

Processament del Llenguatge Natural

lloc: [Institut d'Ensenyaments a Distància de les Illes
Balears](#)
Curs: Models d'intel·ligència artificial
Llibre: Processament del Llenguatge Natural

Imprès per: Carlos Sanchez Recio
Data: dilluns, 3 de febrer 2025, 22:28

Taula de continguts

1. NLP als lliuraments de Models d'IA

2. Processament del Llenguatge Natural

3. Models de llenguatge

3.1. Bag of words

3.2. Models N-gram de paraules

3.3. Altres models n-gram

3.4. Suavitzat de models n-gram

3.5. Representacions de paraules

3.6. Part-of-speech tagging

3.7. Comparació dels models de llenguatge

4. Gramàtica

4.1. Lèxic

5. Parsing

5.1. Dependency parsing

5.2. Aprenentatge d'un parser a partir d'exemples

6. Gramàtiques augmentades

6.1. Complicacions del llenguatge natural real

7. Tasques del llenguatge natural

8. Llibreries Python per a NLP

9. Repositori AIMA

10. Resum

1. NLP als lliuraments de Models d'IA

D'ençà del 2022, hi ha hagut una gran popularització del processament del llenguatge, sobretot arran de l'expansió dels models estesos del llenguatge, després de la publicació de l'article pivotal dels **transformers**, *Attention Is All You Need*.

Per això, dedicarem dos lliuraments al **processament del llenguatge natural** (*Natural Language Processing*, **NLP**), una quarta part del mòdul de Models d'Intelligència Artificial. Segurament, es tracta de l'àrea de la intel·ligència artificial amb un desenvolupament més intens i un impacte social i laboral més gran ara mateix.

En aquest cinquè lliurament del curs, definirem les tasques que entren dins aquesta àrea de la intel·ligència artificial, i veurem els models anteriors a les xarxes neuronals, principalment **Naive Bayes**, **n-grams** i **gramàtiques**. Aquest lliurament es basarà en el capítol 24 (*Natural Language Processing*) del llibre Artificial Intelligence, a Modern Approach (AIMA), de Russell i Norvig. També farem referència a alguns exemples de codi que hi ha disponibles al [repositori del llibre](#), i a una selecció de llibreries Python rellevants en NLP: [NLTK](#), [spaCy](#) i [Stanza](#).

Al sisè lliurament, ja hauré introduït les **xarxes neuronals artificials** al cinquè lliurament de Sistemes d'Aprenentatge Automàtic. Aleshores serà el moment de tractar l'aplicació de l'**aprenentatge profund** (*Deep Learning*) al processament del llenguatge natural. Aquest segon lliurament d'NLP combinarà el capítol 26 (*Deep Learning for Natural Language Processing*) del llibre AIMA amb diversos [exemples de Keras](#) i altres fonts d'informació de desenvolupaments recents que encara no estan recollits a la darrera edició del llibre, del 2022.

- <https://github.com/aimacode>
- <https://www.nltk.org/>
- [spaCy](#)
- [Stanza](#)
- <https://keras.io/examples/nlp/>

2. Processament del Llenguatge Natural

Fa devers 100000 anys els humans vàrem aprendre a parlar, i en fa devers 5000 que aprenguérem a escriure. La complexitat i la diversitat del llenguatge humà separa l'Homo Sapiens de la resta d'espècies animals. És clar que hi ha d'altres atributs que són igualment humans: cap altra espècie no du roba, crea art ni passa hores cada dia a les xarxes socials com feim els humans. Però quan Alan Turing proposà el seu test per a la intel·ligència, el va basar en el llenguatge, no en cap altra habilitat, per ventura pel seu caràcter universal i perquè el llenguatge captura una gran part del comportament intel·ligent: un parlant (o escriptor) té l'**objectiu** de comunicar qualche **coneixement**, aleshores **planifica** algunes frases que representen aquell coneixement, i **actua** per aconseguir l'objectiu. L'oient (o lector) **percep** el llenguatge, i **infereix** el significat que s'hi representa.

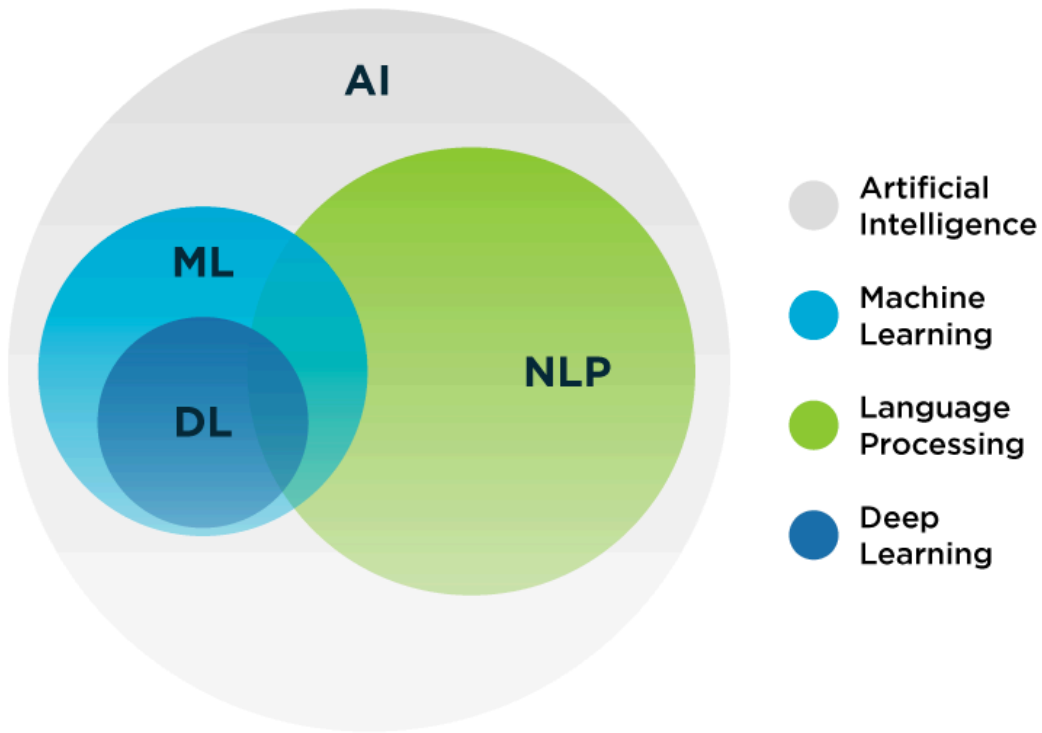
Aquest tipus de comunicació a través del llenguatge ha permès créixer a la civilització, és la nostra forma principal de transmetre el coneixement cultural, legal, científic i tecnològic. Hi ha tres raons principals perquè les computadores facin Processament del Llenguatge Natural (NLP, *Natural Language Processing*):

- Per **comunicar**-se amb els humans. En moltes situacions convé que els humans usin llenguatge per comunicar-se amb les màquines, i sovint convé més usar llenguatge natural que no un llenguatge formal.
- Per **aprendre**. Els humans hem escrit una gran quantitat de coneixement usant llenguatge natural. Només la Wikipedia té trenta milions de pàgines de fets, mentre que gairebé no hi ha fonts de fets escrites amb llenguatge lògic. Si volem que el nostre sistema sàpiga moltes coses, ha d'entendre el llenguatge natural.
- Per avançar en la **comprensió científica** de les llengües i el seu ús, usant les eines de la intel·ligència artificial en combinació amb la lingüística, la psicologia cognitiva i la neurociència. En aquest lliurament explorem diversos models matemàtics per al llenguatge, i les tasques que es poden realitzar usant-los.

