \text{Need_Hotel} : -0.5$
 $s \rightarrow \text{Leaving_From} : -0.5$
 $\text{When} \rightarrow \text{Temp_expr} .5$
 $\text{Temp_expr} \rightarrow \text{Te} .5$

Binary Rules:

$\text{Need_Hotel} \rightarrow \text{Want Hotel}$
 $\text{Hotel} \rightarrow \text{Acc_type in_Where}$
 $\text{Leaving_From} \rightarrow \text{Leaving Departure}$
 $\text{Departure} \rightarrow \text{When from_Where}$
 $\text{Want} \rightarrow \text{Pron Verb}$
 $\text{Leaving} \rightarrow \text{Pron Verb}$
 $\text{When} \rightarrow \text{Prep Temp_expr} .5$
 $\text{Temp_expr} \rightarrow \text{Det Te} .5$
 $\text{In_Where} \rightarrow \text{In_prep City}$
 $\text{From_Where} \rightarrow \text{From_prep City}$

Probabilità assegnata alla produzione. La sintassi è solo una convenzione del parser, non è un valore negativo!

Nota: se non specifico i valori di probabilità associati alle produzioni, il parser accetta comunque la mia grammatica ma assume equiprobabilità tra le regole con medesima parte sinistra

Analogamente per le regole lessicali...

11

E infine il lessico

Nota: La sintassi per le regole lessicali è invertita! (La parte destra si riscrive come indicato nella parte sinistra, corrispondente al simbolo terminale)

Lexicon

$\text{THE} \rightarrow \text{Det}$
 $\text{MORNING} \rightarrow \text{Te}$
 $\text{AFTERNOON} \rightarrow \text{Te}$
 $\text{I} \rightarrow \text{Pron}$
 $\text{WE} \rightarrow \text{Pron}$
 $\text{HOTEL} \rightarrow \text{Hotel}$
 $\text{MOTEL} \rightarrow \text{Hotel}$

$\text{WANT} \rightarrow \text{Verb}$
 $\text{NEED} \rightarrow \text{Verb}$

$\text{IN} \rightarrow \text{In_prep}$
 $\text{FROM} \rightarrow \text{From_prep}$
 $\text{BARI} \rightarrow \text{City}$
 $\text{LONDON} \rightarrow \text{City}$
 $\text{MILANO} \rightarrow \text{City}$

ecc...

ecc...

ecc...

Nota: anche per le lexicon rules, se non specifico i valori di probabilità associati alle produzioni, il parser accetta comunque la mia grammatica ma assume equiprobabilità (in questo caso, tra le regole che presentano a destra la medesima stringa di non terminali, perché la sintassi è invertita)

Prima di utilizzare una grammatica per task di riconoscimento, è sempre necessario validarla verificando che consenta di riconoscere TUTTE e SOLE le stringhe del mio linguaggio, verificando che tali stringhe rispondano a requisiti di correttezza SINTATTICA e SEMANTICA

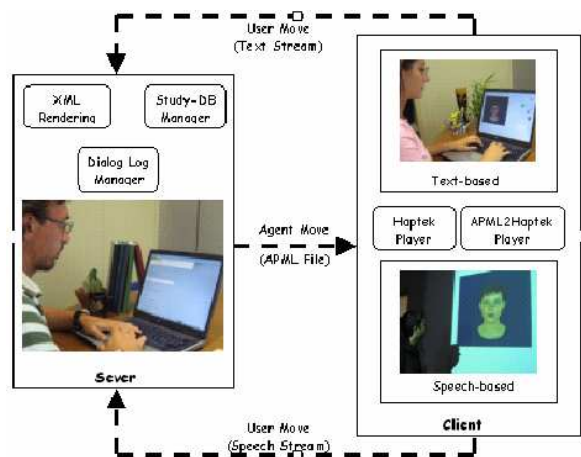
Come si comporta questa grammatica?

Quali stringhe riconosco?

12

Un esempio più complesso

Ci proponiamo di riconoscere gli atti comunicativi dell'utente in un dialogo con un Agente Animato raccolto con il metodo del Wizard of Oz:



13

Uno stralcio da uno dei dialoghi

Con relativa codifica delle move utente, in termini di atto comunicativo espresso

Oz: Ciao, il mio nome e' Valentina. Sono qui per darti dei suggerimenti su come migliorare la tua dieta.
 Subject1: piacere, mi chiamo Francesco
Friendly Self Introduction

Oz: Quali sono le tue abitudini alimentari?
 Subject2: seguo una tipica dieta mediterranea
Inform

...

