# Morfologia computacional

Antoni Oliver

PID_00155231

Universitat Oberta de Catalunya
www.uoc.edu

# Índex

# Introducció

En aquest mòdul presentem una introducció a la morfologia computacional, és a dir, al tractament dels fenòmens morfològics mitjançant procediments informàtics. Aquest material és una adaptació d'un capítol de la tesi doctoral de l'autor en la qual es tractaven fenòmens morfològics per a llengües eslaves, especialment el rus i el croat. Molts dels exemples que s'ofereixen són per al rus i s'ofereixen en alfabet ciríl·lic. Tot i això aquesta introducció és genèrica i serveix per al tractament de moltes altres llengües.

Moltes aplicacions informàtiques que processen aspectes del llenguatge necessiten disposar de grans llistes de paraules en què apareixen la majoria de formes vàlides de la llengua. Per a llengües altament flexives és important disposar de mètodes que ens permetin generar aquestes llistes a partir de llistes de lemes i de regles morfològiques.

La morfologia flexiva està ben tractada des del punt de vista computacional perquè segueix uns processos força regulars. Per exemple, determinar totes les formes del verb *cantar* a partir del lema i d'un conjunt de regles és una tasca senzilla i a més, el conjunt de regles servirà per a un nombre relativament gran de verbs. La morfologia derivativa presenta moltes més dificultats, ja que l'aplicació de les regles no és tan regular. Per exemple, de *content* podem derivar *descontent*, però aquest prefix no serà aplicable a tots els adjectius (**desfeliç*).

Una àrea activa de recerca en morfologia computacional és l'aprenentatge automàtic de la morfologia d'una llengua a partir d'un corpus. La tasca consisteix a descobrir i classificar els fenòmens morfològics d'una determinada llengua a partir d'un conjunt de textos.

# Objectius

Els objectius bàsics que ha d'haver aconseguit l'estudiant una vegada treballats els continguts d'aquest mòdul són els següents:

1. Presentar els conceptes i les tècniques bàsiques en morfologia computacional.
2. Aprendre a generar un formari a partir d'una llista de lemes i un conjunt de regles morfològiques.
3. Comprendre els principis bàsics de les tècniques d'aprenentatge automàtic de la morfologia.

# 1. Fonaments lingüístics

Els llenguatges naturals disposen de mecanismes complexos per a la creació de paraules i de formes a partir d'unitats més petites d'una manera sistemàtica. La part de la lingüística que s'encarrega d'estudiar aquests fenòmens és la morfologia.

Es pot definir la **morfologia** com la part de la gramàtica tradicional que estudia les formes dels mots, independentment de les seves relacions o funcions en la frase o oració (això últim ho estudia la sintaxi) (Mitkov, 2003). La morfologia estudia l'estructura interna del mot tant des del punt de vista de la flexió com de la formació del mot (Tuson, 2000).

Així doncs, l'objecte d'estudi de la morfologia són les paraules. Les llengües tenen centenars de milers de paraules i contínuament estan apareixent noves paraules mentre que altres cauen en desús. Aquesta gran quantitat de paraules es produeixen a partir d'un conjunt molt més reduït d'unitats més petites. La tasca de la morfologia és descobrir i descriure els mecanismes d'aquests processos. En morfologia hi ha dos conceptes clau: la **paraula** i el **morfema** (Moreno, 1994).

## 1.1. Conceptes i unitats bàsics en morfologia

Les unitats bàsiques dels processos morfològics són els **morfemes**. Entenem per morfema la unitat mínima recurrent amb significat, no descomponible en elements menors portadors de significat lèxic o gramatical (Tuson, 2000).

Els morfemes estan formats per **fonemes**, elements mínims del sistema sonor de les llengües, desproveïts de significat però que tenen caràcter contrastiu, és a dir, que són capaços de contrastar i distingir significats. Algunes paraules estan formades per un únic morfema, per exemple *ell*, *sense*, *avui* en català; *ja* ('jo'), *sutra* ('demà') en croat i вдруг ('de sobte'), где ('on'), около ('prop de') en rus. Aquest tipus de paraules reben el nom de **monomorfèmiques**. Moltes altres paraules estan compostes per més d'un morfema, per exemple *des-amortitza-cio-ns* en català, *stan-ar-sk-i* ('relatiu a l'inquilí') en croat i бел-оват-ый ('blanquinós') en rus.

Segons la possibilitat de constituir paraules podem distingir dos tipus de morfemes, els **morfemes lliures** i els **morfemes travats** o **lligats**. Els morfemes lliures poden constituir paraules, per exemple el morfema *door* en anglès o el morfema *cotxe* en català. Els morfemes travats o lligats ocorren només en combinació amb altres morfemes. Tots els afixos són morfemes travats. Per exemple la paraula *doors* consta del morfema lliure *door* i del morfema travat *s*.

Els morfemes també es poden dividir en **morfemes lèxics**, que són els elements amb significat lèxic, és a dir, les arrels dels mots, i en **morfemes gramaticals** o **semàntics categorials**. Aquest segon grup es pot dividir a la vegada en **flexius** i **derivatius**.

Rep el nom de **base** el mot simple o el morfema lèxic al qual afegim afixos per a obtenir un paradigma flexiu o per a formar nous mots (Tuson, 2000). Les bases poden estar constituïdes per un únic morfema, per exemple en rus рук-а ('mà' base: рук); o per més d'un, per exemple en rus бел-оват-ый ('blanquinós', base: беловат).

L'**arrel** és el morfema lèxic, lliure o lligat, comú a tot un paradigma flexiu o derivatiu un cop eliminats tots els afixos (Tuson, 2000). L'arrel coincideix amb la base si la base està formada per un únic morfema (рук-а, base i arrel: рук). Si la base està formada per més d'un morfema un d'aquests morfemes serà l'arrel (бел-оват-ый, base: беловат, arrel: бел). Hi ha paraules que tenen més d'una arrel, llavors parlem de paraules compostes. Per exemple la paraula russa лесозаготвки ('aprovisionament de llenya o fusta') està formada per dues bases: лес i заготовк. La primera d'aquestes bases coincideix amb l'arrel; la segona està formada per l'arrel готов el prefix за i el sufix к. L'arrel, com a morfema, és una forma abstracta que pot presentar algunes alternances en la seva realització.

Entenem per **morf** la representació fònica o gràfica d'un morfema. Un morfema pot estar representat per un sol morf o per més d'un, en aquest segon cas parlem d'**al·lomorfs** d'un mateix morfema. Per exemple, en castellà el morfema plural es pot realitzar com a *-s* (*casa* - *casas*) o com a *-es* (*verdad* - *verdades*). Llavors es diu que *-s* i *-es* són al·lomorfs d'un mateix morfema. L'al·lomorfisme també es pot donar en morfemes lèxics, per exemple en croat: *vuk* (nominatiu singular, 'llop') - *vuč-e* (vocatiu singular) - *vuc-i* (nominatiu plural).

Les llengües tenen a l'entorn de 10.000 morfs, que és una magnitud molt per sota del nombre de paraules. Existeixen unes regles estrictes que governen la combinació d'aquests morfs per a formar paraules. Aquesta manera d'estructurar el lexicó fa que la càrrega cognitiva necessària per a recordar tantes paraules sigui més petita.

Quan es tracta amb textos escrits sovint es fa servir el concepte de **grafema**. Un grafema és la representació gràfica, és a dir, escrita, d'un morfema. Els grafemes poden estar constituïts per una o més lletres.

Un afix és un morf travat que es realitza com una seqüència de fonemes o de grafemes. Els tipus més freqüents d'afixos són els prefixos i els sufixos, però també hi ha els circumfixos i els infixos.

- Un **prefix** és un afix que es posa davant de la base. Per exemple *im-perfecte*.

- Un **sufix** és un afix que es posa després de la base. Per exemple la *s* del plural en *cotxe-s*.

- Un **circumfix** és la combinació d'un prefix i un sufix que junts expressen alguna característica. Per exemple, en la formació del participi passat dels verbs en alemany es fan servir els circumfixos *ge—t* i *ge—n*: *sagen - gesagt* ('dir - dit'); *laufen - gelaufen* ('córrer - corregut').

- Un **infix** és un afix la ubicació del qual es determina segons una o més condicions fonològiques i pot resultar que aparegui dins de l'arrel a la qual s'afixa. En llengua bontoc (família austronèsica) l'infix *-um-* forma un verb a partir de noms i adjectius. L'infix es posa després de la consonant inicial: /fikas/ - /fumikas/ ('fort - ser fort'); /fusul/ - /fumusul/ ('enemic - ser un enemic').

- La **reduplicació** és un cas extrem d'afixació. La forma de l'afix depèn de la base a la qual s'afixa, és a dir, es copia una part de la base. La reduplicació pot ser completa o parcial. En javanès (família austronèsica) la característica habitual-repetitiu s'expressa mitjançant reduplicació completa: /bali/ - /bolabali/ ('tornar'. La reduplicació parcial és més habitual. En yidin (família australiana) el plural s'expressa mitjançant una reduplicació prefixal. Així /mulari/ ('home iniciat') - /mulamulari/ ('homes iniciats'); /guindalba/ ('llangardaix') /guindalgindalba/ ('llangardaixos').

El concepte de *paraula* no és tan clar com pot semblar. En un text podem delimitar les paraules perquè les separem amb espais en blanc (no totes les llengües, però, separen les paraules) o signes de puntuació. Ara bé, en el llenguatge parlat no sempre hi ha una separació entre les diferents paraules. Llavors, com podem definir *paraula*?

> Des d'un punt de vista sintàctic les paraules són les unitats que constitueixen una oració i que es poden classificar depenent de la seva funció en l'estructura de l'oració.

La morfologia tracta de l'estructura interna de les paraules. En moltes llengües una mateixa paraula pot tenir diferents aspectes segons el context sintàctic, per exemple el verb *parlar* pot aparèixer conjugat en diverses formes: *parlo, parles ... parlava, parlaves...* En aquests casos direm que aquestes són diferents *formes* d'una mateixa paraula, és a dir, diferents realitzacions d'una paraula.

Totes les formes d'una determinada paraula constitueixen el seu paradigma. A la taula 1 podem observar el paradigma de la paraula russa книга ('llibre'). Observant aquest paradigma veiem que les formes de datiu i prepositiu singular són iguals, i també les de nominatiu i acusatiu plural. En aquests casos s'acostuma a parlar d'**homonímia paradigmàtica**.

Taula 1. Paradigma del substantiu femení rus книга (*llibre*)

| | Singular | Plural |
|---|---|---|
| Nominatiu | книга | книги |
| Acusatiu | книгу | книги |
| Genitiu | книги | книг |
| Datiu | книге | книгам |
| Instrumental | книгой | книгами |
| Prepositiu | книге | книгах |

Tradicionalment s'ha triat una forma determinada del paradigma com a forma de referència (la forma que surt en els diccionaris). Aquesta forma de referència s'anomena **lema**. Per exemple, en rus el lema dels substantius és la forma corresponent al nominatiu singular.

## 1.2. Funcions de la morfologia

El tipus d'informació que s'expressa morfològicament és molt diferent d'una llengua a una altra. El que en algunes llengües s'expressa sintàcticament, en unes altres s'expressa morfològicament. Per exemple, per a expressar el futur, l'anglès fa servir un verb modal, mentre que el català fa servir un sufix: *I speak*, *parlo*; *I will speak*, *parlaré*. D'igual manera, algun tipus d'informació pot estar present en una llengua i no en una altra. Per exemple, el català marca el plural dels noms i en canvi el japonès no; per exemple, *llibre*, *hon*; *llibres*, *hon*. Aquestes diferències entre les llengües ens permeten classificar-les en certs grups.

### 1.2.1. Classificació de les llengües segons la seva morfologia

Les llengües es poden classificar, segons la seva morfologia, en quatre grups (Trost, 2003):

- **Llengües aïllants, monosil·làbiques o analítiques**: no hi ha afixos (no hi ha morfs lligats) i l'única operació morfològica és la composició. Aquestes llengües es caracteritzen per tenir mots constituïts, generalment, per una síl·laba. Aquests monosíl·labs no tenen terminacions, són invariables i cal destacar que l'entonació pot fer variar-ne radicalment el significat. Així, en xinès el mot *fu* significa 'home, fortuna, ric' i 'prefectura'. En xinès *cinc minuts* es diu *wu feng jung* ('cinc + divisió + hora').

- **Llengües aglutinants**: tots els morfs són lligats, és a dir, tots són afixos que es van unint per a formar les paraules. Per exemple, en turc: *at-lar-im-in* (*cavall* + *plural* + *jo* + *de*), 'dels meus cavalls'.

**Exemples de llengües**

Els següents són alguns exemples de llengües segons la seva morfologia:
- Llengües aïllants: el xinès mandarí.
- Llengües aglutinants: el turc o les llengües finoúgriques, com el finès i l'hongarès.
- Llengües flexives: les llengües indoeuropees.
- Llengües polisintètiques: les llengües inuit (esquimals).

- **Llengües flexives o sintètiques**: les diferents característiques es formen afegint a la base diferents morfs. Per exemple les formes llatines *am-a-ba-s* 'estimaves', *am-a-bi-s* 'estimaràs', *am-a-v-istis* 'estimares'. En molts casos aquests morfs són acumulatius, i per tant la selecció morfema-morf no és tan clara.

- **Llengües polisintètiques**: aquestes llengües expressen morfològicament més informació estructural que la resta de llengües. Per exemple, en llengua aleuta (família esquimoaleuta), partint de *tayá* ('vendre'), *-nacht* (agentiu) i *-luck* (locatiu) es forma *tayánach* ('venedor') i *tayáluch* ('mercat').

**Classificació híbrida**

Les llengües en general no es poden classificar d'una manera clara en aquestes divisions. Per exemple, tot i que el xinès mandarí és una llengua clarament aïllant, també podem trobar sufixos. En el llatí, per exemple, predominen l'aglutinació i la flexió. En aquest treball tractarem amb dues llengües eslaves: el rus i el croat. Aquestes llengües són flexives amb una morfologia molt rica. Són llengües declinables, amb 6 casos el rus (nominatiu, genitiu, datiu, acusatiu, prepositiu i instrumental) i 7 casos el croat (els mateixos que el rus més el vocatiu). Els processos morfològics es poden descriure d'una manera concatenativa i en la morfologia flexiva predomina la sufixació.

