



Inteligența Artificială

Universitatea Politehnică București
Anul universitar 2010-2011

Adina Magda Florea

http://turing.cs.pub.ro/ia_10 și
curs.cs.pub.ro



Curs nr. 10

Prelucrarea limbajului natural

- Prelucrare LN pt achizitia cunostintelor
- Prelucrare LN pentru comunicare



1 – Prelucrarea LN pt achizitia cunostintelor



1.1 Modele ale limbajului

- Gramatici: recursiv numarabile, dependente de context (GDC), independente de context (GIC), regulate (GR)
- 1980 – GIC si GDC
- Apoi si GR

Fernando Pereira: *"The older I get, the further down the Chomsky hierarchy I go"*



Modele N-gram

- **Model N-gram de caractere** – **distributie de probabilitate** peste secvente de caractere
- $P(c_{1:N})$ – probabilitatea unei secvente de N caractere, c_1 la c_N
$$P(\text{"the"}) = 0.27 \quad P(\text{"zgq"}) = 0.00000002$$
- O secventa de simboluri de lungime n – **n-gram**
 - unigram, bigram, trigram
- Un model N-gram este definit ca un lant Markov de ordin $N-1$ (probabilitatea unui caracter depinde de caracterele precedente)



Modele N-gram de caractere

- **Trigram**

$$P(c_i|c_{1:i-1}) = P(c_i|c_{i-2:i-1})$$

$$P(c_{1:N}) = \prod_{i=1,N} P(c_i|c_{1:i-1}) = \prod_{i=1,N} P(c_i|c_{i-2:i-1})$$

Un model trigram a unui limbaj de 100 caractere

$$P(c_i|c_{i-2:i-1})$$

are 1 mil intrari



Modele N-gram de caractere

Ce putem face cu un astfel de model?

- **Identificarea limbajului** = fiind dat un text se determina in ce limba este scris (99%)
- Construiești un model trigram caracter pentru fiecare limbaj candidat l
- $P(c_i|c_{i-2:i-1},l)$; este nevoie de aprox 100 000 caractere pt fiecare limbaj

$$l^* = \operatorname{argmax}_l P(l|c_{1:N}) = \operatorname{argmax}_l P(l) * P(c_{1:N}|l) = \\ \operatorname{argmax}_l P(l) * \prod_{i=1,N} P(c_i|c_{i-2:i-1},l)$$

- **Alte aplicatii:**
 - verificare ortografie, clasificare texte in fct de tipuri, identificarea numelor proprii,...



Modele N-gram de caractere

Omogenizarea modelelor

- *Problema:* pt secvente de caractere comune, cam orice corpus va da o estimare buna ($P(" th")=0.15$)
- Dar $P(" ht")=0$ http?
- *Solutie*
- Calculam N-gram si pentru secvente cu $P=0$ sau f mica
- Calculam N-1-gram + interpolare

$$P(c_i|c_{i-2:i-1},l) = \alpha_1 P(c_i|c_{i-2:i-1},l) + \alpha_2 P(c_i|c_{i-1},l) + \alpha_3 P(c_i)$$

$$\text{cu } \alpha_1 + \alpha_2 + \alpha_3 = 1$$

Evaluarea modelelor – prin validare incrucisata



Modele N-gram de cuvinte

- In acest caz vocabularul este mai mare
- Daca max 100 car in cele mai multe limbaje, sute de mii, milioane de cuvinte
- Cuvinte noi
- Putem adauga un cuvint <NEC> in vocabular (sau mai multe)
- Bigramele si trigramele sunt mai bune decat unigramele



1.2 Clasificarea textelor

- Clasificare sau apartenența la o clasă
- Identificare limbaj, clasificare tip text, analiza stării induse, detectie spam

Detectie spam

Not-spam (Ham)

Spam

m_1, m_2, \dots

n_1, n_2, \dots

- "for cheap" "you can buy" – n-gram de cuvinte
- "yo,u d-eserve" – n-gram de caractere