Oz: Che cosa hai mangiato per pranzo?
 Subject5: peperoni ripieni
Inform

...

Oz: Quanta frutta mangi al giorno, in genere?
 Subject7: Mangio sempre la frutta dopo i pasti
Inform

Oz: Ti piacciono frutta e verdura?
 Subject8: si, io adoro la frutta
Inform

Oz: Bene, bisognerebbe mangiare frutta e verdura tutti i giorni
 Subject8: Quante porzioni di verdura?
AskInfo

...

Oz: Il tuo peso ti sembra giusto, o vorresti cambiarlo?
 Subject11: devo perdere alcuni chili... cosa mi consigli?
AskInfo

Oz: Bisognerebbe mangiare solo se si ha fame, ed aspettare che la digestione sia completa, prima di assumere altri cibi.
 Subject13: Penso che il tuo consiglio non sia affatto facile da seguire!
Object

Oz: Mangiare ad orari fissi aiuta ad evitare di saltare i pasti e di trascurare sostanze che spesso non vengono compensate nei pasti successivi.
 Subject14: sono d'accordo con te
Confirm

14

Riconoscimento di atti comunicativi

Proviamo a formalizzare alcuni degli atti comunicativi frequenti nel dominio dell'healthy eating, scrivendo una grammatica probabilistica per il parser in Java

Ipotizziamo di voler riconoscere tre diversi tipi di reazione dell'utente ad un suggerimento fornito dal sistema

Supponiamo di disporre di un corpus in cui le move utente sono annotate in funzione dell'atto comunicativo e di osservare la seguente **distribuzione**

S: Dovresti mangiare frutta e verdura tutti i giorni

U: lo adoro la verdura!

33% Inform (U Like(U, veg))

Il fatto che il corpus sia equidistribuito, ci assicura un 'bias' in meno al momento della stima delle probabilità e quindi del riconoscimento

U: Quante porzioni di verdura?

33% AskInfo (U HowMany°(veg_portions))

U: Ma non riesco a mangiare frutta tutti i giorni!

34% Object(U CanNot(eat_veg, everyday))

15

A partire dalla distribuzione del corpus

Inform 33%

AskInfo 33%

Object 34%

Definisco le regole di riscrittura dell'assioma

S -> Inform [.33]

S -> AskInfo [.33]

S -> Object [.34]

Procedo quindi con l'analisi del corpus per astrarre la struttura di ognuno degli atti comunicativi che intendo riconoscere

16

Analisi delle macrostrutture

a partire da un corpus di dati

La teoria degli atti comunicativi di Austin e Searle può venirci in aiuto nella scelta dei simboli non terminali e nella definizione della grammatica

Inform

Inform Speaker (S) Hearer (H) ϕ

dove ϕ è una formula che denota un fatto

Scopo: S desidera che H sappia che ϕ è vero;

Condizioni: S crede che H non conosca o non ricordi il valore di verità di ϕ

Effetto: H crederà che ϕ sia vero

Esempi:

Inform S Implies(Running, InShape)
'Correre aiuta a mantenersi in forma'

Inform S H Like(S, vegetables)
'Io adoro la verdura!'

L'argomento oggetto dello scopo dell'atto comunicativo
Potrebbe diventare il predicato della formula

Si assume il valore di verità di ϕ .
A seconda delle scelte fatte (*quali non terminali* utilizzo e come implemento la *procedura di mapping* dall'albero di parsing alla forma logica) posso avere output sintatticamente differenti ma con semantica identica

Es.: 'Io detesto la verdura!'
Inform S H Hate(S, vegetables)
Inform S H -Like(S, vegetables)

Scelta dettata dall'uso che faccio della conoscenza estratta,¹⁷
e da come la rappresento nel mio sistema