### 1.3. Morfologia flexiva i morfologia derivativa

Tradicionalment la morfologia s'ha dividit en **morfologia flexiva** i **morfologia derivativa**. La morfologia flexiva té un caràcter intracategorial i la morfologia derivativa té un caràcter intercategorial.

La morfologia flexiva estudia la combinació de les bases amb els morfemes gramaticals i la morfologia derivativa investiga la construcció de les bases mateixes (Matheson, 1995).

Els morfemes flexius no canvien mai la categoria gramatical de les paraules o morfemes amb les quals s'uneixen ni canvien el seu significat (Fromkin i Rodman, 1988). Les diferents formes d'una paraules produïdes per la seva flexió constitueixen el seu paradigma. Per exemple, la paraula catalana *cotxe* (substantius) té les formes *cotxe* (singular) i *cotxes* (plural). La morfologia flexiva presenta unes característiques que la diferencien de la derivativa: la morfologia flexiva és més sistemàtica i més regular que la morfologia derivativa. La flexió expressa la categoria flexiva de concordança i l'objectiu d'aquesta categoria és indicar una relació sintàctica (Radford i altres, 2000). Per aquest motiu la morfologia flexiva és tancada, és a dir, és un sistema establert a la llengua poc propici perquè es produeixin innovacions per part dels parlants.

La derivació té com a resultat la formació d'una nova paraula afegint a la base un morfema travat. Sovint, en formar la nova paraula hi ha un canvi de categoria gramatical. Els fenòmens derivatius tenen una productivitat restringida,

és a dir, no es poden aplicar de manera sistemàtica. Per exemple, en català del verb *cantar* podem formar el substantiu *cantant*, i del verb *estudiar* podem formar el substantiu *estudiant*. Ara bé, del verb *caçar* no podem formar el substantiu *caçant*. Així doncs, l'aplicació d'un morf derivatiu està restringida a certes subclasses. Així per exemple, el sufix derivatiu anglès *-ity* es combina únicament amb arrels d'origen llatí, mentre que el sufix germànic *-ness* s'aplica a un ventall més ampli de paraules (*rare - rarity - rareness*; *red - *reddity - redness*). La derivació es pot aplicar de manera recursiva, és a dir, una paraula formada per derivació pot generar al seu torn una altra paraula per un procés de derivació (hospital, hospitalitzar, hospitalització). No sempre la interpretació semàntica de la paraula derivada és fàcil. Alguns sufixos donen una informació semàntica única però el significat d'algunes paraules derivades no és sempre composicional.

La composició és la unió de dues o més formes base per a formar una paraula nova, com en les paraules *bioestadística* o *criptoanàlisi*. La darrera part d'un compost normalment defineix les seves propietats morfosintàctiques. La interpretació semàntica és encara més complexa que en el cas de la derivació. Entre els components d'un compost poden existir gairebé totes les relacions semàntiques. Per exemple, en alemany *Wienerschnitzel* és una costella a l'estil vienès; una *Schweineschnitzel* és una costella de porc i una *Kinderschnitzel* és una costella per a nens.

**Divisió entre derivació i composició**

La divisió entre derivació i composició no sempre està clara. Molts sufixos derivatius s'han desenvolupat a partir de paraules fetes servir freqüentment en composició. Un exemple d'això el constitueix el sufix *-ful* anglès: *hopeful*, *wishful*, *thankful*.

### 1.4. Fenòmens no concatenatius

Algunes llengües presenten fenòmens morfològics de natura no concatenativa (Trost, 2003).

- **Morfologia d'arrel i plantilla***: és un tipus de fenomen morfològic que presenten les llengües semítiques. L'arrel, composta per dues, tres o quatre consonants, porta el significat semàntic bàsic. Un patró de marques vocàliques dóna informació sobre la veu i l'aspecte. Una plantilla derivativa dóna informació sobre el tipus de paraula. Els verbs àrabs es construeixen d'aquesta manera. Per exemple, l'arrel *ktb* ('escriure') produeix, entre d'altres, les bases que podem observar a la taula 2.

*En anglès, *root and template morphology*.

Taula 2. Bases produïdes per l'arrel àrab *ktb* ('escriure')

| Plantilla | Patró vocàlic | | |
|---|---|---|---|
| | A (activa) | UI (passiva) | |
| CVCVC | katab | kutib | escriure |
| CVCCVC | kattab | kuttib | fer escriure |
| CVVCVC | ka:tab | ku:tib | cartejar-se |
| tVCVVCVC | taka:tab | tuku:tib | escriure's |
| nCVVCVC | nka:tab | nku:tib | subscriure |
| CtVCVC | ktatab | ktutib | escriure |
| StVCCVC | staktab | stuktib | dictar |

- ***Ablaut***: és l'alternança vocàlica heretada de l'indoeuropeu. És un exemple de la modificació vocàlica com a procés morfològic. Un exemple pot ser el verb anglès *swim*, *swam*, *swum*.

- ***Umlaut***: és un procés d'assimilació regressiva a distància de trets vocàlics. Té el seu origen en un procés fonològic pel qual les vocals de l'arrel es van assimilar amb la vocal d'un sufix. Quan després es va perdre aquest sufix, el canvi en la vocal de l'arrel va ser l'única marca que va quedar de la característica morfològica que presentava el sufix. Per exemple, en alemany el plural d'alguns noms es marquen amb l'*Umlaut*, així, *Mutter* ('mare') - *Mütter* ('mares').

- **Modificació del to**: en algunes llengües la modificació del to pot marcar certes característiques morfològiques. En ngbaka (família nigerokurdufaniana) el temps-aspecte es marca amb quatre varietats tonals.

- **Canvi de l'accent**: per exemple la derivació nom - verb en anglès a vegades es fa amb un canvi de la posició de l'accent, així, per exemple, nom: *éxport*; verb: *expórt**.

***Amb l'accent denotem la síl·laba tònica.**

- **Supleció**: és un fenomen que consisteix a cobrir, en una sèrie morfològica, algunes formes que li falten, amb formes que pertanyen a una altra sèrie. Normalment aquest fenomen té lloc amb formes que s'utilitzen freqüentment. En català, per exemple el verb *anar* té el present *vaig*, *vas*, *va*... i el passat *anava*, *anaves*, *anava*...

- **Morf zero**: algunes vegades una operació morfològica no té cap marca de cap tipus. Per exemple la derivació nom - verb en anglès moltes vegades no es marca formalment: *house* és un nom a la frase "He buys a house"; i un verb a la frase "They house in a cave".

## 1.5. L'estructura de les paraules: morfotàctica. Restricció dels afixos

Com ja hem vist els morfs s'ajunten d'alguna manera per a formar paraules. Una gramàtica a escala de paraula determina com s'ha de fer aquesta unió de morfs. Aquesta part de la morfologia rep el nom de **morfotàctica**.

Les restriccions dels afixos poden ser de diferents tipus: sintàctiques (els afixos s'uneixen a categories específiques), fonològiques, semàntiques o purament lèxiques.

**Exemples**

Un exemple de restricció semàntica és el prefix adjectival *un-* en anglès, que no es pot unir amb un adjectiu que tingui un significat negatiu (*unhappy* - **unsad*).
Un exemple de restricció lèxica és el fet que certs sufixos anglesos només es poden combinar amb paraules d'origen llatí.

## 1.6. La influència de la fonologia

La morfotàctica proporciona les regles de combinació dels morfs en entitats més grans. Però hi ha un altre aspecte a tenir en compte: hi ha regles fonològiques aplicables a la combinació de morfs i que poden introduir canvis en els morfs. La **morfofonologia** estudia aquests canvis.

Moltes aplicacions de morfologia computacional tracten amb textos escrits. El textos escrits no són bones descripcions fonèmiques per a totes les llengües. Per algunes llengües, com el castellà, el finès i el turc, l'ortografia és propera a la transcripció fonètica. En canvi per a altres llengües, com per exemple l'anglès, hi ha una correspondència molt pobra entre escriptura i pronúncia. Com a resultat, molt sovint les aplicacions computacionals tracten amb l'ortografia en comptes de la fonologia. Els principals canvis fonològics que es poden donar en la combinació de morfs són:

- **Assimilació**. És un procés d'influència de dos segments propers a la unió entre morfs pel qual alguna característica canvia per a fer-los més semblants. Per exemple, en castellà el prefix *in-* canvia la *n* per *m* davant de labials (*inseguro* - *improbable*).

- **Epèntesi** (inserció) i **elisió** (supressió) d'un segment sota certes condicions fonològiques. Per exemple, aquest fenomen es dóna en la formació dels plurals anglesos (*door* - *doors*; *dish* - *dishes*).

- Alguns processos morfofonològics funcionen a distàncies més llargues. Els més comuns són els **processos d'harmonia**. L'**harmonia vocàlica** és un procés fonològic pel qual la vocal de més a l'esquerra (en alguns casos la de més a la dreta) d'una paraula influencia totes les vocals que la segueixen (o la precedeixen). Aquest procés té lloc en les llengües finoúgriques, turc i moltes llengües africanes. Per exemple, en turc el plural es forma amb el morfema *-ler* o *-lar* depenent de la vocal que aparegui a l'arrel. Així, si a l'arrel hi ha alguna de les vocals anteriors, el plural es formarà amb *-ler* (*[i]p-ler* 'cordes'); en canvi, si la vocal de l'arrel és posterior, el plural es forma amb *-lar* (*s[o]n-lar* 'finals').

## 1.7. Terminologia i unitats emprades en aquest treball

A banda de les unitats bàsiques introduïdes en aquest apartat (arrel, base, sufix, paradigma, etc.) treballarem també amb unes altres unitats: **pseudoarrel**, **pseudobase**, **pseudoterminació** i **pseudoparadigma**. Una forma determinada la dividirem en dues unitats: una pseudobase i una pseudoterminació. Per exemple, la forma russa мостом la podem dividir en la pseudobase мост i la pseudoterminació ом. En aquest cas la pseudobase coincideix amb l'arrel real i també amb la base, i la pseudoterminació coincideix amb el morfema flexiu d'instrumental singular. Ara bé, aquesta divisió no és l'única possible. La

forma мостом també la podem dividir en la pseudobase мос i la pseudoterminació том. En aquest cas, ni la pseudobase ni la pseudoterminació coincideixen amb les unitats corresponents a l'anàlisi lingüística tradicional. Per aquest motiu hem escollit les denominacions amb els prefix *pseudo*.

Un pseudoparadigma es defineix com el conjunt de pseudoterminacions que són comunes a una o més pseudobases.

Farem servir la denominació *pseudobase*, en comptes de *pseudoarrel*, ja que treballarem principalment amb morfologia flexiva. En molts casos, la pseudobase considerada coincidirà amb l'arrel de la paraula, però en molts d'altres coincidirà amb la base.

**Exemple**

En fer la divisió de la forma беловатый com a pseudobase бел i com a pseudoterminació оватый, la pseudobase coincidiria amb l'arrel real; però en fer la divisió en pseudobase беловат i pseudoterminació ый, la pseudobase coincidiria amb la base de la paraula. En aquesta segona divisió la pseudoterminació ый coincideix amb un sufix flexiu; en canvi, en la primera divisió la pseudoterminació оватый conté el sufix derivatiu ован i el flexiu ый.

## 2. Definició i objectius de la morfologia computacional

La morfologia computacional s'encarrega del processament automàtic de les formes de les paraules, bàsicament en la seva representació grafèmica (forma escrita).

Una tasca fonamental en morfologia computacional consisteix en la segmentació del text en unitats discretes i en l'assignació d'informació morfològica a cada una de les unitats identificades. Per a descriure els diferents fenòmens morfològics s'han desenvolupat diversos formalismes. Segons Karlsson i Karttunen (1997) la recerca en mètodes eficients per a l'anàlisi i generació de formes ja no és una àrea de recerca activa. Actualment la recerca en morfologia computacional està més centrada en l'aprenentatge automàtic de la morfologia, i també en l'adquisició automàtica d'informació lèxica i morfosintàctica. La tasca més bàsica en morfologia computacional consisteix a prendre una cadena de caràcters com a entrada i donar una anàlisi com a resultat. L'entrada podria ser, per exemple, la forma *desamortitzacions* i les sortides podrien ser:

**1)** La cadena de morfemes que componen la paraula: *des-amortitza-cio-ns*.

**2)** La seva interpretació morfosintàctica: *desamortització* NCFP (substantiu comú femení plural).

La primera de les anàlisis la donaria un sistema que s'encarregués de la morfologia derivativa i la segona un que tractés la morfologia flexiva. Fixem-nos especialment en els sistemes que s'encarreguen de la morfologia flexiva, és a dir, en els que donen una anàlisi del tipus 2, ja que aquest treball se centra en aquest tipus d'anàlisi. La manera més fàcil, des del punt de vista computacional, d'aconseguir aquest resultat és disposar d'una llista de parells en la qual a una banda tinguem una determinada forma i a l'altra banda la seva descripció morfològica. Els principals inconvenients d'una solució d'aquest estil són la mida del diccionari, la redundància i la incapacitat de tractar formes desconegudes, és a dir, formes que no estiguin incloses en el diccionari. També cal esmentar com a inconvenient el gran esforç humà que implica confeccionar una llista d'aquestes característiques.

Sobre l'inconvenient que representa la mida del diccionari cal tenir present que els ràpids avenços en els sistemes informàtics, tant en capacitat d'emmagatzematge com de velocitat de processament, i l'aparició d'algorismes de

cerca molt eficients, fan que la mida no sigui tant un problema tècnic, sinó més aviat un problema a l'hora de confeccionar aquestes llistes. L'opció que es pren molt sovint és crear un sistema d'anàlisi morfològica reversible, és a dir, un sistema capaç tant d'analitzar formes com de generar-les. Aquest sistema seria l'encarregat de generar de manera automàtica la llista de formes a partir d'una llista de lemes amb informació morfològica i un conjunt de regles.