Detectie spam

- Calculez $P(\text{Mesaj}|\text{Spam})$ $P(\text{Mesaj}|\text{Ham})$

- Clasific mesaj nou

$$\mathit{argmax}_{C \in \{\text{Spam}, \text{Ham}\}} P(C|\text{Mesaj}) = \mathit{argmax}_{C \in \{\text{Spam}, \text{Ham}\}} P(\text{Mesaj}|C) * P(C)$$

$C \in \{\text{Spam}, \text{Ham}\}$

$C \in \{\text{Spam}, \text{Ham}\}$

unde $P(C)$ este estimat prin numararea nr de mesaje din Spam si Ham

- Se reprezinta mesajul ca o multime de caracteristici $(\text{car}_i, \text{val}_i)$ si se poate aplica un algoritm de clasificare (invatare)
- Caracteristici – cuvinte din vocabular
- Valori – nr de aparitii in mesaj
- Alg posibili: K-nearest-neigh, SVM, AD, Bayes naiv



1.3 Regasirea informatiei

- Scop: Gasirea documentelor care sun relevante pentru o cerere utilizator

Un sistem de IR (Information Retrieval) poate fi caracterizat de:

- **Un corpus de documente** – paragrafe, pagini, texte pe mai multe apgini
- **Interogarea (query) in limbajul de interogare** – lista cuvinte, cuvinte adiacente, op logici, op nelogici (near)
- **Multimea rezultat** – multimea de documente relevante pentru query
- **Prezentarea rezultatelor** – lista ordonata, grafic, etc.



Regasirea informatiei

Primele sisteme IR – **Boolean keyword model**

- Fiecare cuvant din text tratat ca un flag
- Limbajul de interogare – expresii logice peste cuvinte
- Dezavantaje – o singura masura, greu de specificat query

Sisteme actuale - **Functie scor IR**

- Modele statistice bazate pe contoare de cuvinte
- Functia **BM25** (proiectul open source Lucene)
- **BM25**
 - **Term Frequency (TF)** – frecventa cu care apare un cuvant din query in document
 - **Inverse Document frequency (IDF)** – ex "in"
 - **Lungimea documentului** – doc mai scurte cu toate cuvintele din query mai bune

Regasirea informatiei

TF(q_i, d_j) pt **N** documente – nr de aparitii q_i in documentul d_j

DF(q) – **Document frequency counts** – nr de documente care contin cuvantul q_i

Fiind dat un document d_j si un query cu cuvintele $q_{1:N}$ avem

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^N IDF(q_i) * \frac{TF(q_i, d_j) * (k + 1)}{TF(q_i, d_j) + k(1 - b + b \frac{|d_j|}{L})}$$

$|d_j|$ - nr cuvinte din documentul d_j

$L = \sum_i |d_i| / N$ – lungimea medie a documentelor din corpus

K, b – determinati prin validare incrucisata, valori tipice:
 $k=2.0, b=0.75$

$$IDF(q_i) = \log \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5}$$



Regasirea informatiei

- Dificil de aplicat BM25 fiecarui document din corpus
- **Hit list** = index creat anterior care refera pentru fiecare cuvânt din vocabular documentele ce contin acel cuvânt
- Pt un query, se face intersectia între hit list și cuvintele din query și se face cautarea numai pe aceasta intersectie
- BM25 – model care trateaza cuvintele ca fiind independente
- **Imbunatatiri**
 - Corelatii
 - Cuvinte derivate, omonime



Evaluarea sistemelor IR

- **Precision** **Recall**

- 100 documente cu 1 query pt care obtinem o multime de 40

	In multime	Nu in multime
Relevant	30	20
Nerelevant	10	40

- **Precision** = $30/(30+10) = 0.75$ – procentul de documente relevante din multimea obtinuta
- **Recall** = $30/(30+20) = 0.6$ – procentul de documente relevante din colectie care sunt in setul rezultat
- Recall este mai greu de calculat
- Se pot combina $2PR/(P+R)$