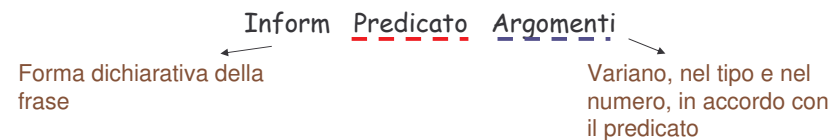
Analisi delle macrostrutture

Struttura generica di una Inform

Analizzo il corpus di dati a disposizione e operando le dovute semplificazioni (si tratta di un trascritto di un'interazione via speech, quindi è necessario eliminare ripetizioni, errori ortografici ecc...) ed astraggo la struttura generica di una Inform

Mangio 2-3 porzioni di frutta secca Inform Eat DFruit
Non mangio verdura molto spesso Inform -Eat Veg Often / Inform Eat Veg Sometime
Mi piace qualsiasi tipo di verdura Inform Like Veg
Se devo scegliere, preferisco la frutta Inform Pref Veg

Posso riassumere quindi la struttura generale di una inform come



18

Analogamente per AskInfo e Object

AskInfo

A che serve la vitamina E?

AskInfo Effects Vit_E

Dove si trova la vitamina C?

AskInfo HowToAssume
Vit_C

Object

C'è un problema: io lavoro e quindi non posso cucinare a pranzo

Think_Object Inform

Non sono d'accordo, i carboidrati gonfiano

Think_Object Claim

La object viene espressa utilizzando una formula che esprime disaccordo o opinione (Think_Object) seguita da una Inform in cui l'utente dichiara informazioni relative alle proprie abitudini o preferenze, o da un Claim generico a supporto della propria opinione/obiezione.

19

Una frase molto semplice

- Inform - U dice 'Io adoro la verdura'
'Io detesto la verdura'

Abbiamo bisogno di un set di produzioni che definiscano

- la struttura della frase (tramite produzioni che coinvolgono l'uso di *simboli non terminali*)
- il lessico con cui la frase viene espressa (*set di terminali*, il linguaggio usato per comporre la frase)

Proviamo ad impostare le regole che consentono di riconoscere inform semplici, come nell'esempio...

20

Rules

Nullary Rules:

Unary Rules:

s -> Inform : -0.33
s -> AskInfo : -0.33
s -> Object : -0.34

Binary Rules:

Inform -> Like Veg : -0.6
Inform -> Hate Veg : -0.4
Like -> I E_Like : -1.0
Hate -> I E_Hate : -1.0
Claim -> Like Veg : -1.0
Object -> Think_obj Like Veg : -0.3
Object -> Think_obj Hate Veg : -0.5
Object -> Think_obj ShDo EatVeg : -0.2
Veg -> The Vegetables : -1.0
ShDo -> Not Can : -1.0
EatVeg -> Eat Vegetables : -1.0
ecc...

Lexicon

VEGETALI-> Vegetables : -0.3
FRUTTA -> Vegetables : -0.4
VERDURA -> Vegetables : -0.3
LA -> The: -1.0
DETESTO -> E_Hate : -1.0
IO -> I: -1.0
MANGIARE -> Eat : -1.0
POSSO -> Can : -1.0
NON -> Not: -1.0
ADORO -> E_Like: -1.0
ecc...

Provate ad estendere questa grammatica in modo da riconoscere anche AskInfo ed Object

Stimo tutte le probabilità utilizzando un approccio frequentista

Nota: pur avendo una struttura simile all'Inform, la parte dichiarativa della Object si differenzia per le probabilità osservate nel corpus: è più facile che in una object si esprima un'attitudine negativa verso l'argomento in discussione

Parser Java: come avviene il riconoscimento

Nel momento in cui la grammatica viene compilata, il parser crea una *Symbol Table* dei non terminali

Symbol Table

0: I
1: Like
2: Hate
3: The
4: Vegetables
5: Can
6: Not
7: Eat
8: s
9: Inform
10: Object
11: Veg
12: ShDo
13: EatVeg
ecc...