Tot i aquesta possibilitat de generació, un sistema basat en llistes de paraules pot no ser vàlid per a algunes llengües. Pot ser interessant per a llengües com l'anglès, amb una morfologia flexiva relativament pobra i fins i tot per a llengües amb una morfologia més rica, com poden ser les romàniques. En canvi, per a llengües aglutinants com el turc, l'hongarès, el finès o l'èuscar un sistema basat en aquesta filosofia seria poc viable, ja que una mateixa paraula pot tenir centenars de formes.

Un sistema menys redundant consisteix a recórrer a un lexicó de lemes. Un algorisme d'interpretació relaciona cada forma amb el seu lema i dóna la interpretació morfosintàctica. Les diferents formes d'una paraula seran concatenacions de cadenes de caràcters de la forma base (el lema o una forma deduïda a partir del lema) i uns afixos. Els afixos s'han d'emmagatzemar en una llista juntament amb la informació morfosintàctica rellevant. El procés d'interpretació consisteix simplement a trobar una seqüència d'afixos i una base que compleixi les regles de la morfotàctica. Un mecanisme tan senzill no sempre és possible per diferents motius:

- Hi ha paraules que tenen formes supletives en la seva flexió. Per exemple, en català, del verb *anar* les formes de present són: *vaig*, *vas*,... En aquests casos és necessari recórrer a un sistema de tractament de les excepcions per a fer el tractament de les formes supletives.

- Alguns morfs s'apliquen d'una manera no concatenativa, per exemple, els temps d'alguns verbs en anglès: *give*, *gave*, *given*. Aquests tipus de fenòmens es poden tractar amb el mateix mecanisme emprat per les formes supletives o bé mitjançant una sèrie de regles específiques que tractin aquests casos.

- Per motius fonològics, les diferents formes d'una paraula poden patir alguns canvis. Per exemple, en anglès els sufixos que comencen per *s* (els plurals dels substantius i la 3a. persona del present dels verbs) no poden seguir directament les arrels acabades en sibilants (p. ex. *dish* - *dishes*). El tractament d'aquest tipus de fenomen no sol a representar cap problema. En aquest exemple és suficient considerar que hi ha dues terminacions de plural: *-s* i *-es*. Per a tots els lemes caldrà especificar quina de les dues terminacions s'aplica, o bé aplicar algun tipus de regla que es pugui deduir a partir del lema.

Els analitzadors morfològics independents del context retornen totes les anàlisis possibles de cada forma d'entrada. Els *taggers* poden fer ús del context

per a determinar quina de les diferents interpretacions d'una forma és la correcta. Alguns sistemes fan servir regles lingüístiques elaborades manualment, mentre que altres fan servir tècniques d'aprenentatge automàtic.

La morfologia computacional té moltes aplicacions pràctiques i esdevé un dels primers passos en moltes aplicacions de processament del llenguatge natural, com poden ser analitzadors sintàctics, sistemes de traducció automàtica, sistemes de recuperació d'informació i sistemes de generació del llenguatge. L'anàlisi morfològica és un prerequisit important per a l'anàlisi sintàctica, ja que l'analitzador necessita conèixer la categoria gramatical i altra informació morfosintàctica de les formes de la frase per analitzar. En molts sistemes, aquesta informació s'obté d'un analitzador morfològic.

El component d'anàlisi i generació morfològica resulta també fonamental en els sistemes de traducció automàtica. En els sistemes més bàsics, de traducció directa, es fa una anàlisi morfològica de les paraules de la frase original que permet generar les formes correctes de la frase traduïda. Altres sistemes, de transferència, fan una anàlisi sintàctica de la frase original, per a la qual és necessària una anàlisi morfològica prèvia. Un cop feta la transferència, caldrà generar les formes correctes de les paraules de la frase destí.

En els sistemes de recuperació d'informació molt sovint es fan servir lematitzadors per a reduir les formes de les paraules relacionades a una única forma canònica, que es farà servir en el procés de recuperació.

**Correctors ortogràfics**

Una aplicació de baix nivell típica són els correctors ortogràfics. Els sistemes que es basen únicament en la comparació amb una llista de paraules tenen molts inconvenients. Una llista d'aquest tipus mai no pot contenir totes les paraules que poden tenir lloc en un text. Molts sistemes fan servir un lexicó d'arrels més un conjunt d'afixos i una sèrie de regles que cobreixen la morfotàctica.

# 3. Tècniques i formalismes en morfologia computacional

En aquest apartat presentem les principals tècniques i formalismes en morfologia computacional: morfologia d'estats finits, morfologia de dos nivells i formalisme de descomposició morfològica.

**Divisió clàssica**

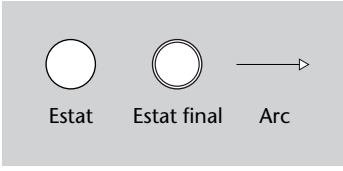
La divisió en morfologia d'estats finits, morfologia de dos nivells i formalisme de descomposició morfològica és una divisió clàssica i es pot trobar en la majoria de textos sobre morfologia computacional, tot i que pot induir a error. Sovint es fa una diferenciació entre morfologia d'estats finits i morfologia de dos nivells, quan la majoria d'implementacions pràctiques de sistemes de dos nivells fan servir tècniques d'estats finits. El formalisme de descomposició morfològica també es pot implementar amb tècniques d'estats finits. Tot i això hem preferit seguir aquest esquema clàssic d'exposició, ja que per a entendre les implementacions de dos nivells cal haver vist primer les tècniques d'estats finits.

## 3.1. Morfologia d'estats finits

Ja que la majoria dels fenòmens morfològics es poden descriure amb expressions regulars, és possible fer servir tècniques d'estats finits per a l'anàlisi morfològica (Karttunen i altres, 1997). En particular, quan la morfotàctica es veu com una simple concatenació de morfs es pot descriure eficientment mitjançant l'ús d'autòmats finits. La seva aplicació no és tan òbvia per a descriure fenòmens no concatenatius o d'infixació, tot i que també és possible aplicar-los a aquest tipus de fenòmens (Beesley i Karttunen, 2000).

Els autòmats d'estats finits sovint es representen mitjançant les anomenades *xarxes de transició d'estats finits*. A la figura 1 podeu observar els símbols estàndard per a representar gràficament una xarxa de transició.

Figura 1. Símbols per a representar una xarxa de transició




**Exemple de xarxa de transició d'estats finits**

Com a exemple de xarxa de transició d'estats finits, a la figura 2 en presentem una que representa les formes de present d'indicatiu del verb català *cantar*. Començant per l'esquerra de la xarxa podem obtenir cada una de les formes desplaçant-nos d'estat a estat. Aquesta xarxa es pot fer servir tant en reconeixement, és a dir, per a veure si una sèrie de caràcters d'entrada són una paraula del llenguatge, com en generació, és a dir, per a enumerar totes les paraules conegudes pel sistema.

Els arcs de la xarxa de transició representada a la figura 2 estan etiquetats només amb el caràcter que hem de consumir per a passar d'un estat a un altre. Una xarxa d'aquest estil serviria únicament per a reconèixer o generar les formes d'un llenguatge. Potser seria útil per a aplicacions del tipus corrector ortogràfic.

Figura 2. Exemple de xarxa de transició d'estats finits

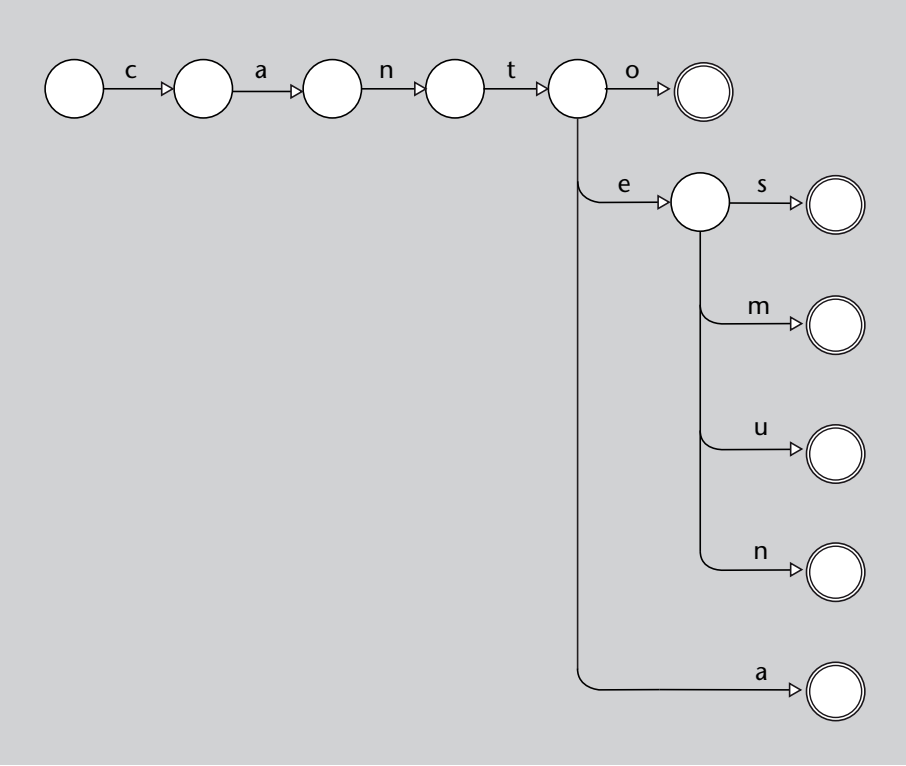



**Figura 2**

Xarxa de transició d'estats finits que representa les formes de present d'indicatiu del verb català *cantar*

En aplicacions més avançades, com pot ser un analitzador morfològic, necessitem relacionar les formes amb el lema i la informació morfosintàctica associades. Un punt fonamental en la morfologia d'estats finits és el fet que la relació entre una forma, el seu lema i la informació morfosintàctica es pot descriure com una relació regular. Si la relació és regular es pot definir mitjançant expressions regulars i es pot compilar en transductors d'estats finits (Karttunen, 2000). En aquest cas cada arc representa una relació regular, de manera que s'obté un transductor lèxic. A la figura 3 podem observar un transductor lèxic que relaciona les formes de present d'indicatiu del verb català *cantar* amb el seu lema i la seva informació morfològica. En aquest cas, el transductor també és reversible, és a dir, es pot fer servir tant en anàlisi com en generació.

Mitjançant tècniques d'estats finits es poden implementar analitzadors morfològics molt eficients i compactes. Per exemple, el transductor lèxic per al castellà de Xerox*, que conté al voltant de 46.000 formes base i pot analitzar o generar al voltant de 3.400.000 formes flexionades, ocupa només 3.349 KB, i és capaç d'analitzar milers de paraules per segon.

**http://www.xrce.xerox.com*

Figura 3. Exemple de transductor lèxic

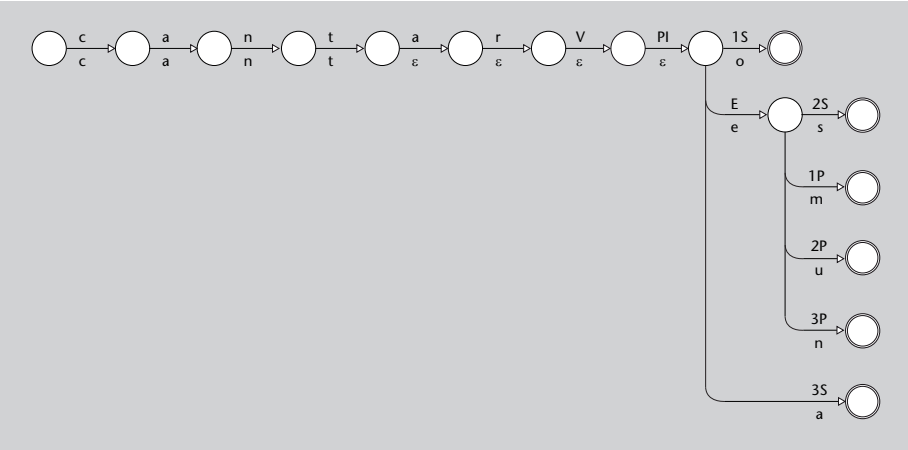



**Figura 3**

Transductor lèxic que representa les formes de present d'indicatiu del verb català *cantar*. El símbol $\varepsilon$ representa l'element buit.

### 3.1.1. Implementació pràctica

Com a exemple representatiu de la metodologia d'estats finits presentem l'analitzador morfològic de Martí (1988). En la seva tesi doctoral, M. A. Martí presenta un analitzador morfològic per al català basat en estats finits. L'analitzador està construït sobre un generador d'analitzadors morfològics especialment dissenyat per a llengües que, com el català, tinguin l'estructura del mot analitzable d'esquerra a dreta; i que tingui mots l'estructura dels quals permeti definir uns components (arrels, afixos, elements de flexió) que presenten un determinat comportament distribucional en el si del mot. Aquest analitzador està constituït pel següent:

- un diccionari d'arrels,
- un diccionari de sufixos que inclou tant els flexius com els derivatius,
- un conjunt de regles que constitueixen l'autòmat i permeten la concatenació de les arrels amb els sufixos flexius i derivatius,
- el conjunt dels models en què s'agrupen les arrels i els sufixos segons les seves característiques de flexió i derivació,
- els atributs morfològics associats a les unitats diverses dels diccionaris i als models, que serveixen també per a expressar les restriccions de les regles de l'autòmat.

L'analitzador segueix el criteri fonamental de rendibilitzar al màxim cada una de les regles i per això tracta amb l'autòmat els aspectes més regulars de les formes i dóna com a noves entrades del diccionari aquelles formes l'anàlisi de les quals exigeix regles específiques. L'analitzador és independent del context i davant d'una forma que pot tenir més d'una interpretació dóna totes les possibles anàlisis.