Regasirea informatiei

- Page Rank (Brin and Larry Page 1998 Google)
- Paginile cu multe in-links au scor mare
- Dar se pondereaza cu links catre situri "de calitate"

$$PR(p) = \frac{1-d}{N} + d \sum_i \frac{PR(in_i)}{C(in_i)}$$

- PR(p) – rangul paginii p
- N – nr total de pagini in corpus
- in_i – paginile care au link la p
- $C(in_i)$ – nr de out-links in pagina in_i
- d – damping factor – probabilitatea ca sa ramana pe aceeaasi pagina
- Calculat iterativ – se incepe cu paginile cu PR(p=1 si itereaza actualizand rangurile pana la convergenta



1.4 Extragerea informatiei

- **Scop:** achizitia cunostintelor prin analiza unui text cu focalizare pe aparitia unei clase particulare de obiecte si a relatiilor dintre aceste obiecte
- Exemple tipice
 - extragerea din pagini web a adreselor (cu campuri strada, nr, etc.)
 - meteo – temp, vat, precipitatii, etc.



Extragerea informatiei

■ **Atomate finite**

Templates

- Extrage informatii relevante unui obiect – valorile unor attribute predefinite
- Se defineste un template (pattern) pentru fiecare atribut de extras
- Template-ul este definit de un automat finit (expresii regulate)
- Template – prefix regex, target regex, postfix regex
- price `[$][0-9]*([.][0-9][0-9])?`



Extragerea informatiei

- Daca template-ul se potriveste 1 dat – extrage target regex
- Daca nu se potriveste de loc – atribut lipsa
- Daca se potriveste de mai multe ori – prioritate, mai multe versiuni de template (prefix regex de ex)
- Intereseaza **Recall**



Extragerea informatiei – ontologii din corpusuri mari

- Construirea unei ontologii dintr-un corpus
- Caracteristici:
 - Corpus de mare dimensiune
 - Domeniu larg
 - Rezultate agregate din mai multe surse
 - Intereseaza **Precision** (nu Recall)

Extragerea informatiei – ontologii din corpusuri mari

Templates predefinite

- Invatarea unei ontologii – categorii si sub-categorii de concepte, dintr-un corpus mare
- Templates generale si cu precizie mare (sunt aproape intotdeauna corecte cand se potrivesc) dar au recall mic (nu se potrivesc pe tot ce este relevant)
- **NP such as NP (, NP)* (,)? ((and|or) NP)?**
- "diseases such as measles affect your child"
- "supports network protocol such as DNS"
- "measles is a disease"
- "she is a little tired"



Extragerea informatiei – ontologii din corpusuri mari

Constructia automata a templates

- Relatia de subsumare este importanta deci templates ot fi construite manual
- Dar daca dorim si alte relatii?
- Se pot genera templates pornind de la exemple
- ("Aut1" "Titlu1")
- ("Aut2" "Titlu 2")
- Se cauta si rezulta **N** potriviri
- Fiecare potrivire (match) este definita
- (Autor, Titlu, Ordine, Prefix, Mijloc, URL)



Extragerea informatiei – ontologii din corpusuri mari

- **(Autor, Titlu, Ordine, Prefix, Mijloc, URL)**
- Ordine = t daca autorul apare intai
- Mijloc = caracterele intre Autor si Titlu
- Prefix = 10 caractere inainte de match
- Sufix = 10 caractere dupa match
- URL = adresa web unde s-a gasit matchPe baza acestoar se pot genera templates
- Autor si Titlu – regex cu orice caracter, cu primul si ultimul litere
- Prefix, Mijloc, Postfix – siruri de caractere
- Fiecare Mijloc distinct genereaza un Template
- Prefix = cel mai lung sufix comun al tuturor prefixelor din match
- Postfix = cel mai lung prefix comun al tuturor sufixelor din match



2 – Prelucrare LN pentru comunicare



2.1 Comunicare

- *Definitie:* schimbul intentional de informatie generat de producerea si perceperea semnelor dintr-un sistem partajat de semne conventionale

- Componentele comunicarii
 - intentie
 - generare
 - sinteza

 - perceptie
 - analiza
 - desambiguare
 - incorporare



Acte de comunicare

J. Austin - How to do things with words, 1962, J. Searle - Speech acts, 1969

Un act de comunicare:

- **locutie** = fraza spusă de locutor
- **illocutie** = înțelesul dorit spre a fi comunicat de locutor (performativă)
- **prelocutie** = acțiunea care rezultă din locutie

Maria i-a spus lui Ion: "Te rog închide ușa"

locutie

illocutie continut

prelocutie: **ușa închisă**

Categoriile ilocutionale

- **Asertive**
- **Directive**
- **Comisive**
- **Permisive**
- **Prohibitive**
- **Declarative**
- **Expresive**



2.2 Modele limbaj

- Modelele n-gram sunt bazate pe secvențe de caractere, cuvinte
- Problema: complexitate = 10^5 cuvinte în vocabular
 $\Rightarrow 10^{15}$ probabilități de trigramuri de estimat!
- Modele de limbaj bazate pe structura gramaticală
- Categoriile lexicale (parts of speech)" substantiv, adjectiv, etc
- categorii sintactice: grup substantiva, grup verbal, etc.
- Structuri de fraze



Modele limbaj

- Modelele n-gram sunt bazate pe secvente de caractere, cuvinte
- Problema: complexitate = 10^5 cuvinte in vocabular
 $\Rightarrow 10^{15}$ probabilitati de trigramuri de estimat!
- Modele de limbaj bazate pe structura gramaticala
- Categoriile lexicale (parts of speech)" substantiv, adjectiv, etc
- categorii sintactice: grup substantiva, griup verbal, etc.
- Structuri de fraze



Definire limbaj

- Lexicon, categorii deschise si inchise
- Analiza lexicala
- Gramatici
- Analiza sintactica (pars oratoris)
- Terminale, neterminale
- Reguli de rescriere (LHS \rightarrow RHS)

- Analza semantica
- Analiza pragmatica



2.3 Gramatici

- Gramatici independente de context probabilistice (GICP)

VP → Verb [0.70]

| Verb NP [0.30]



Grammatici

- Lexicon

Noun → **breeze** [0.10] | **wumpus** [0.15] | **ball** [0.15] ...

Verb → **is** [0.10] | **see** [0.10] | **smells** [0.10] | **hit** [0.10]...

Adjective → **right** [0.10] | **left** [0.10] | **smelly** [0.15] ...

Adverb → **here** [0.05] | **there** [0.05] | **ahead** [0.02] ...

Pronoun → **me** [0.10] | **you** [0.03] | **I** [0.10] | **it** [0.10] ...

RelPronoun → **that** [0.40] | **who** [0.20] ...

Name → **John** [0.1] | **Mary** [0.01] ...

Article → **the** [0.40] | **a** [0.30] | **an** [0.10] ...

Preposition → **to** [0.20] | **in** [0.10] | **on** [0.05] ...

Conjunction → **and** [0.50] | **or** [0.10] | **but** [0.20]...



Grammatici

■ Sintaxa

S → NP VP	[0.90]	I feel a breeze
S Conjunction S	[0.10]	I feel a breeze and it stinks
NP → Pronoun	[0.30]	I
Name	[0.10]	John
Noun [0.10]	[0.10]	pit
Article Noun	[0.25]	the wumpus
NP PP	[0.10]	the wumpus in 1,3
NP RelClause	[0.05]	the wumpus that is smelly
VP → Verb	[0.40]	stinks
VP NP	[0.35]	feel a breeze
VP Adjective	[0.05]	smells dead
VP PP	[0.10]	is in 1,3
VP Adverb	[0.10]	go ahead
PP → Preposition NP	[1.00]	to the east
RelClause → RelPronoun	VP [1.00]	that is smelly



Top-Down Parsing

"John hit the ball"

1. S
2. $S \rightarrow NP, VP$
3. $S \rightarrow \text{Noun}, VP$
4. $S \rightarrow \text{John}, \text{Verb}, NP$
5. $S \rightarrow \text{John}, \text{hit}, NP$
6. $S \rightarrow \text{John}, \text{hit}, \text{Article}, \text{Noun}$
7. $S \rightarrow \text{John}, \text{hit}, \text{the}, \text{Noun}$
8. $S \rightarrow \text{John}, \text{hit}, \text{the}, \text{ball}$