L'output del parser è grafico (rappresentazione grafica di tutti i possibili alberi di parsing)

Guardando nel codice e nella documentazione si nota che c'è una rappresentazione in forma di stringa (Standard Penn Treebank II notation)

Pertanto, all' input:

'Adoro i vegetali'

corrisponde la stringa di parsing

8 (9 (1 ADORO) (11 (3 LA) (4 VERDURA)))

cui è associato un valore di probabilità pari al prodotto delle probabilità delle produzioni utilizzate per ottenerla

22

Mapping in linguaggio logico

Una volta ottenuta la stringa di parsing in output si può facilmente farne un mapping in linguaggio logico, sfruttando la Symbol Table

Symbol Table

0: I
1: Like
2: Hate
3: The
4: Vegetables
5: Can
6: Not
7: Eat
8: s
9: Inform
10: Object
11: Veg
12: ShDo
13: EatVeg
ecc...

8 (9 (1 ADORO) (11 (3 LA) (4 VERDURA)))

Azione	Risultato
L'8 corrisponde all' <i>assioma</i> della grammatica, indica che il riconoscimento è andato a buon fine. Non è utile ai fini del mapping	---
Il 9 si traduce in 'Inform' il cui argomento è sempre U perché si sta analizzando una move utente	Inform U ...
1 = Like, denota il predicato oggetto dell'Inform.	Inform U, Like(U,
11 = Veg, denota l'argomento del Like	Inform U, Like (U, Veg)

Attenzione: l'implementazione della procedura di mapping è strettamente correlata al modo in cui avete definito la grammatica e quindi alla struttura ed ai livelli che appariranno nel parsing tree

23

Il software è molto semplice da usare.

Il problema è definire la grammatica e associare alle sue produzioni gli elementi di incertezza.

Per alcuni atti comunicativi, la grammatica può essere 'grossolana' e semplice da definire

24

Un esempio: riconoscimento di mosse di 'Self Introduction'

A volte può essere d'aiuto partire dalla costruzione di un automa per la formalizzazione del tipo di frase che intendo riconoscere, per poi derivarne la grammatica corrispondente

Piacere mi chiamo Nicole
Ciao Valentina, io sono Nicole
Il mio nome è Nicole
 ...

Si nota facilmente come le mosse elencate abbiano una **struttura più o meno costante**

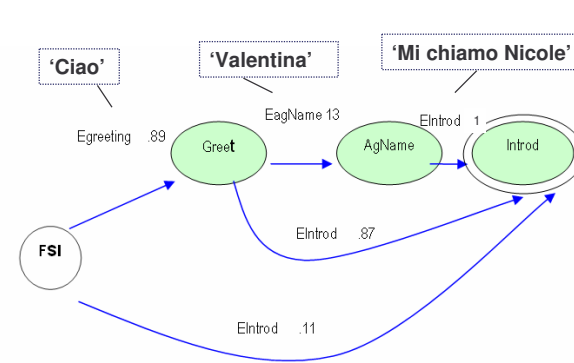
Una formula di presentazione sempre presente...

...Spesso preceduta da una forma di saluto, utilizzata dagli utenti più amichevoli...

.. Che a volte si rivolgono all'agente chiamandolo per nome, nel salutarlo

Proviamo ad ipotizzare una Markov Chain molto semplice con la grammatica corrispondente... ²⁵

Una Markov Chain per la Self Introduction



Molto semplice poiché utilizza **Categorie Lessicali** come simboli terminali

EGreeting = {Ciao, piacere...}

espressioni di saluto

EagName = {Valentina, cara...}

appellativi per l'agente

EIntrod = {mi chiamo, mio nome è...}

formule di presentazione

Lo **stato iniziale** diventa l'assioma della grammatica

Diverse **possibilità di riscrittura** a seconda dei 'path' percorribili nella MC

Si traduce facilmente nella seguente grammatica:

FSI → EGreeting Greet [.89]

FSI → EIntrod Introd [.11]

Greet → EAgName AgName [.13]

Greet → EIntrod Introd [.87]

AgName → EIntrod Introd [1]

EGreeting → *ciao | piacere | ...*

EIntrod → *mi chiamo | mio nome è | ...*

EagName → *Valentina | cara | ...*

Come assegnare le probabilità? ²⁶

Come assegnare i valori di probabilità alle produzioni?

- sulla base della mia conoscenza del dominio e della lingua
- stimandole a partire da un corpus di dati (approccio che noi adoteremo)

Vediamo come...