El processament del llenguatge natural és la branca de la intel·ligència artificial o IA que s'encarrega de donar als ordinadors la capacitat d'entendre text i paraules parlades de la mateixa manera que ho poden fer els éssers humans.

L'NLP combina la lingüística computacional (modelatge basat en regles del llenguatge humà) amb models estadístics, d'aprenentatge automàtic i d'aprenentatge profund. En conjunt, aquestes tecnologies permeten als ordinadors processar el llenguatge humà en forma de text o dades de veu i "entendre" el seu significat complet, amb la intenció i el sentiment del parlant o escriptor.

L'NLP és al cor dels programes informàtics que tradueixen text d'un idioma a un altre, responen a ordres parlades i resumeixen grans volums de text ràpidament, fins i tot en temps real. És possible que hàgiu interactuat amb NLP en forma de sistemes GPS operats per veu, assistents digitals, programari de dictat de veu a text, chatbots d'atenció al client d'altres serveis per al consumidor. Però la NLP també juga un paper creixent en les solucions empresarials que ajuden a simplificar les operacions corporatives, augmentar la productivitat dels empleats i simplificar els processos crítics.



Dins el camp de la intel·ligència artificial, el processament del llenguatge natural combina tècniques de [sistemes experts](#) amb d'altres d'aprenentatge automàtic.

3. Models de llenguatge

Els llenguatges formals estan definits d'una forma precisa. Una **gramàtica** defineix la sintaxi de les oracions correctes i unes **regles semàntiques** en defineixen el significat.

En canvi, els llenguatges naturals, com el català, l'espanyol o l'anglès, no es poden caracteritzar d'una forma tan precisa, per les raons següents.

- Les valoracions lingüístiques varien de persona a persona i en cada època. Tothom està d'acord en la correcció de determinades oracions, però d'altres poden originar discrepàncies.
- El llenguatge natural és ambigu.
- El mapejat de símbols a objectes no està definit formalment. En el llenguatge natural, dues ocurrences de la mateixa paraula (per exemple, *això*) es poden referir a diferents objectes del món real.

Si no podem fer una distinció precisa entre gramaticalitat o no gramaticalitat d'una oració, almanco podem estimar la seva probabilitat.

Es defineix un **model del llenguatge** com una distribució de probabilitat que descriu la versemblança (likelihood) de qualsevol cadena de paraules. Per exemple, en anglès la cadena "Do I dare disturb the universe?" té una probabilitat raonable com a frase, mentre que "Universe dare the I disturb to" és molt més improbable.

Amb un model de llenguatge podem predir quines paraules són probables a continuació d'un text i, per tant, suggerir complecions per a un correu electrònic o un missatge de text. Podem calcular quines alteracions en un text el farien més probable, per suggerir correccions d'ortografia o gramàtica. Amb un parell de models, podem calcular la traducció més probable d'una oració. Amb diversos parells pregunta/resposta com a dades d'entrenament, podem calcular la resposta més probable a una pregunta. Per tant, els models de llenguatge són al centre d'un gran ventall de tasques de llenguatge natural. La tasca mateixa de modelitzar el llenguatge serveix com a mesura del progrés en la comprensió del llenguatge.

Els llenguatges naturals són complexos, de forma que qualsevol model del llenguatge serà, com a màxim, una aproximació. A continuació veurem diversos models senzills del llenguatge que són clarament incorrectes, però tot i així resulten útils per a determinades tasques.

3.1. Bag of words

El model més simple possible té en compte tan sols la probabilitat de cada paraula, sense considerar les paraules que l'envolten. Això és un model naïve de Bayes, i permet classificar les oracions per temes, per exemple les frases següents són sobre economia la primera i sobre el temps la segona.

1. Als mercats cotitzats, dilluns les accions varen pujar, amb els principals índexs guanyant un 1% a causa d'un optimisme sostingut després dels beneficis del primer trimestre.
2. Hi ha hagut pluges intenses a tota la costa est dilluns, amb avisos d'inundacions a Nova York i d'altres zones.

Veurem a continuació el model naïve de Bayes, com a model de llenguatge complet. Això significa que no ens basta saber quina és la categoria més probable d'una oració, sinó que volem una probabilitat de distribució conjunta sobre totes les oracions i categories. Això suggereix que hauríem de considerar totes les paraules de cada frase. A partir d'una frase formada per les paraules w_1, w_2, \dots, w_n , que podem escriure de forma compacta $w_{1:N}$ la fórmula naïve de Bayes ens dona:

$$P(\text{classe} | w_{1:N}) = \alpha P(\text{classe}) \prod_j P(w_j | \text{classe})$$

L'aplicació del model naïve de Bayes a cadenes de paraules s'anomena el model **bag-of-words**, bossa o sac de paraules. És un model generatiu que descriu un procés per generar una oració. Imaginem que per a cada categoria (economia, temps...) tenim una bossa plena de paraules, per exemple escrites cada una en un paper. Per generar text, en primer lloc se selecciona una bossa i es descarten les altres. D'aquella bossa, es van treient papers (amb reposició) i se'n forma una frase fins que surt el final de frase, per exemple un punt.

Està clar que aquest model és erroni: s'assumeix equivocadament que cada paraula és independent de les altres, i per tant no genera frases coherents. Però permet una classificació amb bona precisió: les paraules "mercats" i "beneficis" són bones pistes de la categoria d'economia i "pluja" i "inundacions" suggereixen la secció del temps.

Es poden aprendre les probabilitats necessàries mitjançant aprenentatge supervisat sobre un corpus de text, amb cada segment de text etiquetat amb una classe. Un corpus sol tenir com a mínim un milió de paraules, i almenys desenes de milers de paraules al vocabulari. La mida dels corpus utilitzats creix contínuament.

A partir del corpus s'estima la probabilitat de cada categoria, $P(\text{classe})$, i la probabilitat de cada paraula dins la categoria $P(w_j | \text{classe})$. Aquesta estimació per recompte funciona prou bé quan els recomptes són prou grans (i la variància baixa) però veurem un mètode millor per estimar les probabilitats quan els recomptes són baixos, el suavitzat (*smoothing*).

3.2. Models N-gram de paraules

El model bag-of-words té limitacions. Per exemple, en anglès la paraula *quarter* és freqüent tant en economia com en esports. Però la frase *first quarter earnings report* és freqüent només en economia i la frase *fourth quarter touchdown passes* només en esports.

Es podria manipular el model *bag-of-words* per tractar frases com *first quarter earning report* com si fossin una sola paraula, però una solució més sòlida és introduir un nou model. Podem començar fent que la probabilitat d'una frase depengui de totes les paraules anteriors de l'oració:

$$P(w_{1:N}) = \prod_{j=1}^N P(w_j | w_{1:j-1})$$

Aquest model és perfectament correcte en el sentit que captura totes les possibles interaccions entre paraules, però no és pràctic: amb un vocabulari de 100000 paraules (10^5) i una longitud de cadena de 40 s'haurien d'estimar 10^{200} paràmetres ($10^{5 \cdot 40}$). Es pot trobar una solució de compromís amb un model de **cadena de Markov** que considera només la dependència de les n paraules anteriors. Això es coneix com a model n-gram: una seqüència de n símbols és un n-gram, amb els casos especials **unigram** (1-gram o bag-of-words), **bigram** (2-gram) i **trigram** (3-gram). En un model n-gram, la probabilitat de cada paraula només depèn de les $n - 1$ paraules anteriors.

$$P(w_j | w_{1:j-1}) = P(w_j | w_{j-n+1:j-1})$$

$$P(w_{1:N}) = \prod_{j=1}^N P(w_j | w_{j-n+1:j-1})$$

Els models n-gram funcionen bé per classificar seccions de premsa, altres tasques de classificació com la detecció de spam, anàlisi de sentiment (si una opinió és positiva o negativa) o atribució d'autoria (cada autor té un estil i vocabulari diferent).