Bottom-Up Parsing

1. John, hit, the, ball
2. Noun, hit, the, ball
3. Noun, Verb, the, ball
4. Noun, Verb, Article, ball
5. Noun, Verb, Article, Noun
6. NP, Verb, Article, Noun
7. NP, Verb, NP
8. NP, VP
9. S



Analiza sintactica

- **Eficienta** – dupa ce analizam un subsir, se memoreaza rezultatul – **chart** – **chart parser**
- GIC – orice subsir/fraza analizata pe o ramura a ad poate fi utilizata pe alta ramura
- Chart parser-ul **CYK** (J. Cocke, D. Young, T. Kasami)
- Gramatica trebuie sa fie in forma normala Chomsky
- $X \rightarrow \text{cuvant}$ (reguli lexicale)
- $X \rightarrow Y Z$ (reguli sintactice)



Analiza sintactica

- CYK foloseste un spatiu de $O(n^2m)$, n – nr cuvinte, m – nr neterminale, pentru a construi o tabela de probabilitati P
- Timp $O(n^3m)$
- Nu examineaza toti ad posibili ci calculeaza probabilitatea celui mai probabil ad
- Toti subarborii sunt implicut reprezentati in tabela P de unde se pot obtine daca dorim

algoritm CYK(cuvinte, gramatica) **intoarce P**, o tabela de probabilitati

$N \leftarrow$ Lungime (cuvinte)

$M \leftarrow$ nr de neterminale in gramatica

$P \leftarrow$ matrice de $M \times N \times N$, initial 0

/* insereaza reguli lexicale pt fiecare cuvint */

pentru $i=1$ la N **repeta**

pentru fiecare regula de forma $(X \rightarrow \text{cuvant}_i, [p])$ **repeta**

$P[X, i, 1] \leftarrow p$

/* combina primul si al doilea net din RHS a regulilor sintactice, incepand cu cele mai scurte */

pentru lung = 2 la N **repeta**

pentru start=1 la $N-\text{lung}+1$ **repeta**

pentru lung1=1 la lung-1 **repeta**

$\text{len2} \leftarrow \text{len}-\text{len1}$

pentru fiecare regula de forma $(X \rightarrow Y Z [p])$ **repeta**

$P[X, \text{start}, \text{len1}] \leftarrow \text{MAX}(P[X, \text{start}, \text{len}],$

$P[Y, \text{start}, \text{len1}] \times P[Z, \text{start}+\text{len1}, \text{len2}] \times p)$

intoarce P

sfarsit



Analiza semantica

- GICP ai sa arate ca anumite categorii gramaticale sunt mai probabile decat altele
- Dra probabilitatile depind si de relatia intre cuvinte
- GICP imbogatite

$VP(v) \rightarrow Verb(v) NP(n) \quad [P1(v,n0)]$

$VP(v) \rightarrow Verb(v) \quad [P2(v)]$

$NP(n) \rightarrow Article(a) Adj(j) Noun(n) \quad [P3(n,a)]$

$Noun(dog) \rightarrow dog \quad [pn]$



2.4 Definite Clause Grammar (DCG)

- Gramatici BNF - probleme
- Utilizare LP
- Gramatici cu clauze definite (DCG)
- Fiecare regula din gramatica poate fi vazuta ca o regula din DCG
- Fiecare categorie sintactica se reprezinta printr-un predicat cu un argument sir



Definite Clause Grammar (DCG)

- NP(s) este adevarat daca s este NP

$S \rightarrow NP VP$ devine

$NP(S1) \wedge VP(s2) \Rightarrow S(\text{append}(s1,s2))$

- Parsing = inferenta logica
- Bottom-up parsing – forward chaining
- Top-down parsing – backward chaining
- Aceeasi gramatica utilizata atat pentru analiza cat si pentru generare