L'esempio della Friendly Self Introduction

Corpus di dati

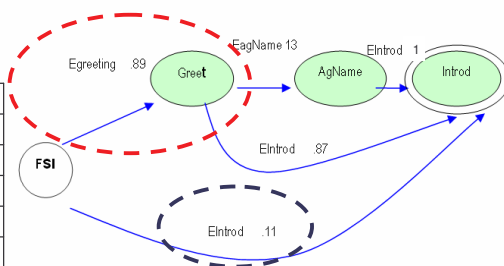
Ciao il mio nome è Angela
Ciao sono Mauro, puoi darmi suggerimenti
Ciao sono antonella
Ciao io sono Vincenzo
Ciao Valentina, io sono Mariangela
Allora mi presento: mi chiamo Marino e sono qui per..
Ciao io mi chiamo Carlo e sono qui ad ascoltarti
Va bene, ciao, io mi chiamo Imma eh... ascolto...
Sono Simone e ci tengo alla mia alimentazione
Piacere mi chiamo Francesco
Ciao! Mi chiamo Isa e sono curiosa...
Ciao sono Carlo e sono molto contento...
Ciao il mio nome è Oriana
Ciao, mi chiamo Filippo
Ciao sono Marco
Ciao sono Carla
Ciao cara, io sono Eustrat
Ciao io sono Giulia

Nota: nella realtà si utilizzano corpora di maggiori dimensioni, non è attendibile stimare delle probabilità a partire da un corpus così piccolo, pena la scarsa robustezza del riconoscimento

L'esempio della FSI

Corpus di dati

Ciao il mio nome è Angela
Ciao sono Mauro, puoi darmi suggerimenti
Ciao sono antonella
Ciao io sono Vincenzo
Ciao Valentina, io sono Mariangela
Mi presento: mi chiamo Marino e sono qui per..
Ciao io mi chiamo Carlo e sono qui ad ascoltarti
Ciao, io mi chiamo Imma eh... ascolto...
Sono Simone e ci tengo alla mia alimentazione
Piacere mi chiamo Francesco
Ciao! Mi chiamo Isa e sono curiosa...
Ciao sono Carlo e sono molto contento...
Ciao il mio nome è Oriana
Ciao, mi chiamo Filippo
Ciao sono Marco
Ciao sono Carla
Ciao cara, io sono Eustrat
Ciao io sono Giulia



Nell'89% dei casi (16/18) la presentazione inizia con una formula di saluto

FSI → EGreeting Greet [.89]

Nel restante 11% l'utente inizia subito con una formula di presentazione

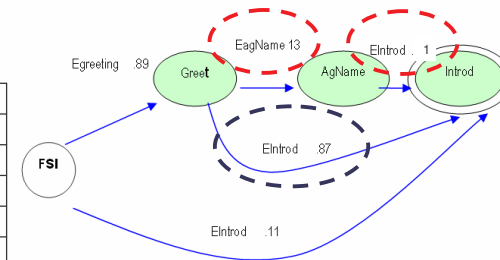
FSI → EIntrod Introd [.11]

29

L'esempio della FSI

Corpus di dati

Ciao il mio nome è Angela
Ciao sono Mauro, puoi darmi suggerimenti
Ciao sono antonella
Ciao io sono Vincenzo
Ciao Valentina, io sono Mariangela
Mi presento: mi chiamo Marino e sono qui per..
Ciao io mi chiamo Carlo e sono qui ad ascoltarti
Ciao, io mi chiamo Imma eh... ascolto...
Sono Simone e ci tengo alla mia alimentazione
Piacere mi chiamo Francesco
Ciao! Mi chiamo Isa e sono curiosa...
Ciao sono Carlo e sono molto contento...
Ciao il mio nome è Oriana
Ciao, mi chiamo Filippo
Ciao sono Marco
Ciao sono Carla
Ciao cara, io sono Eustrat
Ciao io sono Giulia



Il 13% degli utenti che hanno salutato l'agente, gli si rivolgono chiamandolo per nome o utilizzando degli appellativi amichevoli, per poi presentarsi

Greet → EAgName AgName [.13]

AgName → EIntrod Introd [1]

L'87% degli utenti che ha salutato l'agente, si presenta direttamente

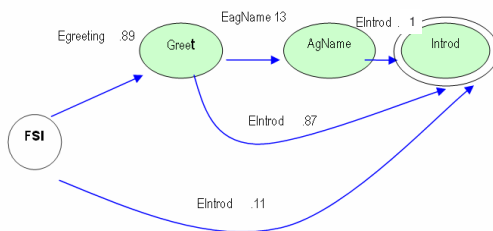
Greet → EIntrod Introd [.87]