3.3. Altres models n-gram

Una alternativa als models n-gram de paraula són els models a nivell de caràcter, en què la probabilitat de cada caràcter depèn dels $n - 1$ caràcters anteriors. Aquest enfocament és útil quan s'ha de tractar amb paraules de fora del vocabulari, o compostes.

Els models de caràcter s'adapten molt bé a la tasca d'identificació del llenguatge: donat un text, decidir en quina llengua està escrit. Fins i tot amb missatges breus com "Hello world" o "Wie geht's dir", els models de caràcter poden identificar correctament el primer com a anglès i el segon com a alemany, amb una precisió en general d'un 99%. Els models de caràcter són bons en certes tasques de classificació, com dir que "dextroamfetamina" és un nom de fàrmac, "Kallenberger" un nom de persona i "Plattsburg" un nom de ciutat, fins i tot si el model no ha vist aquestes paraules abans.

Tenim una implementació de classificació de llenguatge, entre anglès i alemany, en el quadern següent.

https://colab.research.google.com/drive/1_agr7BFRys0JYyZPTKYCsu64dcmfzpJ6?usp=sharing

3.4. Suavitzat de models n-gram

Els n-gram d'alta freqüència com "of the" tenen recomptes elevats en el corpus d'entrenament, de forma que l'estimació de la seva probabilitat serà precisa. Amb un corpus diferent la probabilitat estimada seria semblant. En canvi, els n-grams de baixa freqüència tenen recomptes baixos que estaran subjectes a soroll aleatori (tenen variància elevada). Els models funcionaran millor si podem suavitzar aquesta variància.

A més, sempre hi ha la possibilitat que calgui avaluar un text que contingui una paraula desconeguda o fora del vocabulari, una que no ha aparegut al corpus d'entrenament. Però seria un error donar a aquesta paraula la probabilitat de zero: tota la frase tendria probabilitat zero.

Una manera de modelitzar les paraules desconegudes és modificar el corpus d'entrenament substituint les paraules infreqüents per un símbol especial, tradicionalment <UNK> (de *unknown*, desconegut). Es podria decidir mantenir, per exemple, les 50000 paraules més freqüents, o totes les paraules amb una freqüència més gran que 0.0001%, i substituir les altres per <UNK>.

Fins i tot després d'haver tractat les paraules desconegudes, hi pot haver seqüències de paraules no observades. El mètode més simple de suavitzar el proposà Laplace el segle XVIII. A partir de les dades observades, la probabilitat que el sol no surti demà és zero. Però si suposam una probabilitat a priori uniforme (tan probable és que surti com que no surti), una estimació més bona és $1/(N + 2)$. El 2 correspon als casos que surti i que no surti, i l'1 correspon al cas favorable (en el sentit que es produeix l'esdeveniment, que no surti). El **suavitzat de Laplace**, anomenat també *add-one*, és una passa en la direcció adequada, però en moltes aplicacions de processament del llenguatge no funciona prou bé.

Una altra opció és un **model de backoff**, en què es comença estimant els recomptes de n-grams, però per a cada seqüència de probabilitat zero o baixa, es davalla (*back off*) a considerar (n-1)-grams.

El **suavitzat d'interpolació lineal** és un model de backoff que combina els models trigram, bigram i unigram mitjançant interpolació lineal. L'estimació de la probabilitat es defineix de la forma següent.

$$P_{est}(c_i | c_{i-2:i-1}) = \lambda_3 P(c_i | c_{i-2:i-1}) + \lambda_2 P(c_i | c_{i-1}) + \lambda_1 P(c_i),$$

on $\lambda_3 + \lambda_2 + \lambda_1 = 1$. Els paràmetres λ poden ser fixos o es poden entrenar amb un algorisme d'optimització.

3.5. Representacions de paraules

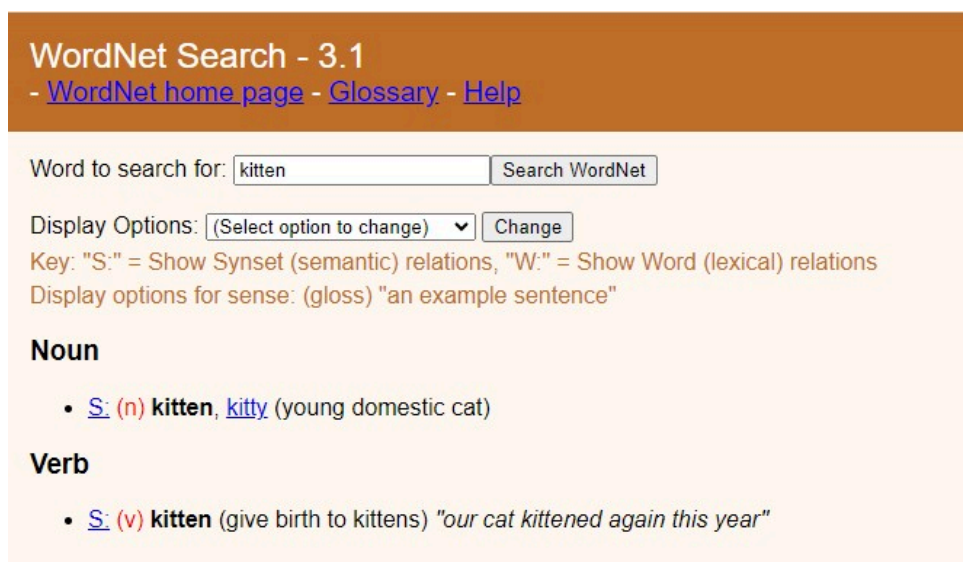
Les n-grams ens poden donar un model que prediu amb precisió la probabilitat de les seqüències de paraules. Tot el que el model sap, ho ha après a partir dels recomptes de seqüències de paraules específiques.

Però un parlant nadiu d'anglès dirà, per exemple, que *a black cat* és una cadena vàlida perquè segueix el patró article-adjectiu-nom, mentre que *cat black a no*.

Ara, després del sintagma *a black cat*, considerem *a fulvius kitten*. Un parlant d'anglès reconeixerà que aquí també se segueix el patró article-adjectiu-nom. Encara que no conegui el significat de *fulvius*, probablement assignarà aquesta paraula acabada en *-ous* a la categoria dels adjectius. Per això, l'aparició de *a black cat* a les dades és una evidència que *a fulvius kitten* també és anglès vàlid, per generalització.

El model n-gram no té aquesta generalització perquè és un model atòmic, cada paraula és un àtom, distinta de qualsevol altra paraula, sense cap estructura interna. Els models factoritzats, o estructurats, tenen una potència expressiva més gran i una generalització més bona. Veurem al proper lliurament com el model factoritzat anomenat **word embedding** té una bona capacitat de generalitzar.

Un tipus de model de paraules estructurat és un **diccionari**, normalment construït manualment. Per exemple, [WordNet](#) és un diccionari de codi obert, curat a mà en format llegible per màquina que ha demostrat la seva utilitat en moltes aplicacions de llenguatge natural. A continuació tenim l'entrada de WordNet per a *kitten*.



The image shows a screenshot of the WordNet Search interface. At the top, there's a header "WordNet Search - 3.1" with links to "WordNet home page", "Glossary", and "Help". Below this, there's a search bar with the text "Word to search for: kitten" and a "Search WordNet" button. Underneath the search bar, there are "Display Options" with a dropdown menu set to "(Select option to change)" and a "Change" button. Below the options, there's a key: "Key: 'S:' = Show Synset (semantic) relations, 'W:' = Show Word (lexical) relations". Then, it says "Display options for sense: (gloss) 'an example sentence'". The main content is divided into two sections: "Noun" and "Verb". Under "Noun", there's a bullet point: "• [S: \(n\)](#) kitten, [kitty](#) (young domestic cat)". Under "Verb", there's a bullet point: "• [S: \(v\)](#) kitten (give birth to kittens) 'our cat kittened again this year'".