**Tècniques d'estats finits per a altres llengües**

Les tècniques d'estats finits s'han aplicat amb èxit a una gran quantitat de llengües. L'empresa Xerox ha desenvolupat sistemes basats en tècniques d'estats finits per les llengües següents: anglès, francès, holandès, alemany, hongarès, italià, portuguès, castellà, txec, danès, finès, norueg, polonès, romanès, rus, suec, turc i àrab.

### 3.2. Morfologia de dos nivells

Les gramàtiques fonològiques tradicionals formalitzades per Chomsky i Hall (1968) consisteixen en una sèrie de regles de reescriptura que converteixen representacions fonològiques abstractes en formes superficials per mitjà d'una sèrie de representacions intermèdies. Aquest tipus de regles tenen la forma general:

$$x \rightarrow y \ / \ z \ \_ \ w$$

en què x, y, z i w poden ser cadenes tan complexes com sigui necessari. Aquest tipus de regles també s'anomenen **regles de reescriptura sensibles al context**. La regla anterior reescriuria la cadena *uzxwv* com *uzywv*. Aquest tipus de regles són més potents que les expressions regulars o les regles de reescriptura lliures de context.

Johnson (1972) va demostrar que les regles fonològiques de reescriptura es podien modelar com a transductors d'estats finits. Kaplan i Kay (1981) van arribar a la mateixa conclusió que Johnson demostrant que les regles de reescriptura fonològiques descriuen relacions regulars i que les relacions regulars es poden representar mitjançant transductors d'estats finits.

**PC-KIMMO**

PC-KIMMO és una implementació del formalisme morfològic de dos nivells. Podeu trobar més informació a: *http://www.sil.org/pckimmo*

Els transductors d'estats finits tenen una propietat matemàtica molt important: per a qualsevol parell de transductors aplicats seqüencialment existeix un transductor únic equivalent (Schützemberger, 1961).

Les regles de reescriptura fonològiques tradicionals descriuen la correspondència entre formes lèxiques i formes superficials d'una manera unidireccional i seqüencial des de la forma lèxica fins a la forma superficial. Encara que fos possible modelar el procés de generació de formes superficials d'una manera eficient, mitjançant transductors d'estats finits, invertint el transductor no s'obté un procés d'anàlisi eficient. Per a il·lustrar aquest problema posarem un exemple extret de (Karttunen i Beesley, 2001). Considerem les dues regles següents de reescriptura aplicades seqüencialment:

1) $N \rightarrow m \ / \ \_ \ p$

2) $p \rightarrow m \ / \ m \ \_$

el transductor corresponent relaciona de manera no ambigua la forma lèxica *kaNpat* amb la forma superficial *kammat*, amb una representació intermèdia (la corresponent a aplicar només la regla 1) *kampat*. Ara bé, si invertim el transductor corresponent per a analitzar la forma superficial *kammat*, obtenim que en el nivell intermedi la regla 2 relaciona tant *kampat* com *kammat* a la mateixa forma superficial. Al seu torn, la forma intermèdia *kampat* pot procedir tant de *kampat* com de *kaNpat* aplicant la regla 1. Podem veure il·lustrat

aquest exemple a la figura 4. Aquesta asimetria és una propietat inherent de l'aproximació generativa a la descripció fonològica.

Figura 4. Regles morfològiques de dos nivells

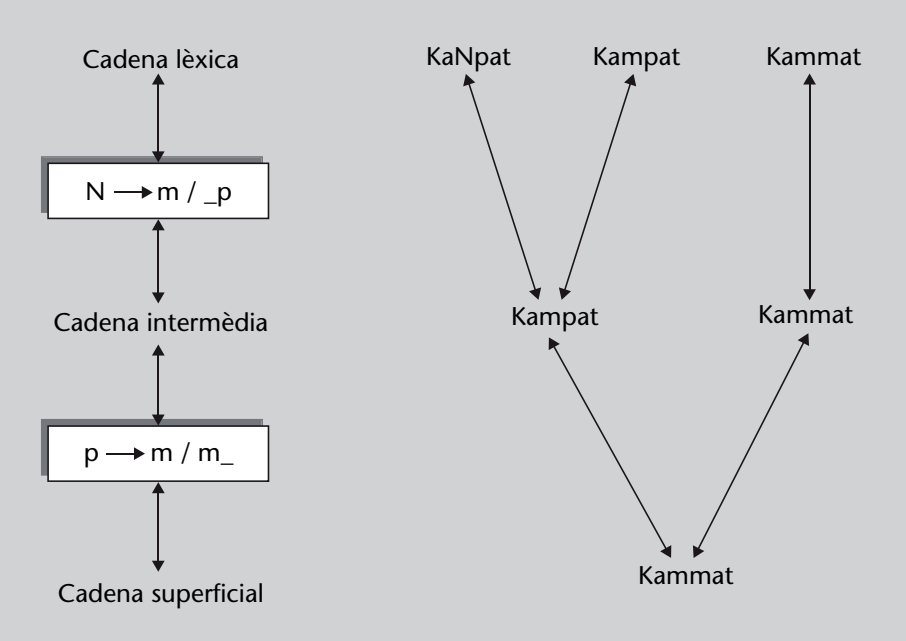



**Figura 4**

Regles que relacionen *kammat* amb *kaNpat*, *kampat* i *kammat*

La solució a aquesta sobreanàlisi implica formalitzar el lexicó mateix com un transductor d'estats finits i compondre'l amb el transductor que codifica les regles.

Kimmo Koskenniemi al principi dels anys vuitanta (Koskenniemi, 1983) va idear una nova manera de descriure les alternances fonològiques en termes d'estats finits. En lloc de regles en cascada amb estats intermedis, les regles estan pensades com a declaracions que restringeixen directament la realització superficial d'una determinada cadena lèxica. Les regles no s'apliquen seqüencialment, sinó en paral·lel. Tal com indica el seu nom la morfologia de dos nivells considera que es pot descriure la morfologia d'una llengua considerant dos nivells: el nivell lèxic i el nivell superficial. En el nivell superficial les paraules apareixen tal com s'escriuen (amb l'única excepció del caràcter nul, que explicarem més endavant). La morfologia de dos nivells es basa en tres idees fonamentals:

**1)** Les regles són restriccions símbol a símbol que s'apliquen en paral·lel, no seqüencialment com les regles de reescriptura.

**2)** Les restriccions es poden referir al context lèxic, al context superficial i a tots dos contextos alhora.

**3)** La cerca lèxica i l'anàlisi morfològica es porta a terme al mateix temps.

Cada regla restringeix una determinada correspondència entre el nivell lèxic i el superficial i l'entorn en què la correspondència està permesa, requerida o prohibida. En el nivell lèxic, l'alfabet inclou alguns símbols especials que es fan servir per a representar característiques que no són grafemes però que constitueixen una informació morfològica necessària. El símbol + indica límit de morf, el símbol # límit de paraula i el símbol `0` és el caràcter nul.

| | | | | |
|---|---|---|---|---|
| Nivell lèxic | `#bliss+s#` | `#fox+s#` | `#dish+s#` | `#watch+s#` |
| Nivell superficial | `0blisses0` | `0foxes0` | `0dishes0` | `0watches0` |

Els dos nivells s'uneixen mitjançant un conjunt de parells de caràcters lèxics i superficials que constitueixen les possibles correspondències. Els parells s'anoten com a caràcter lèxic a l'esquerra i caràcter superficial a la dreta separat per dos punts (:) (p. ex. `a:a`, `+:0`). A aquests parells se'ls poden afegir regles per a restringir-ne l'aplicabilitat. Els parells no afectats per cap regla s'apliquen per defecte. Les regles serveixen per a indicar en quin context s'aplica una determinada correspondència. Les regles es poden veure com a restriccions en la correspondència entre el nivell superficial i la forma lèxica o els morfs. Per tant, s'apliquen en paral·lel i no una darrere de l'altra com en fonologia generativa. Ja que no està implicat l'ordre de les regles, aquesta és una manera de descriure totalment declarativa. Una regla està formada per les parts següents:

- Una substitució que indica el parell de caràcters afectats.
- El context a la dreta i a l'esquerra indica les condicions morfològiques per a la substitució.
- Hi ha quatre operadors disponibles:

– `<=` (operador de restricció de context): fa obligatòria la substitució del caràcter lèxic en el context definit per la regla (altres contextos no es veuen afectats)

– `=>` (operador de coerció superficial): restringeix la substitució del caràcter lèxic a aquest context concret i no pot tenir lloc en cap altre context.

– l'operador `<=>` és una combinació dels dos anteriors; la substitució ha de tenir lloc en aquest context exactament i en cap altre context.

– `/<=` indica prohibició, és a dir, la substitució no pot tenir lloc en aquest context.

**Exemples**

Vegem ara uns quants exemples d'aplicació d'aquestes regles per a l'anglès. La regla (1a):

```
(1a)   +:e <= s x z [ { s c } h ] : _ s ;
```

especifica que el límit de morf lèxic (que s'indica mitjançant +) entre *s*, *x*, *z*, *sh* o *ch* a la banda esquerra i *s* a la banda dreta ha de correspondre a una *e* en el nivell superficial. Per convenció, un parell amb caràcters lèxics i superficials idèntics es pot indicar amb un únic caràcter. Les claus `{ }` indiquen un conjunt d'alternances i els claudàtors `[ ]` una seqüència.

La regla (1a) no fa cap afirmació sobre altres contextos en què + pot correspondre a *e*. La regla cobreix alguns casos en què la *e* s'insereix entre la base i un morf flexiu que comença amb *s* (morfema de plural o de tercera persona singular de present) en anglès. Per defecte el límit de morf es correspon amb el caràcter nul però en el context específic donat es correspon amb *e*. Els següents exemples demostren l'aplicació d'aquesta regla: les barres verticals denoten una parella per defecte i els nombres l'aplicació de les regles corresponents:

| | | | | |
|---|---|---|---|---|
| Nivell lèxic | `#bliss+s#` | `#fox+s#` | `#dish+s#` | `#watch+s#` |
| Regla aplicada | `||||||1||` | `||||1||` | `|||||1||` | `||||||1||` |
| Nivell superficial | `0blisses0` | `0foxes0` | `0dishes0` | `0watches0` |

La regla (1a) no preveu tots els casos en què té lloc l'epèntesi de la *e*. Per exemple les formes *spies*, *shelves* o *potatoes* no estan cobertes. Una regla més completa seria la següent:

```
(1b)   +:e <=> {s x z [ { s c} h:h ] :v [ C y: ] [ C o ] } _ s ;
```

La regla (1b) indica tots els contextos en què + correspon a *e*, ja que fa servir l'operador `<=>`. També fa ús d'algunes convencions addicionals. Dos punts seguits per un caràcter indiquen el conjunt de tots els parells amb aquest caràcter superficial. De la mateixa manera, un caràcter seguit per dos punts indica tot el conjunt de parells amb aquest caràcter lèxic. Es poden definir conjunts de caràcters fent servir un nom global. Per exemple, en la regla anterior la `C` significa tot el conjunt de consonants angleses. Per a poder cobrir el cas de *spies* necessitem una altra regla que permeti la correspondència entre *y* i *i*:

```
(2)   y:i   <=>   C _ { +:e [ +: e ] } ;
                  V C+ _ +: C ;
```

La regla (2) especifica dos contextos diferents. Si qualsevol es compleix, la substitució ha de tenir lloc, ja que els contextos estan units amb l'operador OR. L'operador + del segon context indica com a mínim una ocurrència del signe precedent (l'operador `*` significa un nombre arbitrari d'ocurrències) i `V` significa el conjunt de vocals. Les regles (1) i (2) combinades relacionen correctament *spies* amb `spy+s`. Juntament amb la regla (3) que cobreix la correspondència de *f* a *v*, també es tenen en compte les formes com *shelves* i *potatoes*.

```
(3)   f:v   <=   { e l } _ +: s ;
                 V _ e +: s;
```

Vegem ara com s'interaccionen les tres regles per a produir els resultats esperats:

| | | | | |
|---|---|---|---|---|
| `#spy+s#` | `#toy+s#` | `#shelf+s#` | `#wife+s#` | `#potato+s#` |
| `|||21||` | `|||||||` | `|||||31||` | `|||3||||` | `|||||||1||` |
| `0spies0` | `0toy0s0` | `0shelves0` | `7wives07` | `0potatoes0` |

Les cadenes lèxiques i superficials només es poden correspondre si són de la mateixa longitud. No hi ha possibilitat d'ometre o inserir un caràcter en algun dels nivells, però hi ha molts fenòmens que requereixen inserir o ometre caràcters. Per a poder tractar aquests fenòmens s'inclou el caràcter nul (escrit `0`) tant en l'alfabet lèxic com en el superficial. La correspondència `x:0` representa

l'eliminació d'*x*; mentre que la correspondència `0:x` representa la inserció d'*x*. El caràcter nul no es mostra ni a la sortida ni a l'entrada del sistema. Els símbols + i # es fan correspondre amb el caràcter nul per defecte. Qualsevol altra correspondència d'un símbol + o # s'ha d'explicitar mitjançant una regla.

L'existència del caràcter nul és essencial per al processament. La correspondència entre la cadena lèxica i superficial pressuposa que per cada posició hi ha un parell de caràcters. Això implica que les dues cadenes de caràcters són d'igual longitud (els caràcters nuls es consideren caràcters en aquest sentit). Les regles també es poden interpretar o compilar directament en transductors d'estats finits. L'ús de tècniques d'estats finits permet una implementació molt eficient d'aquest formalisme.

### 3.2.1. El lexicó de continuació

Per ara només hem definit la part de les regles de la morfologia de dos nivells que és responsable de tenir cura dels fenòmens morfofonològics. Aquestes regles es complementen amb un lexicó particionat de morfs (o paraules) que té cura de la formació de paraules per afixació. El lexicó consisteix en sublèxics (no disjuntius), les anomenades **classes de continuació**. Per a cada morf s'especifica en quin sublexicó s'han de buscar les continuacions. Els morfs que poden iniciar una paraula s'emmagatzemen en l'anomenat **lexicó inicial**.

