

## Generazione automatica di frasi in linguaggio naturale

Seminario per il corso di Elaborazione del Linguaggio Naturale

Anno Accademico 2004 / 2005

Mario Latronico

## Introduzione : Psicologia e Linguaggio

- La comunicazione scritta e' possibile in quanto esiste una corrispondenza tra segno (parola) e l'oggetto designato
- L'uomo e' naturalmente predisposto al linguaggio
  - a 3 mesi distingue differenti intonazioni emotive della voce (ira, gioia ...) e voci diverse.
  - a 18 mesi puo' comporre piu' di 2 parole
  - a 2 anni compone frasi sempre diverse, con regole grammaticali
  - dopo i 4 anni compaiono modi di esprimersi del tutto particolari (varianti stilistiche) legati all'ambiente di vita o del tutto personali
- Altri metodi di comunicare : verbale, scritta, Braille, gesti

da [http://www.homolaicus.com/linguaggi/psico\\_linguaggio.htm](http://www.homolaicus.com/linguaggi/psico_linguaggio.htm)

## Introduzione : Suoni e significato

- Ogni conversazione ha due *flussi* di informazione
  - Con le parole “normali” si trasmette il significato
  - Con le interiezioni come “uhm, uh, eh” si cerca di aggiungere un ulteriore commento al flusso di parole per renderlo piu' comprensibile all'ascoltatore
- Le interiezioni citate non sono usate solo nelle conversazioni, ma anche nei monologhi o nei testi scritti
  - E' stato dimostrato che gli ascoltatori comprendono meglio se le interiezioni sono presenti piuttosto che assenti

da [http://ulisse.sissa.it/site/public/ScienzaSette/s7\\_31mag02\\_3.htm](http://ulisse.sissa.it/site/public/ScienzaSette/s7_31mag02_3.htm)

## Introduzione : Suoni e significato

- Il segno linguistico è costituito dall'unione di un *concetto* (significato) e di un *immagine acustica* (significante): per esempio dal concetto di '**albero**' e dall'insieme dei suoni che formano la parola '**albero**'. Il rapporto tra significante e significato non è motivato, ma è frutto di una convenzione.

(M. Dardano, La lingua italiana)

da  
<http://www.arakhne.it/didanext/sito/letteratura/veglia/VegliaForma/SignificatoSignificante.html>

## Sistemi NLG

- *Natural Language Generation* : “Processo di costruzione di un testo o parlato in linguaggio naturale con l'obiettivo di comunicare”
  - *Input* : una qualche rappresentazione astratta e non linguistica dell'informazione
  - *Output* : Documenti, report, dialoghi ...
- Richiede la conoscenza della lingua e del dominio del discorso

## Tipi di applicazione

- Generazione di report (ad esempio informazioni sul meteo), lettere, sistemi di traduzione
- A supporto di sistemi esperti, ad esempio per diagnosi mediche
- Supporto per disabili (ciechi)
- Insegnamento (per interfacciarsi in maniera amichevole con l'utente)
- Divertimento (videogame)
- Sistemi GPS per automobili

## Sommario

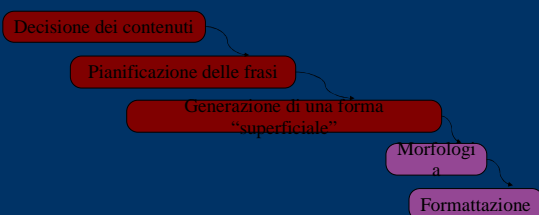
- **Architettura dei sistemi di generazione di testo in linguaggio naturale (NLG System)**
- Generazione di dialoghi per l'interazione uomo macchina a supporto di sistemi informativi
- Generazione di “disfluenze”

## Architettura di un NLG System

- Modulare, pipeline
- La maggior parte dei sistemi NLG comprende:
  - *Syntactic Realizer* : Garantisce che le strutture sintattiche (ad esempio soggetto e verbo) siano rispettate
  - *Text Planner* : Decide come gestire coerentemente le frasi, i paragrafi e altre componenti del testo

## Architettura di un NLG System

- Molti sistemi condividono l'architettura a *pipeline* per connettere i moduli di cui sono composti. Essi sono :