Imatge: Entrada de kitten a WordNet

WordNet ajudarà a separar noms i verbs, i obtenir les categories bàsiques, però no explicarà quin és l'aspecte o el comportament de l'animal. WordNet dirà que cues subclasses de cat són *Siamese cat* i *Manx cat*, però no donarà més detalls d'aquestes races.

3.6. Part-of-speech tagging

Una forma de categoritzar paraules és a través de la seva **part of speech** (POS), també coneguda com **categoria lèxica** o **etiqueta**: nom, adjectiu, verb, etcètera. Les categories permeten als models de llenguatge capturar generalitzacions com "els adjectius solen ser davant els noms en anglès". En francès o en català, en general és a la inversa.

Tothom està d'acord que "nom" i "verb" són categories lèxiques, però quan entrem en detalls no hi ha una llista definitiva de categories. A la taula següent veim les 45 etiquetes del Penn Treebank, un corpus de més de tres milions de paraules anotat amb etiquetes de *part of speech*.

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	<i>and, but, or</i>	SYM	Symbol	<i>+, %, &</i>
CD	Cardinal number	<i>one, two, three</i>	TO	"to"	<i>to</i>
DT	Determiner	<i>a, the</i>	UH	Interjection	<i>ah, oops</i>
EX	Existential 'there'	<i>there</i>	VB	Verb, base form	<i>eat</i>
FW	Foreign word	<i>mea culpa</i>	VBD	Verb, past tense	<i>ate</i>
IN	Preposition/sub-conj	<i>of, in, by</i>	VBG	Verb, gerund	<i>eating</i>
JJ	Adjective	<i>yellow</i>	VCN	Verb, past participle	<i>eaten</i>
JJR	Adj., comparative	<i>bigger</i>	VBP	Verb, non-3sg pres	<i>eat</i>
JJS	Adj., superlative	<i>wildest</i>	VBZ	Verb, 3sg pres	<i>eats</i>
LS	List item marker	<i>1, 2, One</i>	WDT	Wh-determiner	<i>which, that</i>
MD	Modal	<i>can, should</i>	WP	Wh-pronoun	<i>what, who</i>
NN	Noun, sing. or mass	<i>llama</i>	WP\$	Possessive wh-	<i>whose</i>
NNS	Noun, plural	<i>llamas</i>	WRB	Wh-adverb	<i>how, where</i>
NNP	Proper noun, singular	<i>IBM</i>	\$	Dollar sign	<i>\$</i>
NNPS	Proper noun, plural	<i>Carolinas</i>	#	Pound sign	<i>#</i>
PDT	Predeterminer	<i>all, both</i>	"	Left quote	<i>(' or ")</i>
POS	Possessive ending	<i>'s</i>	"	Right quote	<i>(' or ")</i>
PP	Personal pronoun	<i>I, you, he</i>	(Left parenthesis	<i>([, (, {, <)</i>
PP\$	Possessive pronoun	<i>your, one's</i>)	Right parenthesis	<i>(],), }, >)</i>
RB	Adverb	<i>quickly, never</i>	,	Comma	<i>,</i>
RBR	Adverb, comparative	<i>faster</i>	.	Sentence-final punc	<i>(. ! ?)</i>
RBS	Adverb, superlative	<i>fastest</i>	:	Mid-sentence punc	<i>(: ; ... - -)</i>
RP	Particle	<i>up, off</i>			

Imatge: Categories del corpus Penn Treebank

El Penn Treebank també anota moltes oracions amb arbres d'anàlisi sintàctica, d'on el corpus pren el seu nom.

La tasca d'assignar una categoria lèxica a cada paraula d'una oració s'anomena en anglès **part of speech tagging**, etiquetatge de categoria lèxica. És un pas inicial útil per a moltes tasques de NLP, com la resposta a preguntes o la traducció. Fins i tot per a una tasca simple com la conversió de text a veu, és útil tenir en compte que en anglès la el nom *record* es pronuncia diferent que el verb *record*.

Un model habitual per a resoldre el *POS tagging* és el model ocult de Markov, en anglès *Hidden Markov Model*. Un model ocult de Markov pren com a entrada una seqüència temporal d'observacions d'evidències i prediu la seqüència més probable d'estats ocults que pot haver-la generada. En l'aplicació de *POS tagging*, les evidències són les paraules observades, $W_{1:N}$ i els estats ocults són les categories lèxiques $C_{1:N}$. N és la longitud de la cadena d'observació, quantes paraules formen la seqüència.

El HMM és un model generatiu que especifica que la forma de produir el llenguatge consisteix a començar en un estat, per exemple IN, l'estat de les preposicions, i fer dues tries: quina paraula (per exemple, *from*) s'hi ha d'emetre, i quin estat (per exemple, DT) ha de venir després. Aquest model no té en compte més context que la categoria actual ni té cap noció del significat que l'oració intenta transmetre. Tot i això, és un model útil. Aplicant l'algorisme de Viterbi per obtenir la seqüència més probable d'estats s'obté una precisió en l'etiquetatge molt alta, habitualment al voltant del 97%.

Podem veure com etiquetar les categories lèxiques amb la llibreria NLTK amb el següent exemple:

<https://www.nltk.org/api/nltk.tag.html>

3.7. Comparació dels models de llenguatge

Per veure la diferència entre diferents models n -gram, es poden construir unigrams (bag-of-words), bigrams, trigram i models 4-gram sobre les paraules d'un llibre (Artificial Intelligence, de Russell i Norvig) i després mostrejar aleatòriament seqüències de paraules a partir de cada un dels quatre models. Els exemples del llibre citat són els següents.

- $n=1$: logical are as are confusion a may right tries agent goal the was
- $n=2$: systems are very similar computational approach would be represented
- $n=3$: planning and scheduling are integrated the success of naive Bayes model is
- $n=4$: taking advantage of the structure of Bayesian networks and developed various languages for writing "templates" with logical variables, from which large networks could be constructed automatically for each problem instance.

A partir d'aquesta petita mostra queda clar que el model unigram no té cap idea de sintaxi, mentre que a mesura que augmenta la mida de la finestra de context la qualitat millora.

Hi ha un límit dels models n -gram, i és que a mesura que creix n el llenguatge produït és més fluid, però tendeixen a reproduir literalment passatges de les dades d'entrada, en comptes de generar text nou. Els models de llenguatge que tenen representacions més complexes de les paraules i del context ho poden fer més bé. La resta d'aquest lliurament mostra com la gramàtica pot millorar un model de llenguatge, i el lliurament següent mostra com els mètodes d'aprenentatge profund (deep learning) donen models de llenguatge impressionants. Una d'aquestes famílies de models de llenguatge, GPT, dona com a resultat exemples de text ben fluids com a resposta a un *prompt*.

4. Gramàtica

Una **gramàtica** és un conjunt de regles que defineixen l'estructura en arbre de les frases admissibles, i un **llenguatge** és el conjunt de frases que segueixen aquestes regles.