Tot el procés és equivalent a anar pas per pas a través d'un autòmat finit. Una correspondència es pot veure com un moviment d'un estat *X* a un estat *Y* d'un autòmat. Les entrades lèxiques es poden considerar com a arcs d'un autòmat: un sublexicó és una col·lecció d'arcs que tenen un estat origen comú.

El lexicó en la morfologia de dos nivells es fa servir per a dos propòsits:

- Descriure quines combinacions de morfs són permeses en el llenguatge.
- Actuar com a filtre sobre si una forma superficial s'ha de correspondre amb una forma lèxica.

L'ús per al segon propòsit és crucial perquè, si no, no hi hauria manera de limitar la inserció del caràcter nul.

Per a permetre un accés ràpid, els lexicons s'organitzen de manera que es fa una cerca incremental, és a dir, lletra per lletra, i en cada punt de l'arbre estan disponibles exactament aquelles continuacions que porten a morfs permesos. Amb cada node que representa un morf permès s'emmagatzemen les seves classes de continuació. En reconeixement podem fer ús d'aquesta estructura

iniciant el procés per a l'arrel de l'arbre. Cada caràcter que es proposa s'ha de comparar amb el lexicó. Només si és una continuació permesa, aquest node de l'arbre es podrà considerar com una correspondència possible.

En les implementacions més recents el lexicó i les regles de dos nivells es compacten en un únic transductor més gran, que dóna com a resultat un sistema molt compacte i eficient.

### 3.2.2. Formalismes relacionats amb el de dos nivells

Black i altres (1987) proposen un nou format de regles per a solucionar els problemes del formalisme de Koskennieni en descriure els canvis fonològics o ortogràfics que afecten seqüències de caràcters. Proposen un format de regles consistent en:

- Una cadena superficial (denominada LHS)
- Un operador (<= o =>)
- Una cadena lèxica (denominada RHS)

**LHS és l'abreviatura de *left hand side*.**
**RHS és l'abreviatura de *right hand side*.**

Les cadenes lèxiques i superficials han de ser de la mateixa longitud. Les regles per a passar del nivell superficial al lèxic (=>) indiquen que existeix una partició de la cadena superficial en què una part és la LHS de la regla i la cadena lèxica és la concatenació de la RHS corresponent. Les regles per a passar del nivell lèxic al superficial (<=) indiquen que una subcadena de la cadena lèxica que sigui igual que la RHS de la regla ha de correspondre a la cadena superficial de la LHS de la mateixa regla. Les regles següents són equivalents a la regla (1a) del subapartat anterior:

```
ses => s+s
ses <= s+s
shes => sh+s
shes <= sh+s
xes => x+s
xes <= x+s
zes => z+s
zes <= z+s
ches => ch+s
ches <= ch+s
```

La regla `ses => s+s` indica que la cadena superficial *blisses* correspon a la cadena lèxica `bliss+s`, perquè traient *ses* de *blisses* obtenim *blis* i afegint-li

`s+s` obtenim `bliss+s`. La regla inversa `ses <= s+s` serveix per a passar de la cadena lèxica `bliss+s` a la superficial *blisses*.

Aquest tipus de regles integren el context i la substitució en una sola unitat. En lloc d'expressions regulars només es permeten cadenes de caràcters. Un inconvenient és que les regles de nivell superficial a escala lèxica no es poden superposar. Si tenen lloc dos canvis propers, s'han de capturar en una única regla. A més, els fenòmens de llarga distància, com l'harmonia vocàlica, no es poden descriure amb aquest esquema. Per a superar aquests inconvenients, Ruessink (Ruessink, 1989) torna a introduir contextos en les LHS i RHS. A més, les LHS i RHS poden ser de diferent longitud fent servir de nou el caràcter nul.

### 3.2.3. Algunes implementacions pràctiques

**Català**

**CATMORF** (Badia i altres, 1997) és un analitzador basat en un formalisme de múltiples passos de dos nivells implementat en SEGMORF. Tot el sistema d'anàlisi morfològica es pot dividir en tres grans mòduls:

- Un mòdul de manipulació del text, que serveix per a reconèixer aquells elements textuals que no poden ser tractats amb CATMORF: nombres, dates, noms propis, expressions multiparaula, abreviacions; els assigna una etiqueta.

- El mòdul anomenat CATMORF pròpiament, que assigna totes les etiquetes possibles a cada paraula, fent una anàlisi morfològica a partir d'un diccionari de 70.000 entrades, regles de dos nivells i regles gramaticals a escala de paraula.

- Un etiquetador que és una adaptació de l'etiquetador de Multext

SEGMORF és una variant del formalisme morfografèmic ALEP. La principal característica d'aquest formalisme és que permet al lingüista expressar contextos morfografèmics i morfotàctics que restringeixen l'aplicació de les regles de dos nivells. L'especificació d'aquests contextos té sentit, ja que d'aquesta manera es poden restringir l'aplicació de les regles de dos nivells a certes classes d'arrels o quan es tracten fenòmens morfològics que impliquen interacció entre els contextos morfografèmic i morfotàctic.

Augmentant l'expressivitat del formalisme de dos nivells, la gramàtica a escala de paraula es pot mantenir molt simple. Aquesta és una gramàtica d'estil DCG (*definite clause grammar*, 'gramàtica de clàusules definides') (Pereira i Warren, 1980), que construeix les paraules a partir dels morfemes en què ha estat dividida la cadena superficial. En el cas de la flexió s'ha dividit la gramàtica a escala de paraula en dos grups: un per a la morfologia verbal i un altre per a

la morfologia nominal (que s'aplica a noms i adjectius). També s'han implementat regles de morfologia derivativa, però només regles molt productives per a formar paraules d'una determinada categoria a partir de paraules d'una categoria diferent.

El lexicó s'ha creat de manera semiautomàtica a partir d'un diccionari en format llegible per l'ordinador. El lexicó, atès que el sistema tracta la morfotaxi fent servir una gramàtica a escala de paraula i, per tant, no hi ha classes de continuació, conté la informació següent:

- forma de la paraula,
- lema,
- el paradigma flexiu de verbs, noms i adjectius, i
- el bloqueig de regles per a diverses classes d'arrels. Alguns canvis grafèmics són opcionals per a algunes arrels però obligatoris per a altres. Per aquest motiu les entrades lèxiques es marquen amb aquesta informació.

Aquest analitzador proposa una modificació del paradigma de dos nivells per a fer-lo més adequat a la morfologia catalana i a la de la resta de les llengües romàniques. Aquesta modificació s'anomena *múltiples passos de dos nivells*. Aquest analitzador fa servir 114 regles per a la flexió nominal i només 10 regles per a la flexió verbal. Per tant, molt poques regles són aplicables a les dues flexions. En termes d'eficiència aquestes dades demostren que la morfologia del català es pot tractar millor en termes de múltiples passos de dos nivells, que és una filosofia lleugerament diferent de la de dos nivells original. El sistema disposa d'un conjunt diferent de regles de dos nivells i de regles de gramàtica a escala de paraula, segons el procés de formació de paraules a cobrir. Per exemple, donada una forma superficial, primer es provaria de fer servir les regles de dos nivells i la gramàtica a escala de paraula corresponents a la flexió nominal per a trobar descomposicions lèxiques. Després es farien servir les corresponents a la flexió verbal per a trobar descomposicions lèxiques alternatives. El que es pretén amb això és reduir l'espai de cerca.

**Èuscar**

A Alegría i altres (1996) es presenta un sistema d'anàlisi morfològica per a l'èuscar. Atès que l'èuscar és una llengua aglutinant aquest analitzador va més enllà de la segmentació morfològica de paraules i inclou un mòdul extra que fa una anàlisi morfosintàctica completa de cada paraula. Per a aquest propòsit s'ha definit una gramàtica a escala de paraula amb la qual es fa una anàlisi morfosintàctica profunda de cada paraula. El sistema per a fer l'anàlisi morfològica fa servir:

- informació lèxica,
- regles de dos nivells,
- una gramàtica a escala de paraula.

La gramàtica morfosintàctica a escala de paraula s'ha definit fent servir el formalisme PATR-II, ja que és adequat per al tractament de fenòmens complexos, com la concordança dels constituents en cas i nombre. També és útil per a la definició d'estructures lingüístiques complexes. S'han definit 25 regles:

- 11 regles per a la combinació de morfemes de declinació i la combinació d'aquestes amb categories principals,
- 9 regles per a la descripció de morfemes de subordinació verbal,
- 2 regles generals per a la derivació,
- 1 regla per a cada un dels fenòmens següents: el·lipsi, grau de comparació dels adjectius (comparatiu i superlatiu) i composició de noms.

### 3.3. Formalisme de descomposició morfològica

El tercer formalisme que tractarem és el de descomposició morfològica (Alshawi, 1992). La idea bàsica d'aquest formalisme és senzilla i es basa en dos tipus de coneixement:

- Un diccionari que conté informació morfosintàctica sobre la base o la paraula que es considera forma de referència.
- Regles que contenen informació sobre la morfologia de la llengua.

Per anar introduint els diferents conceptes i tècniques presentarem alguns programes en Python. Començarem per un de molt senzill (`morpho1.py`) que simplement demostra com podem concatenar cadenes per a flexionar un verb.

```
arrel="cant"
t1="o"
t2="es"
t3="a"
t4="em"
t5="eu"
t6="en"

print arrel+t1
print arrel+t2
print arrel+t3
print arrel+t4
print arrel+t5
print arrel+t6
```

En aquest programa definim una sèrie de variables: una que conté el lema d'un verb i altres que contenen les terminacions de present d'indicatiu. Mitjançant l'operador + que concatena cadena obtenim les diferents formes corresponents al present d'indicatiu

Una primera modificació d'aquest programa consistiria a tenir el lema (és a dir l'infinitiu del verb) i que les variables que donen les terminacions incorporin també la terminació de lema (separada per ":"). Ara el procés de concatenació serà una mica més complex, ja que abans d'incorporar la terminació de forma caldrà treure la terminació de lema a l'infinitiu. Mostrem el possible programa (recordeu que les possibilitats d'implementació són diverses) i més endavant l'expliquem amb detall:

```
lema="cantar"
regla="o:ar"

(tf,tl)=regla.split(":");
if (lema.endswith(tl)):
    print lema[0:(len(lema)-len(tl))]+tf
```

En aquest exemple només calculem una forma, ja que, com veurem una mica més endavant, convé tenir emmagatzemades les regles en alguna estructura de dades que ens faciliti la manipulació.

Un cop definides les variables `lema` i `regla`, el que hem de fer és descompondre la regla en les terminacions de forma (`tf`) i de lema (`tl`). Això ho fem mitjançant el mètode `split`. En la línia següent verifiquem si el lema acaba amb la terminació de lema (condició per a poder treure aquesta terminació del final del lema per a obtenir l'arrel). Finalment escrivim l'arrel i concatenem la terminació de forma. Fixeu-vos com fem per a obtenir l'arrel a partir del lema i de la terminació de lema. Concretament, l'arrel la calculem com:

```
lema[0:(len(lema)-len(tl))]
```

És a dir, calculem quants caràcters queden de restar la llargada del lema menys la llargada del lema i agafem els caràcters del lema des del 0 fins al resultat d'aquesta resta.

Ara encara podem considerar una millora addicional. Tindrem una llista que conté més d'un infinitiu classificats per tipus (en l'exemple que exposo a continuació tindrem verbs del tipus V1 i del tipus V2). A continuació tindrem una llista de regles que contenen la informació següent: terminació de forma, terminació de lema, etiqueta morfosintàctica, tipus de verb al qual s'aplica la regla. Les etiquetes morfosintàctiques que hem fet servir són les PAROLE* (Eagles). Tot això ho veiem implementat en el programa `morpho2.py`):

***Les etiquetes morfosintàctiques que hem fet servir les hem pres de *http://garraf.epsevg.upc.es/freeling/doc/userman/parole-ca.html*.**

```
diccionari=("cantar:V1", "llegir:V2")
regles=("o:ar:VMIP1S:V1","es:ar:VMIP2S:V1","a:ar:VMIP3S:V1","em:ar:VMIP1P:V1",
"eu:ar:VMIP2P:V1","en:ar:VMIP3P:V1","eixo:ir:VMIP1S:V2","eixes:ir:VMIP2S:V2",
```

```
"eix:ir:VMIP3S:V2","im:ir:VMIP1P:V2","iu:ir:VMIP2P:V2","eixen:ir:VMIP3P:V2")

for entrada in diccionari:
    (lema,tipus)=entrada.split(":")
    for regla in regles:
        (tf,tl,etiqueta,tipus2)=regla.split(":")
        if ((tipus2 == tipus)&(lema.endswith(tl))):
            print lema[0:(len(lema)-len(tl))]+tf,lema,etiqueta
```

En aquest cas el programa escriu la sortida següent: "forma lema etiqueta". Fixeu-vos que en la condició `if` hem afegit la condició que el tipus del lema que estem tractant coincideixi amb la regla que volem aplicar, de manera que les regles de tipus V1 s'apliquin únicament als lemes de tipus V1. La sortida d'aquest programa és la que mostrem a continuació:

```
canto cantar VMIP1S
cantes cantar VMIP2S
canta cantar VMIP3S
cantem cantar VMIP1P
canteu cantar VMIP2P
canten cantar VMIP3P
llegeixo llegir VMIP1S
llegeixes llegir VMIP2S
llegeix llegir VMIP3S
llegim llegir VMIP1P
llegiu llegir VMIP2P
llegeixen llegir VMIP3P
```

Escriure les regles dins del programa mateix no és gaire adequat. Ara proposem una millora addicional en el nostre programa. Tant el diccionari com les regles estaran emmagatzemades en fitxers de text. Tindrem, doncs, un fitxer de text que es pot anomenar `diccionari.txt`, que contindrà:

```
cantar:V1
llegir:V2
menjar:V1A
jugar:V1B
```

Un fitxer de regles que es pot anomenar `regles.txt`, que contindrà:

```
o:ar:VMIP1S:V1
es:ar:VMIP2S:V1
a:ar:VMIP3S:V1
em:ar:VMIP1P:V1
```

```
eu:ar:VMIP2P:V1
en:ar:VMIP3P:V1
eixo:ir:VMIP1S:V2
eixes:ir:VMIP2S:V2
eix:ir:VMIP3S:V2
im:ir:VMIP1P:V2
iu:ir:VMIP2P:V2
eixen:ir:VMIP3P:V2
jo:jar:VMIP1S:V1A
ges:jar:VMIP2S:V1A
ja:jar:VMIP3S:V1A
gem:jar:VMIP1P:V1A
geu:jar:VMIP2P:V1A
gen:jar:VMIP3P:V1A
go:gar:VMIP1S:V1B
gues:gar:VMIP2S:V1B
ga:gar:VMIP3S:V1B
guem:gar:VMIP1P:V1B
gueu:gar:VMIP2P:V1B
guen:gar:VMIP3P:V1B
```

El primer que haurà de fer el nostre programa és llegir les dades dels fitxers per a processar-les. L'estratègia que seguirem serà obrir el fitxer de regles i emmagatzemar-les en una llista i posteriorment tractar el fitxer de diccionari línia a línia (`morpho3.py`):

```
fregles=open("regles.txt","r")

regles=[]

while True:
    linia=fregles.readline().rstrip()
    if not linia:break
    regles.append(linia)
fregles.close()

fdiccionari=open("diccionari.txt","r")

while True:
    linia=fdiccionari.readline().rstrip()
    if not linia:break
    (lema,tipus)=linia.split(":")
    for regla in regles:
        (tf,tl,etiqueta,tipus2)=regla.split(":")
        if ((tipus2 == tipus)&(lema.endswith(tl))):
            print lema[0:(len(lema)-len(tl))]+tf,lema,etiqueta
fdiccionari.close()
```

La sortida en aquest cas és similar però més extensa, ja que en els fitxers de diccionari i de regles contenen més informació.