## Decisione dei contenuti

- Dato un input iniziale produce una forma semantica, ossia una specifica del significato del contenuto che dovrà avere il testo di output
  - Quali informazioni devono essere comunicate all'utente
- Una sorta di *rete semantica*, dove gli elementi primitivi sono concettuali invece che linguistici
  - per esempio concetti del dominio della base di conoscenza invece che parole nel linguaggio naturale

## Pianificazione delle frasi

- Convertire la rappresentazione semantica in una forma linguistica astratta specificando le parole e la loro relazione grammaticale.
- I suoi compiti comprendono:
  - Raggruppare singole parole in proposizioni
  - Associare concetti del dominio del discorso con parole e relazioni grammaticali
  - Generare espressioni di riferimento per entità individuali del dominio

## Generazione di una forma "superficiale"

- Parte di un sistema NLG che riguarda come le relazioni grammaticali sono espresse nella lingua del sistema
- Ad esempio si preoccupa di gestire la corretta gestione dei pronomi : invece di "*Giovanni rimprovera Giovanni*" viene usato "*Giovanni rimprovera se stesso*"

## Considerazioni dell'architettura a pipeline

- Pro
  - Dal punto di vista ingegneristico e' piu' facile da progettare e da mantenere
  - Dal punto di vista strutturale (ma *non* funzionale) e' simile al cervello umano, il quale e' altamente "modularizzato"
- Contro
  - Non ottimale per gestire frasi complesse
  - Non uguale al funzionamento del cervello (richiede backtracking e feedback dei moduli)

## Considerazioni dell'architettura a pipeline

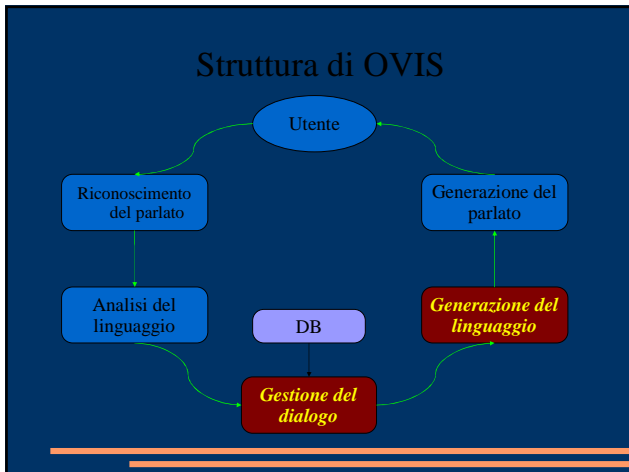
- Progettata per monologhi, quindi non sempre adatta nei sistemi NLG che prevedono il dialogo con l'utente
- *Nei dialoghi solo parte dell'informazione puo' essere fornita inizialmente*
- *Successivamente in base alla reazione dell'utente il sistema potrebbe non continuare, infatti l'utente puo' indicare che il resto dell'informazione non e' necessaria o voluta.*

## Sommario

- **Architettura dei sistemi di generazione di linguaggio naturale (NLG System)**
- **Generazione di dialoghi per l'interazione uomo macchina a supporto di sistemi informativi**
- Generazione di "disfluenze"