Els llenguatges naturals no funcionen exactament com els llenguatges formals: no tenen una frontera nítida entre frases admissibles i no admissibles, ni una única estructura possible per a cada frase. No obstant això, l'estructura jeràrquica és important en el llenguatge natural. Per exemple, la paraula "Stocks" a la frase "Stocks rallied on Monday" no és només una paraula, ni tampoc un nom; en aquest cas és també un sintagma nominal, que és el subjecte del sintagma verbal que el segueix. Les **categories sintàctiques** com per exemple sintagma nominal (en anglès, Noun Phrase, NP) o sintagma verbal (en anglès, Verb Phrase, VP) restringeixen les paraules possibles en cada moment dins la frase, i l'**estructura de la frase** emmarca el significat o **semàntica** de l'oració.

Hi ha molts de models de llenguatge basats en la idea d'estructura sintàctica jeràrquica. Aquí veurem la **gramàtica lliure de context probabilística** (PCFG, probabilistic context-free grammar). Una gramàtica probabilística assigna una probabilitat a cada cadena; lliure del context significa que totes les regles es poden aplicar en qualsevol context: les regles d'un sintagma nominal a l'inici de la frase són les mateixes a l'inici de la frase, enmig o al final, i també les seves probabilitats.

Vegem un exemple de PCFG extret de la pàgina de la llibreria NLTK.

<https://www.nltk.org/howto/grammar.html>

```
>>> from nltk import PCFG
>>> toy_pcfg1 = PCFG.fromstring("""
... S -> NP VP [1.0]
... NP -> Det N [0.5] | NP PP [0.25] | 'John' [0.1] | 'I' [0.15]
... Det -> 'the' [0.8] | 'my' [0.2]
... N -> 'man' [0.5] | 'telescope' [0.5]
... VP -> VP PP [0.1] | V NP [0.7] | V [0.2]
... V -> 'ate' [0.35] | 'saw' [0.65]
... PP -> P NP [1.0]
... P -> 'with' [0.61] | 'under' [0.39]
... """)
```

Imatge: Gramàtica lliure de context probabilística

4.1. Lèxic

El **lèxic** és la llista de paraules permeses. A continuació en tenim un exemple.

<i>Noun</i>	→	stench [0.05] breeze [0.10] wumpus [0.15] pits [0.05] ...
<i>Verb</i>	→	is [0.10] feel [0.10] smells [0.10] stinks [0.05] ...
<i>Adjective</i>	→	right [0.10] dead [0.05] smelly [0.02] breezy [0.02] ...
<i>Adverb</i>	→	here [0.05] ahead [0.05] nearby [0.02] ...
<i>Pronoun</i>	→	me [0.10] you [0.03] I [0.10] it [0.10] ...
<i>RelPro</i>	→	that [0.40] which [0.15] who [0.20] whom [0.02] ...
<i>Name</i>	→	Ali [0.01] Bo [0.01] Boston [0.01] ...
<i>Article</i>	→	the [0.40] a [0.30] an [0.10] every [0.05] ...
<i>Prep</i>	→	to [0.20] in [0.10] on [0.05] near [0.10] ...
<i>Conj</i>	→	and [0.50] or [0.10] but [0.20] yet [0.02] ...
<i>Digit</i>	→	0 [0.20] 1 [0.20] 2 [0.20] 3 [0.20] 4 [0.20] ...

Figure 24.3 The lexicon for \mathcal{E}_0 . *RelPro* is short for relative pronoun, *Prep* for preposition, and *Conj* for conjunction. The sum of the probabilities for each category is 1.

Cada categoria lèxica acaba amb punts suspensius per indicar que hi ha altres paraules a la categoria.

La suma de probabilitats dels membres de cada categoria és la unitat.

En el cas dels **noms comuns**, **noms propis**, **verbs**, **adjectius** i **adverbis**, és inviable en principi enumerar totes les paraules. No només hi ha desenes de milers de mots en cada classe, sinó que se n'afegeixen de nous contínuament. Aquestes cinc categories s'anomenen **classes obertes**.

Els **pronoms**, **pronoms relatius**, **articles**, **preposicions** i **conjuncions** s'anomenen **classes tancades**; tenen un nombre de paraules petit (alguna dotzena) i canvien al llarg dels segles, no de mesos. Per exemple, en anglès, els pronoms *thee* i *thou* eren comuns el segle XVII, estaven de declivi el segle XIX, i avui només es veuen en poesia i alguns dialectes.

5. Parsing

L'anàlisi sintàctica (*parsing*) cerca l'estructura de la frase, d'acord amb les regles de la gramàtica. La podem considerar la cerca d'un arbre vàlid que té a les fulles les paraules de l'oració. Podem començar de dalt a baix, pel símbol *S* que inclou tota l'oració (*S* de l'anglès *sentence*) o bé de baix a dalt, començant per cada paraula. Les aproximacions pures *top-down* o *bottom-up* poden ser ineficients, però, perquè poden acabar repetint esforç en àrees de l'espai de cerca que duen a punts morts. Considerem les dues frases següents en anglès:

1. Have the students in section 2 of Computer Science 101 take the exam.
2. Have the students in section 2 of Computer Science 101 taken the exam?

Aquestes dues oracions comparteixen les primeres deu paraules, però tenen un arbre sintàctic totalment diferent: la primera és una ordre i la segona una pregunta. Un algorisme d'anàlisi d'esquerra a dreta hauria d'endevinar si la primera paraula és part d'una ordre o d'una pregunta, i no ho podria resoldre fins a l'onzena paraula, *take* o *taken*. Si no l'encerta, haurà de tornar enrere fins a l'inici de la frase i recomençar amb l'altra interpretació.

Per evitar aquesta causa d'ineficiència, es pot usar programació dinàmica: cada vegada que s'analitza una subseqüència, s'emmagatzemen els resultats de forma que no s'ha de tornar a analitzar més. Per exemple, una vegada que hem descobert que "the students in section 2 of Computer Science 101" és un sintagma nominal (Noun Phrase, NP), podem emmagatzemar aquest resultat en una estructura de dades anomenada **chart** (taula). Un algorisme que funcionen així s'anomenen **chart parser**. Hi ha molts de tipus de chart parser; aquí tractarem l'**algorisme CYK**.

L'algorisme CYK necessita totes les regles en una d'aquestes dues formes:

Regles lèxiques de la forma $X \rightarrow \text{paraula}[p]$

Regles sintàctiques de la forma $X \rightarrow Y Z [p]$, amb exactament dues categories a la dreta.

Aquest format de gramàtica rep el nom de **forma normal de Chomsky**. Sembla restrictiu, però no ho és. Qualsevol gramàtica lliure de context es pot transformar automàticament a la seva forma normal de Chomsky.

- <https://www.nltk.org/howto/parse.html>

List of items	Rule
<i>S</i>	$S \rightarrow NP VP$
<i>NP VP</i>	$VP \rightarrow VP Adjective$
<i>NP VP Adjective</i>	$VP \rightarrow Verb$
<i>NP Verb Adjective</i>	$Adjective \rightarrow \text{dead}$
<i>NP Verb dead</i>	$Verb \rightarrow \text{is}$
<i>NP is dead</i>	$NP \rightarrow Article Noun$
<i>Article Noun is dead</i>	$Noun \rightarrow \text{wumpus}$
<i>Article wumpus is dead</i>	$Article \rightarrow \text{the}$
<i>the wumpus is dead</i>	