En els programes que hem presentat fins ara estem fent servir el formalisme per a generar les formes, és a dir, a partir de l'infinitiu expressat en el diccionari cerquem les regles aplicables que permetin eliminar la terminació de lema i afegir la terminació de forma. Aquest formalisme també es pot fer servir per a analitzar una determinada forma, és a dir, per a indicar el lema o lemes i la informació morfològica associades a la forma. El programa (`morpho4.py`) serà molt similar a l'anterior:

```
fregles=open("regles.txt","r")

regles=[]

while True:
    linia=fregles.readline().rstrip()
    if not linia:break
    regles.append(linia)
fregles.close()

fdiccionari=open("diccionari.txt","r")

diccionari={}

while True:
    linia=fdiccionari.readline().rstrip()
    if not linia:break
    (lema,tipus)=linia.split(":")
    diccionari[lema]=tipus;

fdiccionari.close()

forma="cantem"
for regla in regles:
    (tf,tl,etiqueta,tipus2)=regla.split(":")
    if (forma.endswith(tf)):
    lema=forma[0:(len(forma)-len(tf))]+tl
        if (diccionari[lema]==tipus2):
        print forma, lema, etiqueta
```

El programa funciona de la manera següent. Davant una forma, p. ex. *cantem*, es buscaria una regla que tingui la terminació de forma coincident amb les darreres lletres de la forma. Al que queda de l'entrada després de treure-li el sufix, és a dir, *cant*, se li afegeix la terminació corresponent al lema, *ar*, i s'obté *cantar*. Finalment es comprova que en el diccionari existeixi una entrada,

*cantar*, amb la mateixa restricció que la que es troba en la regla, en el nostre cas V1, és a dir, 'verb de la primera conjugació'.

El formalisme de descomposició morfològica implica que tota forma es pot descompondre en una pseudobase i en una pseudoterminació* i que la pseudobase es manté invariable per a tot el paradigma. Per a paradigmes altament irregulars, això implica que s'hagi de considerar com a possible l'arrel zero (que designarem amb el símbol ∅). Així, per exemple, si considerem les formes de present d'indicatiu del verb català *anar*, és a dir, *vaig*, *vas*, *va*, *anem*, *aneu* i *van*, caldrà considerar una arrel zero i tenir el conjunt de regles:

***En aquesta explicació considerem que els processos morfològics són sufixals, però l'algorisme de descomposició morfològica es pot aplicar també amb èxit a fenòmens morfològics en què intervingui la prefixació.**

```
vaig:anar:VIP1S:V126
vas:anar:VIP2S:V126
va:anar:VIP3S:V126
anem:anar:VIP1P:V126
aneu:anar:VIP2P:V126
van:anar:VIP3P:V126
```

i l'entrada següent del diccionari:

∅:V126

El tractament de paradigmes altament irregulars amb aquest formalisme implica implementar un conjunt de regles per a un conjunt d'arrels molt reduït. En el cas del present d'indicatiu del verb català *anar*, les regles només serveixen per a una arrel, en aquest cas l'arrel zero (∅). És per aquest motiu que les formes irregulars s'acostumen a tractar amb una llista que relaciona cada forma amb el seu lema i la informació morfosintàctica associada. Així, tant en l'anàlisi com en la generació es verifica la presència de les formes o del lema en la llista d'irregulars i, en cas que existeixi, es fa servir directament la informació d'aquesta llista. A continuació podem observar la llista que tracta el present d'indicatiu del verb català *anar*:

| | | |
|---|---|---|
| vaig | anar | VIP1S |
| vas | anar | VIP2S |
| va | anar | VIP3S |
| anem | anar | VIP1P |
| aneu | anar | VIP2P |
| van | anar | VIP3P |

Veurem l'ús dels diccionaris morfològics en el mòdul "Etiquetatge morfosintàctic".

A efectes pràctics, i pel fet que els ordinadors actuals tenen una gran velocitat de processament i capacitats d'emmagatzemament molt elevades, l'anàlisi morfològica sovint es fa mitjançant diccionaris de formes amb el lema i la informació morfosintàctica associada. Aquests diccionaris són de gran mida i cobreixen un gran nombre de paraules de la llengua. Aquests sovint es generen a partir de regles i diccionaris de lemes, tal com estem veient en aquest apartat.

El formalisme de descomposició morfològica podrà també tractar processos de canvis fonètics. Per exemple, en croat es produeix un fenomen de sibilarització (pas de velar a sibilant) en el datiu i locatiu singular dels substantius femenins amb l'arrel acabada en *k*, *g* i *h*. A la taula 3 es pot observar la declinació completa dels substantius femenins croats *srna* ('cabirol') i *noga* ('cama, peu'). Podem observar que en el datiu i locatiu de *noga* es produeix un canvi de *g* a *z*.

Taula 3. Declinacions dels substantius femenins croats *srna* ('cabirol') i *noga* ('cama, peu')

| | | | | | |
|---|---|---|---|---|---|
| NS | srna | noga | NP | srne | noge |
| GS | srne | noge | GP | srna | nogu |
| DS | srni | nozi | DP | srnama | nogama |
| AS | srnu | nogu | AP | srne | noge |
| VS | srno | nogo | VP | srne | noge |
| LS | srni | nozi | LP | srnama | nogama |
| IS | srnom | nogom | IP | srnama | nogama |

Aquest tipus de fenòmens es poden tractar considerant dos paradigmes diferenciats. Amb aquesta possibilitat tindríem les regles següents:

```
a:a:NCFSN:NCF1
e:a:NCFSG:NCF1
i:a:NCFSD:NCF1
u:a:NCFSA:NCF1
o:a:NCFSV:NCF1
i:a:NCFSL:NCF1
om:a:NCFSI:NCF1
e:a:NCFPN:NCF1
a:a:NCFPG:NCF1
ama:a:NCFPD:NCF1
e:a:NCFPA:NCF1
e:a:NCFPV:NCF1
ama:a:NCFPL:NCF1
ama:a:NCFPI:NCF1

ga:ga:NCFSN:NCF2
ge:ga:NCFSG:NCF2
zi:ga:NCFSD:NCF2
gu:ga:NCFSA:NCF2
```

```
go:ga:NCFSV:NCF2
zi:ga:NCFSL:NCF2
gom:ga:NCFSI:NCF2
ge:ga:NCFPN:NCF2
ga:ga:NCFPG:NCF2
gama:ga:NCFPD:NCF2
ge:ga:NCFPA:NCF2
ge:ga:NCFPV:NCF2
gama:ga:NCFPL:NCF2
gama:ga:NCFPI:NCF2
```

i les entrades següents del diccionari:

```
srna:NCF1
noga:NCF2
```

Com veiem, aquesta opció fa augmentar considerablement el nombre de regles. Si ens fixem en els dos paradigmes, observem que, excepte pel que fa al datiu i locatiu singular, són exactament iguals. Així doncs, podem cercar alguna estratègia que ens permeti tractar les dues paraules dins del mateix paradigma.

Una primera opció consisteix a definir unes regles d'alternances que s'apliquin un cop generades les formes. Per al cas que estem comentant podríem disposar d'un conjunt de regles (`regles-cro.txt`):

```
a:a:NCFSN:NCF1
e:a:NCFSG:NCF1
i:a:NCFSD:NCF1
u:a:NCFSA:NCF1
o:a:NCFSV:NCF1
i:a:NCFSL:NCF1
om:a:NCFSI:NCF1
e:a:NCFPN:NCF1
a:a:NCFPG:NCF1
ama:a:NCFPD:NCF1
e:a:NCFPA:NCF1
e:a:NCFPV:NCF1
ama:a:NCFPL:NCF1
ama:a:NCFPI:NCF1
```

un diccionari (`diccionari-cro.txt`):

```
srna:NCF1
noga:NCF1
```

i les regles d'alternances (`alternances-cro.txt`):

```
g+i:zi:NCF1
```

Ara necessitarem modificar el nostre programa de generació per a poder tractar les alternances (`morpho5.py`):

```
fregles=open("regles-cro.txt","r")
regles=[]
while True:
    linia=fregles.readline().rstrip()
    if not linia:break
    regles.append(linia)
fregles.close()

fregles2=open("alternances-cro.txt","r")
regles2=[]
while True:
    linia=fregles2.readline().rstrip()
    if not linia:break
    regles2.append(linia)
fregles.close()

fdiccionari=open("diccionari-cro.txt","r")
while True:
    linia=fdiccionari.readline().rstrip()
    if not linia:break
    (lema,tipus)=linia.split(":")
    for regla in regles:
        (tf,tl,etiqueta,tipus2)=regla.split(":")
        if ((tipus2 == tipus)&(lema.endswith(tl))):
            forma=lema[0:(len(lema)-len(tl))]+"+"+tf
            for regla2 in regles2:
                (t1,t2,tipus3)=regla2.split(":")
                if (tipus3 == tipus2):
                    forma=forma.replace(t1,t2)
            forma=forma.replace("+","")
            print forma, lema, etiqueta
fdiccionari.close()
```

El funcionament d'aquesta variant és idèntic a les anteriors, però ara disposem d'un conjunt de regles que ens defineixen una sèrie d'alternances. El programa llegeix aquest fitxer de regles, i en les línies següents de codi les processa:

```
forma=lema[0:(len(lema)-len(tl))]+"+"+tf
for regla2 in regles2:
    (t1,t2,tipus3)=regla2.split(":")
    if (tipus3 == tipus2):
        forma=forma.replace(t1,t2)
forma=forma.replace("+","")
```

En la primera línia creem formes una mica especials, ja que entre l'arrel que es forma i la terminació de forma apareix un signe "+". Són formes de l'estil:

```
srn+a
nog+i
```

Posteriorment es recorren la llista de regles d'alternança i si el tipus és el mateix que l'expressat per a la regla es substitueix, si es pot, la primera part de la regla per la segona. En el cas de *srn+a* no és possible fer cap substitució, però en l'últim pas se substitueix el signe "+" per res, transformant *srn+a* en *srna*. En el cas de `nog+i` és possible fer servir la regla `g+i:zi:NCF1` i substituir `g+i` per `zi` i obtenir `nozi`.

**Exercici**

Modifiqueu el programa d'anàlisi (`morpho4.txt`) de manera que pugui tractar les regles d'alternança.

Una altra possibilitat de tractar aquest tipus de fenòmens és fer servir una notació que ens permeti definir contextos d'aplicació de les regles per a poder unificar tots dos paradigmes en un. A continuació exposem la representació d'aquestes dues declinacions amb aquesta possibilitat (`regles-cro2.txt`):

```
a:a:NCFSN:NCF1
e:a:NCFSG:NCF1
\1i:([^kgh])a:NCFSD:NCF1
zi:ga:NCFSD:NCF1
u:a:NCFSA:NCF1
o:a:NCFSV:NCF1
\1i:([^kgh])a:NCFSL:NCF1
zi:ga:NCFSL:NCF1
om:a:NCFSI:NCF1
e:a:NCFPN:NCF1
a:a:NCFPG:NCF1
ama:a:NCFPD:NCF1
e:a:NCFPA:NCF1
e:a:NCFPV:NCF1
ama:a:NCFPL:NCF1
ama:a:NCFPI:NCF1
```

En les regles (3) i (7) el context s'especifica amb l'expressió `[^kgh]`, que significa els caràcters no inclosos en la llista `{k,g,h}`. El símbol `\1` significa el caràcter de la llista que indica el context que coincideix amb el la forma d'entrada. Per exemple, si s'aplica la regla (3) a *srna*, el símbol `\1` s'unificaria amb la *n*. En les regles (4) i (8) el context s'indica directament amb el caràcter *g*.

Ara ens cal adaptar el nostre programa de generació perquè pugui tractar regles d'aquest tipus (`morph6.py`).

```
import re

fregles=open("regles-cro2.txt","r")

regles=[]

while True:
    linia=fregles.readline().rstrip()
    if not linia:break
    regles.append(linia)
fregles.close()

fdiccionari=open("diccionari-cro.txt","r")

while True:
    linia=fdiccionari.readline().rstrip()
    if not linia:break
    (lema,tipus)=linia.split(":")
    for regla in regles:
        (tf,tl,etiqueta,tipus2)=regla.split(":")
        if (tipus2 == tipus):
            if re.sub(tl,tf,lema):
                forma=re.sub(tl,tf,lema)
                print forma, lema, etiqueta

fdiccionari.close()
```

La inclusió de contextos en les regles ens permet reduir considerablement el nombre de regles necessàries.