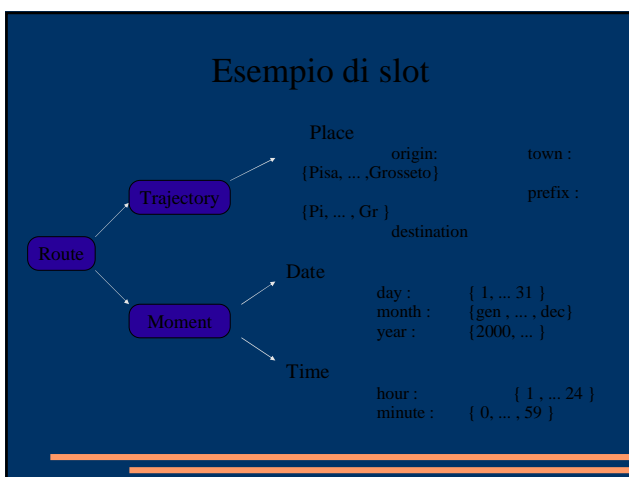
## Applicazioni : OVIS

- OVIS (*Openbaar Vervoer Informatie Systeem*) : Sistema informativo per il trasporto pubblico (olandese)
  - Fornisce informazioni sui treni e i prezzi dei biglietti, dialogando con l'utente in linguaggio naturale
- 2 Macromoduli presi in considerazione
  - *Dialog manager* : specifica quali azioni devono essere portate a compimento durante il proprio turno
  - *Language generation module* : specifica l'informazione che deve essere espressa nel dialogo



### Esempio di interazione tra DM e LGM

- DM : `the_route.trajectory.origin ??`
- LGM : *Da dove vuoi partire ? [ domanda aperta per sapere la stazione di partenza ]*
- *La struttura dati a cui accede il DM viene chiamata slot , la quale puo' essere riempita durante il dialogo con l'utente*



### Interazione tra sistema e utente

- I messaggi che genera il sistema quando dialoga con l'utente sono di due tipi:
  - Richieste di informazioni
  - Meta-azioni
    - ad esempio verifiche per capire se il riconoscimento della frase pronunciata dall'utente e' corretto
- *Strategia di focalizzazione del DM, zooming strategy*
  - *Inizia ponendo questioni riguardanti gli slot piu' in alto nella gerarchia e via via pone domande per "riempire" gli slot a livelli piu' bassi*

## Generazione del linguaggio

- Le frasi sono generate da *modelli sintattici* (*syntactic templates*) che combinano strutture marcate (tag) con condizioni che determinano se la frase e' ammissibile nello stato corrente in cui si trova il sistema
- I templates vengono usati perche' sono semplici da gestire

## Modelli sintattici

- Composti da 4 elementi
  - S : Albero sintattico con "vuoti" da riempire per le informazioni variabili
  - E : lista di funzioni di espressione usate per riempire i "vuoti" nell'albero sintattico
  - C : lista di condizioni del modello sintattico
  - L : livello, usato per la gestione del dialogo (vedi dopo)

## Esempio di modello sintattico

### Template DestinationVerify

S = Per <dest> ?  
E = dest ← Express ( the\_route.trajectory.destination )  
C = toBeVerified ( the\_route.trajectory.destination )  
L = 1

### Template DestinationTimeVerify

S = A <time> per <dest> ?  
E = dest ← Express ( the\_route.trajectory.destination ) ,  
time ← Express ( the\_route.moment.time )  
C = toBeVerified ( the\_route.trajectory.destination ) and  
toBeVerified ( the\_route.moment.time ) and  
L = 2

## Dire tutto in un unico turno

- Una domanda in un dialogo e' un'azione a turni, ossia la fine di essa indica all'altro partecipante che deve prendere la parola
- Generare piu' di una domanda in un unico turno puo' provocare confusione
  - Esempio : A che ora vuoi partire ? Verso Pisa ?
  - L'utente potrebbe iniziare a rispondere immediatamente alla prima domanda non sentendo la seconda e generando confusione nel sistema

## Dire tutto in un unico turno

- Per ovviare a questi si usa l'elemento *L*, ossia il livello del dialogo che indica quanti slot sono presi in considerazione dal template
- LGM usa la seguente strategia : sceglie sempre il template che riguarda il piu' alto numero di slot e da questo genera una sola domanda
  - il piu' alto numero di slot e' dato dal livello *L*
- Esempio, invece di dire: *A che ora vuoi partire ? Verso Pisa ?* , dira'
  - *A che ora vuoi partire per Pisa ?*

## Coerenza nei turni

- Un'informazione dal DM puo' riguardare piu' slot ma puo' essere difficile da esprimere con una sola domanda, ad esempio
  - DM : ( variant ?+ the\_route; the\_route.moment.time ?+ 10.30 )
    - LGM (a) : Vuoi avere informazioni sul viaggio e vuoi viaggiare alle 10:30 ?
    - LGM(b) : Informazioni sul viaggio, 10:30 ?
- La risposta b e' poco intuitiva ...