Figure 24.4 Parsing the string "The wumpus is dead" as a sentence, according to the grammar \mathcal{G}_0 . Viewed as a top-down parse, we start with *S*, and on each step match one nonterminal *X* with a rule of the form $(X \rightarrow Y \dots)$ and replace *X* in the list of items with *Y ...*; for example replacing *S* with the sequence *NP VP*. Viewed as a bottom-up parse, we start with the words "the wumpus is dead", and on each step match a string of tokens such as $(Y \dots)$ against a rule of the form $(X \rightarrow Y \dots)$ and replace the tokens with *X*; for example replacing "the" with *Article* or *Article Noun* with *NP*.

```

function CYK-PARSE(words, grammar) returns a table of parse trees
  inputs: words, a list of words
           grammar, a structure with LEXICALRULES and GRAMMARRULES
   $T \leftarrow$  a table //  $T[X, i, k]$  is most probable  $X$  tree spanning  $words_{i:k}$ 
   $P \leftarrow$  a table, initially all 0 //  $P[X, i, k]$  is probability of tree  $T[X, i, k]$ 
  // Insert lexical categories for each word.
  for  $i = 1$  to LEN(words) do
    for each  $(X, p)$  in grammar.LEXICALRULES(words $i$ ) do
       $P[X, i, i] \leftarrow p$ 
       $T[X, i, i] \leftarrow \text{TREE}(X, words_i)$ 
  // Construct  $X_{i:k}$  from  $Y_{i:j} + Z_{j+1:k}$ , shortest spans first.
  for each  $(i, j, k)$  in SUBSPANS(LEN(words)) do
    for each  $(X, Y, Z, p)$  in grammar.GRAMMARRULES do
       $PYZ \leftarrow P[Y, i, j] \times P[Z, j+1, k] \times p$ 
      if  $PYZ > P[X, i, k]$  do
         $P[X, i, k] \leftarrow PYZ$ 
         $T[X, i, k] \leftarrow \text{TREE}(X, T[Y, i, j], T[Z, j+1, k])$ 
  return  $T$ 

function SUBSPANS(N) yields  $(i, j, k)$  tuples
  for length = 2 to N do
    for  $i = 1$  to N + 1 - length do
       $k \leftarrow i + \text{length} - 1$ 
      for  $j = i + 1$  to  $k - 1$  do
        yield  $(i, j, k)$ 

```

Figure 24.5 The CYK algorithm for parsing. Given a sequence of words, it finds the most probable parse tree for the sequence and its subsequences. The table $P[X, i, k]$ gives the probability of the most probable tree of category X spanning $words_{i:k}$. The output table $T[X, i, k]$ contains the most probable tree of category X spanning positions i to k inclusive. The function SUBSPANS returns all tuples (i, j, k) covering a span of $words_{i:k}$, with $i \leq j < k$, listing the tuples by increasing length of the $i : k$ span, so that when we go to combine two shorter spans into a longer one, the shorter spans are already in the table. LEXICALRULES(word) returns a collection of (X, p) pairs, one for each rule of the form $X \rightarrow \text{word} [p]$, and GRAMMARRULES gives (X, Y, Z, p) tuples, one for each grammar rule of the form $X \rightarrow Y Z [p]$.

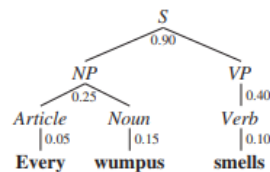


Figure 24.6 Parse tree for the sentence “Every wumpus smells” according to the grammar \mathcal{G}_0 . Each interior node of the tree is labeled with its probability. The probability of the tree as a whole is $0.9 \times 0.25 \times 0.05 \times 0.15 \times 0.40 \times 0.10 = 0.0000675$. The tree can also be written in linear form as $[S [NP [Article \text{every}] [Noun \text{wumpus}]] [VP [Verb \text{smells}]]]$.

5.1. Dependency parsing

Hi ha un enfocament alternatiu molt usat que s'anomena gramàtica de dependències. Això suposa que l'estructura sintàctica està formada per relacions binàries entre els elements lèxics, sense que calguin constituents sintàctics.

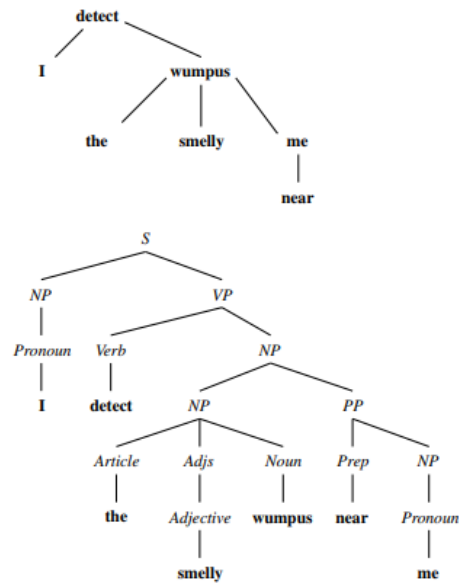


Figure 24.7 A dependency-style parse (top) and the corresponding phrase structure parse (bottom) for the sentence *I detect the smelly wumpus near me.*

En un sentit, la gramàtica de dependència i la gramàtica d'estructura de frase són només variants de notació. Si l'arbre d'estructura sintàctica està anotat amb el cap (*head*) de cada frase, se'n pot obtenir l'arbre de dependència.

En l'altre sentit, es pot convertir un arbre de dependències en un arbre d'estructura d'oració introduint categories arbitràries.

Per tant, no hem de preferir una notació sobre l'altra perquè sigui més potent, sinó perquè sigui més natural (més familiar per als desenvolupadors, o més natural per al sistema d'aprenentatge automàtic que haurà d'aprendre les estructures. En general, els arbres d'estructura d'oració són naturals per a llengües (com l'anglès) amb ordre de paraules més aviat fix; els arbres de dependència són naturals per a llengües (com les romàniques) amb ordre de paraules més aviat lliure, en què l'ordre de les paraules el fixa més la pragmàtica que no les categories sintàctiques.

La popularitat actual de les gramàtiques de dependència ve en gran mesura del projecte **Universal Dependencies**, un projecte open-source de bancs d'arbres que defineix un conjunt de relacions i dona milions d'oracions analitzades en més de 70 llengües.

5.2. Aprenentatge d'un parser a partir d'exemples

Construir una gramàtica per a una fracció significativa d'una llengua és laboriós i vulnerable a errors. Aleshores seria millor **aprendre** les regles de la gramàtica i les probabilitats en lloc d'escriure-les a mà. Per aplicar aprenentatge supervisat, necessitam parells d'entrada-sortida d'oracions i els seus arbres d'anàlisi. El Penn Treebank és la font més coneguda d'aquestes dades, amb més de 100000 oracions en anglès anotades amb estructura d'arbre sintàctic. A continuació en tenim un exemple.

```
[ [S [NP-2 Her eyes]
  [VP were
    [VP glazed
      [NP *-2]
      [SBAR-ADV as if
        [S [NP she]
          [VP did n't
            [VP [VP hear [NP *-1]]
              or
              [VP [ADVP even] see [NP *-1]]
              [NP-1 him]]]]]]]]]]
.]
```

Figure 24.8 Annotated tree for the sentence "Her eyes were glazed as if she didn't hear or even see him." from the Penn Treebank. Note a grammatical phenomenon we have not covered yet: the movement of a phrase from one part of the tree to another. This tree analyzes the phrase "hear or even see him" as consisting of two constituent VPs, [VP hear [NP *-1]] and [VP [ADVP even] see [NP *-1]], both of which have a missing object, denoted *-1, which refers to the NP labeled elsewhere in the tree as [NP-1 him]. Similarly, the [NP *-2] refers to the [NP-2 Her eyes].

Donat un banc d'arbres, se'n pot crear una PCFG (una gramàtica probabilística lliure de context) simplement comptant les vegades que apareix cada tipus de node a l'arbre, tenint en compte que caldrà suavitzar els recomptes molt baixos (*smoothing*). A la figura, hi ha dos nodes de la forma [S[NP...][VP...]]. Els comptaríem, i tots els altres subarbres amb arrel S del corpus.

Si hi ha 1000 nodes S, dels quals 600 són d'aquesta forma, cream la regla següent.