# 4. Aprenentatge de la morfologia

En aquest apartat presentem diverses tècniques d'aprenentatge de la morfologia. Aquesta és una àrea d'investigació molt activa i que reuneix especialistes de diverses àrees, com poden ser la lingüística, la compressió de dades i la recuperació d'informació. L'interès que té aquesta àrea en el nostre treball es deu al fet que les tècniques d'adquisició d'informació lèxica i morfosintàctica que presentem precisen el coneixement de la morfologia. Com veurem, hi ha diverses tècniques d'aprenentatge no supervisat de la morfologia a partir de textos sense anotar.

Les estratègies d'aprenentatge de la morfologia es poden dividir en tres grans grups:

**1)** Aprenentatge supervisat

**2)** Aprenentatge no supervisat

**3)** Aprenentatge parcialment supervisat

## 4.1. Estratègies d'aprenentatge supervisat

Entenem per aprenentatge supervisat el cas en què a l'algorisme se li presenta un conjunt de parells de formes, en la majoria dels casos els parells <forma flexionada, forma base>.

Un exemple el trobem en Golding i Thompson (1995), en què es presenta un algorisme que pren parelles de formes (forma flexionada, forma base) i escriu un conjunt de regles de reescriptura simples. Molts dels treballs en l'àrea de l'aprenentatge supervisat de la morfologia s'han dedicat a l'aprenentatge del passat dels verbs anglesos, com per exemple Ling i Marinov (1993), Ling (1994), Mooney i Califf (1995) i Mooney i Califf (1996). També s'han desenvolupat models per a altres llengües, per exemple l'alemany (Westermann i Goebel, 1995) i l'àrab (Plunkett i Nakisa, 1997). Manning (1998) comenta la predominança de l'anglès en els diferents experiments portats a terme fins al moment i presenta una sèrie d'experiments d'aprenentatge de la morfologia per a llengües com el llatí, grec, inuit, cashinahua, anmajere, kayardild i

algunes llengües australianes. La informació que necessita l'algorisme d'aprenentatge és un conjunt de formes superficials amb una representació del seu significat. Per exemple, la forma corata *jabuku*, que és l'acusatiu singular de *jabuka* ('poma'), es representaria com [PRED: apple, NUM: SG CASE: ACC]. L'algorisme funciona fent servir el formalisme de descomposició morfològica i un sistema de categorització basat en l'algorisme ID3. L'ID3 (Quinlan, 1979) és un algorisme de classificació que construeix arbres de decisió a partir d'un conjunt d'exemples.

Los autores van den Bosch i Daelemans (1999) presenten un algorisme de segmentació morfològica que fa servir aprenentatge basat en memòria (MBL, *memory-based learning*) per a segmentar les paraules en morfemes. L'aprenentatge basat en memòria és un tipus d'algorismes d'aprenentatge automàtic supervisat que aprèn emmagatzemant en la memòria exemples de la tasca per dur a terme. Quan es presenta un nou cas l'algorisme busca en memòria els exemples que coincideixen millor respecte a una determinada mètrica, i es pren la solució del nou cas a partir dels exemples més semblants. Clark (2002) presenta també una metodologia d'aprenentatge de la morfologia que fa servir aprenentatge basat en memòria en combinació amb transductors estocàstics.

Theron i Cloete (1997) presenten un algorisme per a aprendre un conjunt de regles de dos nivells per a anglès, xhosa i afrikaans. L'entrada de l'algorisme consisteix en conjunts "forma base - forma flexionada". El procés d'adquisició consta de dues fases: segmentació de la forma flexionada en morfemes i determinació de la regla de dos nivells òptima amb el mínim nombre de contextos necessaris per a restringir-ne l'aplicabilitat.

Alguns treballs fan servir algorismes de lògica inductiva per a l'aprenentatge de la morfologia. Trobem un bon exemple d'aquesta estratègia en el treball fet per Dzerovski i Erjavec (1997a) per a l'eslovè, una llengua eslava del sud. L'objectiu d'aquest treball és veure si un programa de lògica inductiva pot inferir els principis de la morfologia eslovena, de manera que pugui preveure correctament el nominatiu singular d'un substantiu si se li dóna una forma obliqua (és a dir, una forma diferent de la del nominatiu). El sistema aprèn a partir de parells de paraules: el nominatiu i la forma obliqua. Dzerovski i Erjavec (1997b) donen una descripció detallada de l'aplicació de l'algorisme FOIDL (Mooney i Califf, 1995) en l'aprenentatge de la morfologia flexiva nominal eslovena. L'algorisme s'aplica sobre un lexicó de gran mida, format per parelles lema-forma per a induir regles de generació de les formes de genitiu. FOIDL aprèn llistes de decisió de primer ordre, és a dir, llistes ordenades de clàusules. Les llistes de decisió de primer ordre resulten un formalisme molt apropiat per a representar el coneixement lingüístic, ja que permeten representar excepcions a les regles generals d'una manera elegant. Un altre aspecte important de FOIDL és la seva habilitat d'aprendre llistes de decisió a partir únicament d'exemples positius, característica molt interessant dins del processament del llenguatge natural. Molts altres algorismes de lògica inductiva es basen en exemples negatius per a evitar massa hipòtesis generals. En l'algo-

**FOIDL és l'abreviatura de *first order induction of decision lists.***

risme es fa servir el predicat de Prolog `split(A,B,C)`, que divideix una llista A en dues llistes no buides B i C. Aquest predicat es defineix com:

```
split([X,Y|Z],[X],[Y|Z]).
split([X|Y],[X|Z],W):- split(Y,Z,W).
```

Aplicant aquest algorisme per a l'eslovè els resultats obtinguts són prou satisfactoris: el grau d'encert (*accuracy*) de les regles induïdes per al genitiu singular és del 99% per als substantius femenins, del 95% per als substantius neutres i del 85% per als substantius masculins.

Posteriorment, en Manandhar i altres (1998), trobem l'aplicació de l'algorisme CLOG en l'aprenentatge de la morfologia flexiva nominal de cinc llengües: anglès, romanès, txec, eslovè i estonià. CLOG és també un algorisme d'aprenentatge d'arbres de decisió de primer ordre que aprèn a partir d'exemples positius. Comparant els resultats assolits amb CLOG i FOIDL s'observa que el grau d'encert augmenta en aproximadament dos punts en fer servir CLOG.

Altres estratègies se centren en la identificació de bigrames i trigrames que tinguin una alta probabilitat de ser interns a un morfema. Aquestes estratègies es basen en la hipòtesi que la informació local de la cadena de lletres és suficient per a identificar els límits dels morfemes. Aquesta hipòtesi es pot considerar correcta si tots els límits de morfema es trobessin entre parells de lletres $l_1$ - $l_2$ que mai ocorren en l'interior de morfemes, i la hipòtesi s'invalida si les probabilitats condicionals d'una lletra donada la lletra anterior fossin independents de la presència d'un límit de morfema. El procediment descrit en Janssen (1992) i Flenner (1994) comença amb un corpus d'entrenament amb els límits de morfemes marcats manualment. Cada bigrama s'associa a un conjunt de tres valors, la suma dels quals ha de ser menor o igual que 1, que indiquen la freqüència en el corpus d'entrenament d'un límit de morfema que tingui lloc a l'esquerra, a l'interior o a la dreta d'aquest bigrama.

S'ha fet també una extensió de l'algorisme que treballa amb trigrames. Per a una paraula donada, a cada espai entre lletres se li assigna una puntuació que és la suma dels valors rellevants derivats del corpus d'entrenament. Per exemple, en la paraula *cases*, la puntuació corresponent al tall potencial entre *cas* i *es* és la suma de tres valors: la probabilitat que hi hagi un límit de morfema després d'*as* (donat *as*), la probabilitat que hi hagi un límit de morfema entre *s* i *e* (donat *se*) i la probabilitat que hi hagi un límit de morfema abans d'*es* (donat *es*). El fet que aquest valor doni algun tipus d'indicació de la presència d'un límit de morfema és clar, ja que es calcula a partir de la suma d'unes xifres que es deriven a partir d'un corpus amb els límits de morfemes marcats manualment. Queden problemes per resoldre sobre si s'han de cercar pics locals de la suma o bé s'ha d'establir un llindar a partir del qual es consideri límit de morfema. Janssen va observar que la paraula francesa *linguistique* presenta tres pics fent servir un model de trigrames l'anàlisi dóna "lin-guist-ique". El

motiu del pic espuri després de *lin* és que *lin* ocorre amb alta freqüència a final de paraula; també *gui* apareix amb alta freqüència al principi de la paraula. Això porta a pensar que no s'han d'aplicar probabilitats de trigrames calculats per a final de paraula per a trigrames que apareixen al principi de la paraula i a la inversa. Janssen també s'adona que els altres trigrames que hi apareixen (*ing* i *ngu*) tenen una freqüència zero com a límit de morfema en el corpus d'entrenament i proposa que la presència de qualsevol zero faci que la suma total sigui zero. Aquesta proposta, però, no és gaire justificable.

Un altre treball que es pot considerar també dins d'aquest tipus d'estratègia és el de Gaussier (1999). L'objectiu d'aquest treball és adquirir regles derivatives a partir d'una llista de formes flexionades amb la informació de categoria lèxica. Tot i el títol d'aquest article, la metodologia que fa servir no es pot considerar com a no supervisada. Gaussier considera candidates a sufix aquelles terminacions que apareixen com a mínim en dues bases de longitud de 5 caràcters. La primera tasca que es planteja és inferir els paradigmes a partir de les signatures. Anomenem signatures el conjunt de terminacions que apareixen amb una determinada base. Per exemple, si tenim les formes angleses *depart*, *departure* i *departers* la signatura que trobem és "∅:ure:ers" per a la base *depart*. Gaussier fa servir un mètode d'agrupació aglomeratiu jeràrquic que comença amb totes les signatures formant grups diferents i de manera successiva col·lapsa els dos grups més similars. La similitud entre bases es defineix com el nombre de sufixos que comparteixen dues bases i la similitud entre grups es defineix com la similitud entre les dues bases menys similars en el grup respectiu. El treball de Gaussier s'engloba dins de la morfologia derivativa, tot i que es troba amb problemes d'agrupació deguts a fenòmens morfològics flexius.

Jacquemin (1997) explora una font nova d'evidència respecte a l'agrupació de les segmentacions hipotetitzades de les paraules en bases i sufixos. L'algorisme fa servir un corpus i una llista de termes multiparaula i no requereix cap altre tipus d'informació lingüística. Aquest algorisme es podria considerar com d'aprenentatge no supervisat, però l'hem considerat com a supervisat, ja que necessita una llista de termes multiparaula. Considera la hipòtesi que, per exemple, les paraules *gene* i *genetic* tinguin la base comuna *gen* i que les paraules *expression* i *expressed* tinguin la base *express*, es reforça per l'existència de petites finestres (és a dir contextos formats per unes poques paraules) en un corpus que continguin la parella *genetic .. expression* i la parella *gene .. expressed* (no cal, però, que les paraules siguin adjacents). L'algorisme de Jacquemin consisteix a trobar signatures amb les bases més llargues possibles i establir parelles de bases que apareguin juntes en dues o més finestres de longitud 5 o inferior.

En Neuvel i Fulop (2002) es proposa un sistema que indueix relacions morfològiques a partir d'un lexicó de paraules amb informació de categoria gramatical, sense intentar descobrir o identificar morfemes, i que és capaç de generar noves paraules no presents en els exemples d'aprenentatge. El sistema analitza les diferències entre parells de paraules que comparteixen uns

certs caràcters inicials iguals i dedueix unes relacions morfològiques. Aquestes relacions morfològiques s'accepten si es donen entre més d'un parell de paraules. Per exemple, entre les paraules *receive* (V) i *reception* (NS) s'estableix una relació:

```
|*##ceive|V <-> |*##ception|NS
```

en què # indica una lletra que s'ha d'instanciar, però que no està definida, i * indica una lletra que no està definida i que es pot o no instanciar. Aquesta mateixa relació es dóna també entre les paraules *perceive* (V) i *perception* (NS).

## 4.2. Estratègies d'aprenentatge no supervisat

Els algorismes d'aprenentatge no supervisat de la morfologia reben com a única entrada un conjunt de paraules sense cap mena d'informació addicional, ja sigui en forma de llista de formes o com a conjunt de textos sense anotar.

Hi ha un conjunt d'estratègies que intenten identificar prèviament els límits dels morfemes i, després, identificar els morfemes de manera indirecta, basant-se en el grau de predictibilitat del caràcter $n+1$, donats els primers $n$ caràcters. Aquest tipus d'estratègia va ser proposada per primera vegada per Harris (1955) basant-se en l'entropia condicional. Es tracta de construir un dispositiu que generi una llista finita de paraules, el corpus de treball, lletra per lletra amb una probabilitat uniforme, de manera que en qualsevol moment de la generació de la llista (havent generat les primeres $n$ lletres $l_1 l_2 l_3 ... l_n$) podem calcular l'entropia de la lletra següent de totes les continuacions que pot tenir. Aquesta entropia l'anomenarem **entropia condicional prefixal**. De manera similar es calcula la **entropia condicional sufixal** començant la construcció per la banda dreta de les paraules. Harris va proposar que els pics d'entropia condicional prefixal i sufixal indiquen punts de canvi de morfemes. Posteriorment Hafer i Stephen (1974) van explorar diverses millores sobre l'algorisme de Harris.

Un altre grup d'estratègies intenta trobar l'anàlisi òptima del corpus, entenent com a anàlisi òptima la més concisa possible. Aquesta estratègia es basa en l'observació que el nombre de lletres d'una llista de paraules és superior al nombre de lletres d'una llista de les bases més el nombre de lletres dels afixos. Per exemple, totes les formes del verb català *cantar* (*canto*, *cantes*, *canta*, *cantem*, *canteu*, *canten*) sumen un total de 34 lletres. Si aquestes mateixes formes les expressem com a base *cant* i com a conjunt de sufixos *o, as, a, em, eu, en*, en sumen només 14. Aquest és el nucli de l'estratègia de **longitud de descripció mínima** (MDL).