## Coerenza nei turni

- Per evitare questi casi viene scelto il modello sintattico che riguarda slot collegati tra loro, ossia che condividono lo stesso "genitore" nella struttura dati
- Ad esempio *origin e destination* , il genitore e' *trajectory*
- Ma anche *time e destination*, il genitore e' *route*

## Variazioni sintattiche

- Alcune variazioni sintattiche che pero' hanno la stessa semantica sono usate per non "annoiare" l'utente e per avere piu' probabilita' di ottenere una risposta giusta
- Piu' modelli che esprimono lo stesso contenuto semantico, ad esempio per il controllo degli errori sul riconoscimento del parlato
  - DM : the\_route.trajectory.destination ?+ Pisa
  - LGM : Per Pisa ?
  - Utente : (?)
  - DM : the\_route.trajectory.destination ?+ Pisa
  - LGM : Mi sembra di aver capito che vuoi andare a Pisa, e' corretto ?
  - Utente : Si

## Grounding

- In un dialogo, informazioni note ad uno dei partecipanti non necessariamente sono note all'altro
- *Grounding* : processo interattivo che consiste nell'aggiungere informazione alla "base" (ground) comune dei due interlocutori. *Procede in 2 fasi*
  - *Presentazione* : Un interlocutore manda un messaggio all'altro
  - *Accettazione* : L'altro interlocutore segnala se il messaggio e' stato correttamente capito o meno

## Grounding

- L'informazione viene aggiunta alla "base" solo se la fase di accettazione ha esito positivo
- In OVIS, dato che potrebbero esserci errori di riconoscimento del parlato, l'accettazione puo' fallire, per esempio
  - Utente : Vorrei andare a Pistoia ...
  - LGM : A Pisa ?
  - Utente : No, Pistoia
  - LGM : Pistoia ?
  - Utente : Si, e' corretto
  - LGM : Da dove vuoi partire ?
  - ....

## Wrapping

- Il wrapping (*to wrap, avvolgere, coprire*) e' legato al concetto di grounding, consiste nell'avvolgere l'informazione principale con informazioni secondarie per facilitare il dialogo
- Esempio :
  - Utente : Quando parte il prossimo treno per Pistoia ?
    - a) LGM : 14:30
    - b) LGM : Partira' alle 14:30
    - c) LGM : Il prossimo treno per Pistoia partira' alle 14:30
- Da a) a c) il sistema fornisce riposte sempre piu' concise in base a quanta conoscenza e' condivisa

## Prosodia

- *Gestione della prosodia ossia lo "studio delle caratteristiche di una lingua relative ad accento, tono, intonazione, quantita'"* [Voc. della lingua italiana Zingarelli, ed 1994]
- A volte, quando parliamo, tendiamo ad accentare le parole per dare loro un particolare significato, enfatizzarle o per segnalare un problema a livello di comprensione



## Prosodia

- *Esempio*
  - Utente : Vorrei andare da Pistoia a *Prisa* ?
  - LGM : *Prisa* ? [ Accentato, indica che il sistema non ha correttamente compreso la stazione di arrivo ]
  - Utente : Ho sbagliato. Vorrei andare da Pistoia a Pisa
  - LGM : ...

## Sommario

- Architettura dei sistemi di generazione del linguaggio naturale (NLG System)
- Generazione di dialoghi per l'interazione uomo macchina a supporto di sistemi informativi
- **Generazione di “disfluenze”**

## Generazione di disfluenze nei dialoghi

- I sistemi NLG spesso producono parlato o testi “perfetti”, ossia privi delle incertezze, ripensamenti che a volte caratterizzano il parlato umano
  - Mhhh, no, volevo dire che, cioe' ...
- Generare questo comportamento li renderebbe piu' “credibili” per un utente

## Disfluenza

- *Definizione* : “Disturbo del linguaggio parlato caratterizzato da inceppamenti e interruzioni del ritmo, ripetizioni e prolungamenti dei suoni” ( da [www.edufamily.it/get\\_lexeme.php?lex\\_id=1960](http://www.edufamily.it/get_lexeme.php?lex_id=1960) )
- *La ricerca sulla disfluenza nel linguaggio parlato e' ancora piuttosto rara*
- *Tuttavia, si e' notato che il numero di disfluenze in un discorso e' legato alla lunghezza delle frasi che lo compongono.*