In BNF

$S \rightarrow NP VP$

In LP / DGC

$NP(s_1) \wedge VP(s_2) \Rightarrow S(\text{Append}(s_1, s_2))$

BNF

$\text{Noun} \rightarrow \text{ball} \mid \text{book}$

In LP / DGC

$(s = \text{“ball”} \vee s = \text{“book”}) \Rightarrow \text{Noun}(s)$

BNF, DCG, Prolog

BNF	FOPL/DCG	PROLOG
<p> $S \rightarrow NP \ VP$ $NP \rightarrow \text{Noun}$ $\text{Noun} \rightarrow \text{stench}$ $\text{Noun} \rightarrow \text{wumpus}$ $VP \rightarrow \text{Verb}$ $\text{Verb} \rightarrow \text{smells}$ $\text{Verb} \rightarrow \text{kills}$ </p>	<p> $NP(s1) \wedge VP(s2) \Rightarrow S(\text{append}(s1,s2))$ $\text{Noun}(s) \Rightarrow NP(s)$ $\text{Verb}(s) \Rightarrow VP(s)$ $(s = \text{"stench"} \vee s = \text{"wumpus"}) \Rightarrow$ $\text{Noun}(s)$ $(v = \text{"smells"} \vee v = \text{"kills"}) \Rightarrow$ $\text{Verb}(v)$ </p>	<p> $\text{sentence}([S1, S2]) :-$ $\text{np}(S1), \text{vp}(S2).$ $\text{np}(S):- \text{noun}(S).$ $\text{vp}(S):- \text{verb}(S).$ $\text{noun}(\text{stench}).$ $\text{noun}(\text{wumpus}).$ $\text{verb}(\text{smells}).$ $\text{verb}(\text{kills}).$ $?- \text{sentence}([\text{wumpus}, \text{smells}]).$ $?- \text{sentence}([S1, S2]).$ </p>



Imbogatire DCG

- Imbogatesc neterminale cu argumente suplimentare
- Verifica corectitudinea gramaticala
- Ataseaza semantica
- Adauga expresii / functii care se testeaza



Argument pt semantica

DCG	FOPL	PROLOG
$S(\text{sem}) \rightarrow NP(\text{sem1}) VP(\text{sem2})$ {compose(sem1, sem2, sem)}	$NP(s1, \text{sem1}) \wedge VP(s2, \text{sem2}) \Rightarrow$ $S(\text{append}(s1, s2)),$ compose(sem1, sem2, sem)	slide urmator

semantica compozitionala

The dog has legs.

(caine *parti* picioare)

The ball is yellow.

(minge *proprietate* galbena)

The ball is red.

(mine *proprietate* rosie)

The dog bites.

(caine *actiune* musca)

sentence(S, Sem) :- np(S1, Sem1), vp(S2, Sem2), append(S1, S2, S),
Sem = [Sem1 | Sem2].

np([S1, S2], Sem) :- article(S1), noun(S2, Sem).

vp([S], Sem) :- verb(S, Sem1), Sem = [actiune, Sem1].

vp([S1, S2], Sem) :- verb(S1, _), adjective(S2, Sem1), Sem = [proprietate, Sem1].

vp([S1, S2], Sem) :- verb(S1, _), noun(S2, Sem1), Sem = [parti, Sem1].

noun(dog,caine).

noun(ball,ball).

noun(legs,picioare).

verb(bytes,musca).

verb(is,este).

verb(has,are).

adjective(yellow,galbena).

adjective(red,rosie).

?- sentence([the,ball,is,yellow],Sem).

Sem = [minge, proprietate, galbena]

Yes

?- sentence([the,dog,bytes],Sem).

Sem = [caine, actiune, musca]

Yes

?- sentence([is,dog,bytes],Sem).

No

?- sentence([the,dog,has,legs],Sem).