30

L'esempio della FSI

Corpus di dati

Ciao il mio nome è Angela
Ciao sono Mauro, puoi darmi suggerimenti
Ciao sono antonella
Ciao io sono Vincenzo
Ciao Valentina, io sono Mariangela
Mi presento: mi chiamo Marino e sono qui per..
Ciao io mi chiamo Carlo e sono qui ad ascoltarti
Ciao, io mi chiamo Imma eh... ascolto...
Sono Simone e ci tengo alla mia alimentazione
Piacere mi chiamo Francesco
Ciao! Mi chiamo Isa e sono curiosa...
Ciao sono Carlo e sono molto contento...
Ciao il mio nome è Oriana
Ciao, mi chiamo Filippo
Ciao sono Marco
Ciao sono Carla
Ciao cara, io sono Eustrat
Ciao io sono Giulia



Analogamente per le regole lessicali

EGreeting → *ciao* [.94] / *piacere* [.06] / ...

EIntrod → *mi chiamo* [.33] / *mio nome è* [] / ...

EAgName → *Valentina* [.50] / *cara* [.50]

31

Proviamo a farlo con il nostro software

Null Rules: ---

Unary Rules:

s → FSI : -0.11

(supponendo di osservare, nel mio corpus l'11% di move di FSI)

Eventuali altre produzioni per la generazione di altri atti comunicativi (regole di riscrittura di s)

Binary Rules:

FSI → EGreeting Greet : -0.89

FSI → EIntrod Introd : -0.11

Greet → EAgName AgName : -0.13

Greet → EIntrod Introd : -0.87

AgName → EIntrod Introd : -1.00

Lexicon:

CIAO → EGreeting -0.94

PIACERE → EGreeting -0.06

VALENTINA → EAgName : -0.50

CARA → EAgName : -0.50

ecc...

32

Stimare la probabilità delle regole lessicali può aiutarci a risolvere i problemi legati all'ambiguità (sinonimia, polisemia) del linguaggio naturale

1. *Piacere*, mi chiamo Nicole (*Friendly Self Introduction*) (a)
2. Mangiare verdura ogni giorno per me è un *piacere* (*Inform*)

1. Ciao *cara*, mi chiamo Nicole (*Friendly Self Introduction*) (b)
2. La tariffa che mi proponi per questo volo è troppo *cara* (*Reject?*)

Le grammatiche probabilistiche, se definite in modo corretto e stimando le probabilità da un corpus sufficientemente grande, mi consentono di disambiguare situazioni di questo tipo, che non sarebbero gestibili con una semplice analisi per parole chiave.

Supponendo di aver definito correttamente una grammatica per il riconoscimento di *Inform*, *FSI* e *Reject*, allora avrò:

- per (a) $p(\text{FSI}|\text{move_a1}) > p(\text{Inform}|\text{move_a1})$ e che
 $p(\text{FSI}|\text{move_a2}) < p(\text{Inform}|\text{move_a2})$
per (b) $p(\text{FSI}|\text{move_b1}) > p(\text{Reject}|\text{move_b1})$ e che
 $p(\text{FSI}|\text{move_b1}) < p(\text{Reject}|\text{move_b1})$

33

Per l'esercitazione

Sullo stile dell'esempio proposto (riconoscimento di atti comunicativi nel dominio dell'healthy eating) impostate un piccolo caso di studio che preveda:

- l'analisi di corpora esistenti già raccolti allo scopo di
 - a. definire una grammatica probabilistica per il riconoscimento dei vari atti comunicativi dell'utente
 - b. stimare le probabilità delle produzioni a partire dal corpus selezionato

Quali atti comunicativi riconoscere? -> Il set di atti comunicativi rilevanti per il dominio scelto, va definito in funzione del tipo di move osservabili negli esempi raccolti nel Dialogue Diversity Corpus (W.C Mann, 2003)

<http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.htm>

Potete lavorare con dei sub-corpora a scelta in uno dei domini proposti:

- Tutoring
- Transport planning
- Interaction between friends
- Computer help

34