## Ricerca psicolinguistica sulla disfluenza

- In una recente ricerca di psicolinguistica [Brennan, 2000] e' emerso che
  - le disfluenze apportano un utile contenuto semantico
  - il parlato “disfluente” in certi caso e' piu' facile da comprendere rispetto al parlato “perfetto”
- Un sistema NLG puo' essere reso piu' “realistico” se puo' modificare il suo output con pause appropriate tra le parole e con la giusta intonazione

## Parlato vs. testo scritto

- Rispetto al testo scritto, il parlato e' piu' “permissivo”
- *L'orecchio nota meno le disfluenze (nel parlato) rispetto all'occhio (nei testi scritti) ...*
- *... anche perche' sono molto piu' frequenti nel parlato piuttosto che nei testi scritti*
- *Ricerche effettuate dimostrano che le disfluenze aumentano l'interazione tra l'uomo e la macchina*

## Tipi di disfluenze

- I tipi di disfluenze sono molti, alcune di queste sono:
- *Fase di accettazione nel grounding :*
  - A : Hai capito ?
  - B : mm hmm
  - A : Ok, andiamo avanti ...
  - B “annuisce” indicando che la spiegazione di A e' stata compresa
- *Esitazioni :*
  - A : Ok ... scusa cosa e' questo ?
  - Breve pausa indicata dai tre punti per indicare un momento di esitazione su un punto

## Tipi di disfluenze

- *Ripensamento per grounding*
  - A : e' l'oppos ... Se qualcosa si ferma, la sua accelerazione e' il contrario della direzione della sua velocita'
  - *Inizialmente chi parla crede che il vocabolo sia noto all'interlocutore, ma quando si accorge che non e' cosi' si ferma e dopo una breve pausa fornisce una spiegazione del termine*
- *Ripetizioni*
  - A : Guarda !! Ho fatto questo
  - B : Bene, bene, molto bene !!
  - *Enfasi di B nel giudicare cio' che ha fatto A*

## Tipi di disfluenze

- *Ripetizione dell'intera frase quando (non) viene in mente una parola*
  - A : In che modo calcoli ... in che modo calcoli la variabile X ?
- *Pause per permettere di ragionare su un elemento visivo o fisico*
  - A : Ok, allora ... guardiamo la [indicando la lavagna] componente y della velocità' ...

## Approcci possibili per la generazione di disfluenze

- A volte basta semplicemente traslasciare l'espressione che viene inviata al sintetizzatore vocale e *riplanificare una nuova frase da zero*
- *E' tuttavia necessaria una temporizzazione piu' fine nel sintetizzatore vocale che deve interrompersi nel mezzo di una frase o persino nel mezzo di una parola*

## Strategie di realizzazione

- Nell'architettura a pipeline dei sistemi NLG , sono due i moduli che necessitano di modifiche, il text planner e il surface realizer
  - *Text Planner* : Deve essere in grado di *riplanificare da zero una frase, cosi' come indicare dei punti di arresto*
  - *Surface Realizer* : Deve essere in grado di *generare delle disfluenza con dei tag appropriati per il sintetizzatore vocale*
- *L'architettura a pipeline non e' piu' ottimale, un'alternativa sono i generatori incrementali, capaci di produrre soltanto frammenti di frase immediatamente prima che tutta la frase sia processata.*

## Bibliografia

- *From Monologue to Dialogue : Natural Language Generation in OVIS*, Mariet Theune , University of Twente
- *Do We Need Deep Generation of Disfluent Dialogue ?*, Charles B.Callaway, ITC-irst, TCC Division
- *Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible ?*, Ehud Reiter , CoGenTex, Inc.
- *Process that shape conversation and their implications for computational linguistics*, Susan E. Brennan, In *Proceedings of the 38<sup>th</sup> Annual Meeting of the Association for Computational Linguistics*, Hong Kong, October 2000