Sem = [caine, parti, picioare]

Yes



Verificare corectitudine gramaticala

- cazuri
 - subcategorii verbe: complementul pe care il poate accepta un verb
 - acord subiect predicat
 - etc.
-
- Parametrizarea neterminalelor

Cazuri

Nominativ (subjective)

I take the bus

You take the bus

He takes the bus

Eu iau autobuzul

Tu iei autobuzul

El ia autobuzul

Acuzativ (objective)

He gives me the book

Imi da cartea

S → NP(Subjective) VP

NP(case) → Pronoun (case) | Noun | Article Noun

// I

VP → VP NP(Objective)

//

believe him

VP → VP PP

// turn to the right

VP → VP Adjective

VP → Verb

PP → Preposition NP(Objective)

Pronoun(Subjective) → I | you | he | she

Pronoun(Objective) → me | you | him | her

sentence(S) :- np(S1,subjective), vp(S2),
append(S1, S2, S).

np([S], Case) :- pronoun(S, Case).

np([S], _) :- noun(S).

np([S1, S2], _) :- article(S1), noun(S2).

pronoun(i, subjective).

pronoun(you, _).

pronoun(he, subjective).

pronoun(she, subjective).

pronoun(me, objective).

pronoun(him, objective).

pronoun(her, objective).

noun(ball).

noun(stick).

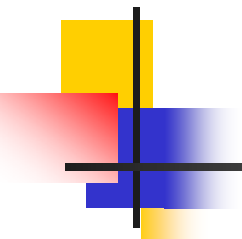
article(a).

article(the).



Subcategorii verbe

- Lista de subcategorii: ce complemente accepta verbul
- depinde de verb
- $S \rightarrow NP(\text{Subjective}) VP(\text{subcat})$
dar cazuri in care nu depinde
- $VP(\text{subcat}) \rightarrow VP(\text{subcat}) PP$
- $\quad \quad \quad | VP(\text{subcat}) \text{ Adverb}$
smell the wumpus now



VP(subcat) → { subcat = np } VP(np) NP(Objective)
 | { subcat = adj } VP(adj) Adjective
 | { subcat = pp } VP (pp) PP
 | Verb

smell	[NP]	smell a wumpus
	[Adjective]	smell awful
	[PP]	smell like a wumpus

is	[Adjective]	is smelly
	[PP]	is in box
	[NP]	is a pit

give	[NP, PP]	give the gold in box to me
	[NP, NP]	give me the gold

died	[]	died
------	----	------



$S \rightarrow \text{NP(Subjective)} \text{ VP(subcat)}$

$\text{NP(case)} \rightarrow \text{Pronoun (case)} \mid \text{Noun} \mid \text{Article Noun}$

$\text{Pronoun(Subjective)} \rightarrow \text{I} \mid \text{you} \mid \text{he} \mid \text{she}$

$\text{Pronoun(Objective)} \rightarrow \text{me} \mid \text{you} \mid \text{him} \mid \text{her}$

$\text{VP(subcat)} \rightarrow$

- $\{ \text{subcat} = \text{np} \} \text{ VP(np)} \text{ NP(Objective)}$
- $\mid \{ \text{subcat} = \text{adj} \} \text{ VP(adj)} \text{ Adjective}$
- $\mid \{ \text{subcat} = \text{pp} \} \text{ VP (pp)} \text{ PP}$
- $\mid \text{Verb}$
- $\mid \text{VP(subcat)} \text{ PP}$
- $\mid \text{VP(subcat)} \text{ Adverb}$

VP(subcat) → {subcat = np} VP(np) NP(Objective)
| {subcat = adj} VP(adj) Adjective
| {subcat = pp} VP (pp) PP
| Verb
| **VP(subcat) PP**
| VP(subcat) Adverb

sentence(S) :- np(S1,subjective), vp(S2, Subcat),
append(S1, S2, S).

VP(subcat) → VP(subcat) ... !!!

**vp(S, Subcat) :- Subcat = np, vp1(S1, np),np(S2, objective),
append(S1, S2, S).**

vp(S,Subcat) :- vp1(S1, Subcat), pp(S2), append(S1, S2, S).

vp1([S],np):-verb(S).

verb(give).

verb(make).



2.5 Analiza pragmatica

- Analiza semantica
- Desambiguare
- Interpretare pragmatica – utilizare si efect asupra ascultatorului
- Indexical – refera situatia curenta
- Anafora – refera obiecte deja mentionate



2.6 Ambiguitate

- Lexicala – același cuvânt diverse înțelesuri
- Sintactica – arbori diferiți de analiză
- Referențială – referire la obiecte anterioare
- Pragmatică – referire la loc, timp