S -> NP VP [0.6]

Tot plegat, el Penn Treebank té devers 10000 tipus de node diferent. Això reflecteix la complexitat de l'anglès, però també indica que els anotadors que crearen el banc d'arbres afavoriren els arbres plans. Per exemple, el sintagma "*the good and the bad*" s'analitza com un sol sintagma nominal en comptes de la conjunció de dos, cosa que dona la regla següent.

NP -> Article Nom Conjunció Article Nom

Hi ha centenars de regles semblants que defineixen un sintagma nominal com una cadena de categories amb una conjunció a qualche lloc pel mig. Una gramàtica més concisa capturaria tots els sintagmes nominals compostos amb una única regla.

NP -> NP Conjunció NP

Es poden obtenir automàticament regles com aquesta, reduint enormement el nombre de regles que s'obtenen del banc d'arbres, i obtenint una gramàtica que generalitza més bé en oracions que no s'han vist abans. Aquesta aproximació s'anomena **data-oriented parsing**.

Els bancs d'arbres no són perfectes, poden contenir errors. Crear un banc d'arbres demana molta feina; necessàriament els bancs d'arbres seran de mida relativament petita, comparat amb tot el text que no ha estat anotat amb arbres. Un enfocament alternatiu és l'**anàlisi no supervisat** (*unsupervised parsing*), en què s'aprèn una nova gramàtica (o se'n millora una d'existent) usant un corpus d'oracions sense etiquetar amb arbres.

Per exemple, l'**algorisme inside-outside**, aprèn a estimar les probabilitats d'una PCFG a partir d'oracions d'exemple sense arbre. Una altra opció d'aprenentatge no supervisat és l'anomenat **curriculum learning**: es comença amb la part més simple del currículum (oracions de dues paraules) i a partir d'aquí es van aprenent estructures més complexes, de 3 paraules, 4 paraules, fins a potser 40 paraules.

També es pot usar anàlisi semisupervisada, en què es comença amb una petita nombre d'arbres per construir una gramàtica inicial, i després s'hi afegeixen un gran nombre d'oracions sense analitzar per millorar la gramàtica. L'enfocament semisupervisat pot aprofitar els parèntesis parcials (*partial bracketing*): es pot usar text àmpliament disponible marcat pels autors i no per experts lingüístics, en forma d'estructura arbòria parcial, com a HTML o anotacions semblants. En el text HTML, la majoria d'etiquetes corresponen a components sintàctics, de forma que aquest *partial bracketing* pot ajudar a aprendre una gramàtica.

Considerem, per exemple, aquest text HTML d'un article de premsa:

```
In 1998, however, as I <a>established in  
<i>The New Republic</i></a> and Bill Clinton just  
<a>confirmed in his memoirs</a>, Netanyahu changed his mind
```

Les paraules envoltades per etiquetes <i></i> formen un sintagma nominal, i les dues cadenes envoltades per <a> formen sintagmes verbals.

6. Gramàtiques augmentades

Fins ara hem parlat de gramàtiques lliures del context. A la pràctica, però, no tots els sintagmes nominals poden aparèixer en qualsevol context amb la mateixa probabilitat. Per exemple, la frase en anglès "I ate a banana" és correcta, "Me ate a banana" és agramatical, i "I ate a bandanna" és improbable.

El problema és que la gramàtica considera les categories lèxiques, com "Pronom", però tot i que tant "I" com "me" són pronoms tots dos, només "I" pot ser el subjecte de l'oració. Es diu que "I" està en cas subjectiu (és el subjecte d'un verb) i "me" en cas objectiu (és l'objecte d'un verb). També es diu que "I" està en primera persona del singular. Una categoria com "Pronom" que s'ha augmentat amb característiques com "cas subjectiu, primera persona singular" s'anomena una **subcategoria**.

Una gramàtica capaç de representar aquesta mena de coneixement és capaç d'afinar més la probabilitat de cada frase.

6.1. Complicacions del llenguatge natural real

La gramàtica de les llengües és interminablement complexa. En aquest apartat mencionarem breument alguns dels aspectes que contribueixen a aquesta complexitat.

Quantificació: pensem la frase "tothom té un objectiu a la vida". Vol dir que totes les persones tenen un mateix objectiu, o que cada una té el seu? Són dues interpretacions diferents i cal decidir l'àmbit d'aplicació del quantificador.

Pragmàtica: per exemple, a la frase "ja som aquí", cal interpretar a què es refereix "aquí": en aquest lloc físic o en aquest punt respecte d'un procés, per exemple.

Dependència a llarg termini: En una frase com "ella no el veia ni el sentia", cal decidir a qui es refereixen els pronoms "ella" i "el".

Ambigüïtat, que pot ser lèxica, sintàctica, semàntica.

Ambigüïtat lèxica: quan una paraula té més d'un significat.

Ambigüïtat sintàctica: quan una frase té diversos arbres sintàctics possibles.

Ambigüïtat semàntica: deriva de l'ambigüïtat sintàctica i genera un significat per a cada estructura possible de l'oració.

Metonímia: hi pot haver ambigüïtat entre significats literals i figurats. Per exemple, si es diu "Chrysler ha anunciat un nou model" no significa que les empreses parlin, sinó que ho fa un portaveu de l'empresa. La metonímia és freqüent i els interlocutors humans la interpretem conscientment.

Metàfora: és una altra figura retòrica, en què un sintagma amb un significat literal s'usa per suggerir un significat diferent mitjançant una analogia. La metàfora es pot entendre com una forma de metonímia quan la relació és de similitud.

Desambiguació

La desambiguació és el procediment de recuperar el significat intentat més probable d'una frase. En un sentit ja tenim un marc per resoldre aquest problema: cada regla té una probabilitat associada, de forma que la probabilitat d'una interpretació és el producte de les probabilitats de les regles que porten a la interpretació. Malauradament, però, les probabilitats reflecteixen la freqüència de les frases en el corpus d'on s'ha après la gramàtica, i aleshores reflecteixen el coneixement general, no el coneixement específic de la situació actual. Per desambiguar adequadament, s'han de combinar els quatre models següents.

1. El **model del món:** la probabilitat que una oració ocorri en el món. Això permet entendre que la frase "estic mort", per exemple, no significa que la meua vida biològica ha acabat però així i tot estic parlant, sinó que estic molt cansat, que tenc un greu problema o que he acabat les vides en un videojoc.
2. El **model mental:** la probabilitat que el parlant tenguí la intenció de comunicar un determinat fet a l'interlocutor. Això combina el que el parlant pensa, el que el parlant pensa que l'interlocutor pensa, i així successivament.
3. El **model del llenguatge:** la probabilitat que es triï una determinada cadena de paraules, donat que el parlant té la intenció de comunicar un fet determinat.
4. El **model acústic:** en la comunicació parlada, la probabilitat que es generi una determinada seqüència de sons, donat que el parlant ha triat una determinada seqüència de paraules.

7. Tasques del llenguatge natural

El processament del llenguatge natural és un camp molt gran. En aquesta secció descrivim algunes de les seves tasques principals: el **reconeixement de la parla**, la **síntesi de text a parla**, la **traducció automàtica**, l'**extracció d'informació**, la **recuperació d'informació** i la **resposta de preguntes**.

El **reconeixement de la parla** (*speech recognition*) és la transformació de so parlat a text. El repte és respondre adequadament fins i tot si hi ha errades en el reconeixement de les paraules individuals. Les xarxes neuronals profundes són un bon model per a aquest problema perquè el problema admet un punt de vista de composició: l'ona acústica forma fonemes, els fonemes formen paraules i les paraules frases.

La **síntesi de text a parla** (*text-to-speech synthesis*) és el procediment invers, de text a so. El repte és pronunciar cada paraula correctament, a més d'aconseguir un flux natural, amb les pauses i èmfasi adequats.

La **traducció automàtica** (*machine translation*) transforma text d'una llengua a una altra. Els sistemes se solen entrenar usant un corpus bilingüe: un conjunt de documents aparellats. Els sistemes de fa dues dècades usaven n-grams, que permetien copsar el significat del text però contenien errors. El 2015, amb els models seqüència a seqüència amb xarxes neuronals recurrents s'obtenia una generalització millor i es podien formar models composicionals que es transmeten la informació entre els diversos nivells. Després, a partir del 2018 amb el mecanisme d'autoatenció del model transformer el rendiment va millorar, i encara més amb sistemes híbrids que combinen característiques de les xarxes recurrents i els transformers.

L'**extracció d'informació** (*information extraction*) és el procés d'adquisició de coneixement a partir de textos, cercant-hi aparicions de classes particulars d'objectes i les seves relacions. Una tasca típica és l'extracció d'adreces físiques en pàgines web, amb camps de base de dades per al carrer, la ciutat i el codi postal. Si el text està estructurat, s'hi poden usar tècniques simples com les expressions regulars. Els sistemes més recents usen xarxes neuronals recurrents, aprofitant la flexibilitat dels *word embeddings*.

La **recuperació d'informació** (*information retrieval*) és la cerca de documents rellevants i importants per a una determinada consulta. Els motors de cerca com Google i Baidu realitzen aquesta tasca milers de milions de vegades cada dia.

La **resposta de preguntes** (*question answering*) és una tasca distinta, en què la consulta realment és una pregunta, i el resultat no ha de ser una llista de documents, sinó una resposta concreta. Per exemple, la resposta a la pregunta "Quina és la capital de Mallorca?" és simplement "Palma". Hi ha hagut sistemes de resposta a preguntes des de la dècada de 1960, però des del canvi de segle aquests sistemes han ampliat moltíssim el seu abast amb l'ús d'informació extreta de la web.

8. Llibreries Python per a NLP

Entre les llibreries més destacades de Python per a NLP hi ha NLTK, spaCy i Stanza. Cadascuna d'elles té característiques úniques que les fan adequades per a diferents necessitats. A continuació es presenten les seves funcionalitats principals i quan utilitzar-les.

NLTK (Natural Language Toolkit)

NLTK és una de les llibreries més antigues i àmpliament utilitzades per al NLP. Ofereix eines flexibles per treballar amb text i llenguatge natural. És ideal per a finalitats educatives i projectes de recerca.

Característiques clau:

- **Processament bàsic del llenguatge:** Tokenització, stemming, lematització i etiquetatge gramatical (POS tagging).
- **Corpus integrats:** Inclou col·leccions de textos, com Gutenberg i WordNet.
- **Anàlisi sintàctica:** Suport per a anàlisi de dependències i arbres sintàctics.
- **Modularitat:** Les seves eines es poden combinar fàcilment per construir fluxos de treball personalitzats.

Avantatges:

- Documentació detallada.
- Gran comunitat i recursos educatius.
- Molt flexible per a aplicacions diverses.

Inconvenients:

- Velocitat relativament lenta per a tasques grans.
- Requereix més configuració que altres llibreries modernes.

Exemple bàsic:

```
import nltk
from nltk.tokenize import word_tokenize

text = "Aquest és un exemple bàsic."
nltk.download('punkt')
tokens = word_tokenize(text)
print(tokens)
```

2. spaCy

spaCy és una llibreria de NLP moderna, optimitzada per a aplicacions industrials. És coneguda per la seva velocitat i precisió, així com pel seu suport a múltiples idiomes.

Característiques clau:

- **Modelatge avançat:** Inclou models preentrenats per a tasques com etiquetatge gramatical, reconeixement d'entitats (NER) i anàlisi de dependències.
- **Integració amb altres eines:** Compatible amb llibreries com TensorFlow i PyTorch per a models personalitzats.
- **Simplicitat d'ús:** Ofereix una API clara i intuïtiva.
- **Suport multilingüe:** Inclou models per a una àmplia varietat d'idiomes.

Avantatges:

- Alt rendiment i velocitat.
- Ideal per a aplicacions de producció.
- Eines avançades per a personalitzar models.

Inconvenients:

- No tan modular com NLTK.
- Dependència dels seus propis models (menys flexibilitat per crear fluxos a mida).

Exemple bàsic:

```
import spacy

nlp = spacy.load("ca_core_news_sm")
text = "Aquest és un exemple amb spaCy."
doc = nlp(text)

for token in doc:
    print(f"{token.text} -> {token.pos_}")
```

3. Stanza

Stanza, desenvolupada per la Universitat de Stanford, és una llibreria potent per al processament del llenguatge natural basada en xarxes neuronals profundes. Està dissenyada per ser precisa i fàcil d'utilitzar per a múltiples idiomes.

Característiques clau:

- **Anàlisi profunda:** Inclou funcionalitats com l'etiquetatge gramatical, reconeixement d'entitats i anàlisi de dependències.
- **Multilingüe:** Suporta més de 60 idiomes.
- **Arquitectura basada en xarxes neuronals:** Utilitza models preentrenats per a una major precisió.

Avantatges:

- Alt nivell de precisió gràcies a les tècniques d'aprenentatge profund.
- Suport a idiomes poc comuns.
- Integració senzilla amb altres eines de NLP.

Inconvenients:

- Pot ser més lent que spaCy per a tasques grans.
- Necessita més recursos de maquinari.

Exemple bàsic:

```
import stanza

stanza.download('ca')
nlp = stanza.Pipeline('ca')

text = "Aquest és un exemple amb Stanza."
doc = nlp(text)

for sentence in doc.sentences:
    for word in sentence.words:
        print(f"{word.text} -> {word.upos}")
```

9. Repositori AIMA

El repositori de NLP publicat per Peter Norvig, autor juntament amb Stuart Russell del llibre Artificial Intelligence, a Modern Approach, és a l'adreça següent

https://github.com/aimacode/aima-python/blob/master/nlp_apps.ipynb

Hi ha diversos exemples d'aplicacions: classificació de text, segmentació en paraules d'un text sense espais en blanc, anàlisi de sentiment...

10. Resum

Aquests són els punts principals del lliurament.

- Els models de llenguatge probabilístics basats en n-grams modelitzen una quantitat sorprenent d'informació sobre un llenguatge. Poden tenir un bon rendiment en tasques tan diverses com la identificació del llenguatge, correcció ortogràfica, anàlisi del sentiment, classificació del gènere o reconeixement d'entitats pròpies (NER, *Named Entity Recognition*)
- Aquests models de llenguatge poden tenir milions de paràmetres, de forma que el preprocessament i el suavitzat de les dades per reduir el soroll són importants.
- Quan es construeix un sistema de llenguatge estadístic, és millor dissenyar un sistema que pot fer un bon ús de les dades disponibles, fins i tot si el model sembla massa simple.
- Els *word embeddings* poden donar una representació rica de les paraules i les seves semblances.
- Per capturar l'estructura jeràrquica del llenguatge, són útils les gramàtiques d'estructura de frase (en particular, les gramàtiques lliures de context). És molt utilitzat el formalisme PCFG (probabilistic context-free grammar), així com la gramàtica de dependència.
- Un treebank pot ser un recurs per aprendre una gramàtica PCFG amb paràmetres.
- Convé augmentar una gramàtica per manejar problemes com ara la concordança entre subjecte i verb i el cas dels pronoms, i per representar informació al nivell de les paraules, en comptes de només al nivell de les categories.
- El llenguatge natural és complex i difícil de capturar amb una gramàtica formal.