**MDL és l'abreviatura de *minimum description length*.**

Kazakov (1997) presenta una aplicació directa d'aquesta estratègia. En aquest treball presenta una metodologia per a derivar un lexicó de morfemes a partir d'una llista de paraules sense cap mena d'informació addicional. L'algorisme

fa servir la longitud de descripció mínima com a funció d'adaptació (*fitness function*) d'un algorisme genètic simple. Els experiments que presenta estan fets per al francès. En un article posterior, Kazakov i Manandhar (1998) presenten una combinació de tècniques d'aprenentatge supervisat i no supervisat per a la generació de regles de segmentació de paraules a partir d'una llista de formes. En una primera fase supervisada fan servir algorismes genètics que es combinen, en una segona fase no supervisada, amb els algorismes de lògica inductiva FOIDL i CLOG. Aquesta mateixa combinació la presenten també en Kazakov i Manandhar (2001), però aquesta vegada fent servir només l'algorisme CLOG.

Brent (1993) i de Marcken (1995) apliquen una estratègia similar, però fent ús de la noció de compressió pròpia de la teoria de la informació. Brent intenta descobrir el conjunt de sufixos presents en un corpus (i no la divisió en base i sufix de les paraules d'un corpus) fent servir la noció de codificació mínima (*minimal encoding*). La tasca plantejada per de Marcken és determinar la separació en paraules d'una cadena de caràcters que no presenta espais en blanc entre paraules. Aquesta problemàtica es dóna en llengües com el xinès, el japonès i el coreà, en què no se separen les paraules. Els experiments els va dur a terme sobre un corpus del xinès i sobre un corpus de l'anglès en què s'havien eliminat els espais en blanc. L'algorisme comença considerant tots els caràcters com a elements del lexicó i va afegint elements al lexicó si resulten útils per a crear una compressió millor del corpus mateix, o bé si la millora de compressió que s'assoleix és més gran que la longitud (o cost) associada al nou element en el lexicó. A un element lèxic de freqüència $F$ se li associa una longitud de compressió (*compressed length*) de $-logF$. L'algorisme de de Marcken calcula la longitud de compressió de la millor anàlisi del corpus, en què la longitud de compressió és la suma de la longitud de compressió de totes les paraules (o millor dit, fragments que ha detectat l'algorisme) més la longitud de compressió del lexicó. L'anàlisi millor del corpus es calcula fent servir l'algorisme de Viterbi (1967). Si s'aplica l'algorisme de de Marcken a un corpus en què sí que estiguin marcades les separacions entre paraules, s'obtenen resultats interessants, però que no s'assemblen a una anàlisi lingüística d'identificació de bases i afixos.

Déjean (1998) presenta una metodologia d'aprenentatge no supervisat per a generar regles de *stemming* en diverses llengües. Els algorismes de *stemming*, o *stemmers*, redueixen les formes de les paraules a la seva base o a la seva arrel. En aquest article es descriuen els aspectes generals d'un mètode per a descobrir estructures sintàctiques a partir d'un corpus sense anotar. Aquesta operació la divideix en tres parts: el descobriment dels morfemes més freqüents de la llengua, el descobriment de la resta de morfemes i la segmentació de les paraules del corpus. El mètode per al descobriment dels morfemes més freqüents està inspirat en els treballs de Harris i es basa en el nombre de lletres diferents que segueixen a una seqüència de lletres determinada. L'increment d'aquest nombre es fa servir com a indicació de límit de morfema. La resta de morfemes es troben a partir dels morfemes més freqüents: es verifica si, per a una seqüència

de lletres donada, la resta de lletres correspon a un morfema dels ja descoberts. Si com a mínim la meitat de les possibles continuacions són morfemes dels ja trobats, es considera que la resta de continuacions són morfemes. Els principis generals de la metodologia i els algorismes s'han provat en 20 llengües, entre les quals, l'anglès, l'alemany, el turc, el vietnamita, el swahili, el finès, el llatí i l'indonesi.

Schone i Jurafsky (2000) presenten una innovació interessant: en lloc de fixar-se només en la forma de les paraules també filtren els resultats per a l'anàlisi semàntica. El sistema únicament proposa aquells afixos per als quals la base i la base més l'afix són suficientment similars des del punt de vista semàntic. Ara bé, les relacions semàntiques també s'han d'induir automàticament a partir del corpus i, per a assolir això, fa servir anàlisi semàntica latent (LSA). L'algorisme extreu automàticament afixos potencials a partir d'un corpus sense anotar, identifica les parelles de paraules que comparteixen la mateixa arrel però que tenen afixos diferents, i fa servir LSA per a avaluar la relació semàntica entre les paraules per a identificar les relacions morfològiques vàlides.

**LSA és l'abreviatura de *latent semantic analysis*.**

En Snover i Brent (2001) es descriu un sistema per a l'aprenentatge no supervisat d'afixos a partir de textos o de llistes de paraules. El sistema està compost per un model probabilístic generatiu i un algorisme de cerca. En l'article es presenten els resultats dels experiments realitzats per a l'anglès i el francès sobre el corpus del *Wall Street Journal* i sobre el Hansard Corpus.

Unes altres metodologies que s'engloben dins de l'aprenentatge no supervisat són les que fa servir Goldsmith en els seus programes *Automorphology* (Goldsmith, 1999) i *Linguistica* (Goldsmith, 2001).

A la metodologia de Goldsmith li dedicarem l'apartat 5 ja que serà la que utilitzarem com a comparació amb la metodologia d'aprenentatge no supervisat de la morfologia que proposem en aquest treball.

Creutz i Lagus (2002) presenten dos mètodes per a l'aprenentatge no supervisat de la segmentació de paraules en unitats semblants a morfemes, especialment indicats per a llengües amb una morfologia aglutinant. Aquests mètodes no fan distinció entre base i afixos. L'objectiu de la segmentació és adquirir un vocabulari més reduït d'unitats de la llengua i que generalitzi millor que no pas el vocabulari consistent en les paraules tal com apareixen en els textos. Un vocabulari d'aquest tipus es pot fer servir en un model estadístic de la llengua i les unitats presents poden correspondre aproximadament a morfemes. Un dels mètodes que presenten es basa en longitud de descripció mínima (MDL) (Rissanen, 1978) i l'altre en l'optimització del màxim de versemblança (ML) (Fisher, 1925).

**MDL és l'abreviatura de *minimum description length* i ML, de *maximum likelihood*.**

### 4.3. Estratègies d'aprenentatge parcialment supervisat

Clark (2001) introdueix una tercera estratègia general d'aprenentatge de la morfologia: l'aprenentatge parcialment supervisat. Clark en la seva tesi doctoral proposa diverses metodologies d'adquisició no supervisada del llenguatge. En aquest context, considera que l'aprenentatge no supervisat de la morfolo-

gia no és necessari, ja que és possible induir un conjunt de classes sintàctiques a partir de text no etiquetat i posteriorment utilitzar aquesta informació en un algorisme parcialment supervisat per a aprendre les relacions morfològiques. Vist en conjunt, però, el seu algorisme global (el d'aprenentatge de la sintaxi i el d'aprenentatge de la morfologia) es pot considerar com a no supervisat, ja que no fa servir cap coneixement lingüístic previ i l'única font d'informació és un conjunt de textos sense anotar. L'aproximació que proposa Clark consisteix a començar amb un algorisme per a aprendre transductors d'estats finits en un entorn supervisat, per a estendre'l després perquè pugui funcionar en un entorn parcialment supervisat.

Yarowski i Wicentowski (2000) proposen una idea similar, fent servir diverses fonts d'informació per a aprendre tant la morfologia flexiva regular com la irregular. Aquests autors veuen la tasca de l'anàlisi i generació morfològiques com una tasca d'alineament sobre una llista de formes amb prou cobertura. En aquest treball es fa servir una distància d'edició ponderada que ajuda en la tasca d'alineament. L'algorisme és capaç d'induir anàlisis morfològiques tant de formes regulars com irregulars a partir de patrons distribucionals en textos monolingües de gran mida sense supervisió directa. L'algorisme combina quatre models d'alineament basats en la freqüència relativa dins del corpus, la similaritat contextual, la similaritat ponderada de cadena i les probabilitats de transducció flexiva reentrenada de manera incremental. La inducció morfològica descrita es basa únicament en el conjunt de recursos següent (alguns opcionals):

- Una taula de les categories flexives de la llengua juntament amb els sufixos canònics per a cada categoria gramatical.
- Un corpus sense anotar de gran mida.
- Una llista d'arrels nominals, verbals i adjectivals de la llengua (obtinguda normalment a partir d'un diccionari).
- Una llista de consonants i vocals de la llengua.
- Tot i que no és imprescindible, una llista de paraules funcionals freqüents de la llengua, que és útil per a l'extracció de les característiques de similaritat contextual.
- Si està disponible, és útil però no imprescindible disposar de diverses taules de distància / similaritat generades per aquest mateix algorisme en llengües estudiades prèviament, especialment si aquestes llengües són properes.

A continuació presentem els diferents mètodes d'alineament de lemes:

- **Alineament de lemes basat en similaritat de freqüència**. L'alineament d'un lema amb una de les seves formes es pot fer a partir de la freqüència relativa d'aparició dins d'un corpus. Tot i això, no es tracta de fer un alineament simplement buscant freqüències similars, ja que algunes formes són poc freqüents i tenen una freqüència molt més petita que la forma base. Per tant s'han de treballar amb distribucions de freqüències i veure quins dels candidats s'ajusta (o es desvia) més de la distribució de freqüències.

Però l'estimació d'aquestes distribucions presenta el problema que se suposa que no coneixem *a priori* els alineaments correctes (i per tant tampoc les seves freqüències relatives). La simplificació que proposen aquests autors és la de suposar que les freqüències relatives entre una forma determinada i el lema no són significativament diferents entre els processos morfològics regulars i irregulars.

- **Alineament de lemes basat en la similaritat contextual**. Un mètode eficaç per a fer l'alineament entre els lemes i les seves formes és basar-se en la similaritat contextual dels candidats. Es calcula la similaritat cosinus tradicional entre els vectors de les característiques de context ponderades i filtrades. Encara que aquesta mesura també dóna un alt índex de similaritat amb paraules relacionades semànticament, és poc freqüent, fins i tot per a sinònims, que aquestes donin distribucions d'arguments i preferències de selecció més similars que entre les variants flexives de la mateixa paraula. Per a minimitzar la necessitat de recursos d'entrenament s'han identificat els contextos mitjançant un conjunt d'expressions regulars simples sobre les categories tancades. La resta de paraules pertanyents a classes obertes s'etiqueten col·lectivament com a CW. Aquestes expressions sens dubte extrauran molt soroll i fallaran en detectar molts contextos vàlids, però atès que es fan servir sobre un corpus de gran mida, la cobertura parcial i la relació senyal-soroll seran tolerables.

- **Alineament de lemes basat en la distància de Levenshtein ponderada**. Aquest mètode considera la distància d'edició total de la base utilitzant una mesura de distància de Levenshtein ponderada. Aquesta mesura té en compte que en els processos morfològics les vocals i els conjunts vocàlics tenen més probabilitat de patir canvis que les consonants.

- **Alineament de lemes basat en les probabilitats de transformació morfològica**. L'objectiu no consisteix únicament a extreure una taula d'alineaments flexió-arrels acurada, sinó també generalitzar aquesta funció de correspondència amb un model probabilístic generatiu. La probabilitat dependent del context (*context-sensitive probability*) de cada transformació morfològica es pot utilitzar com a mesura de la similaritat de l'alineament.

- **Alineament de lemes basat en la combinació de models i el principi del classificador (*pigeonhole principle*)**. Cap dels mètodes presentats no és prou efectiu per si mateix. S'han aplicat tècniques de combinació de classificadors tradicionals per a combinar els resultats de cada un, escalant-los per a aconseguir un marge dinàmic compatible. El principi del classificador suggereix que, per a una categoria gramatical donada, una arrel no hauria de tenir més d'una flexió ni diverses flexions de la mateixa categoria gramatical haurien de compartir una mateixa arrel.

La restricció d'alineament final que han aplicat es basa en el **principi del classificador** (*pigeonhole*). Aquest principi diu que, per a una categoria gramatical determinada, una arrel no hauria de tenir més d'una flexió, o dit d'una altra manera, que diferents flexions d'una mateixa categoria no poden compartir una mateixa arrel.

Començant amb exemples de <flexió, arrel> no emparellats i sense una indicació prèvia de les transformacions morfològiques vàlides, el grau d'encert de les anàlisis induïdes de 3.888 formes en passat de verbs anglesos supera el 99,2% per al conjunt, amb el 80% per a aquells verbs altament irregulars i el 99,7% per a aquelles formes que presenten una sufixació no concatenativa. El mètode presentat és vàlid per a un model transformacional basat en sufixació i no és vàlid per a llengües amb morfologies prefixals, infixals ni reduplicatives.

## 5. Els programes Automorphology i Linguistica

En aquest apartat presentem en detall un treball molt complet d'adquisició no supervisada de la morfologia d'una llengua portat a terme per John Goldsmith, de la Universitat de Chicago. Goldsmith ha desenvolupat dos programes que es poden descarregar i fer servir lliurement: Automorphology i Linguistica. La descripció del funcionament bàsic del primer programa es pot trobar en un article no publicat de l'autor (Goldsmith, 1999) i la metodologia que fa servir el segon programa es descriu en detall a Goldsmith (2001).

**Adreces web recomanades**

Podeu trobar informació i descarregar els programes Automorphology i Linguistica al lloc web *http://humanities.uchicago.edu/faculty/goldsmith/Automorphology/* i *http://humanities.uchicago.edu/faculty/goldsmith/Linguistica2000*.

Com ja hem comentat, entenem per aprenentatge no supervisat de la morfologia un algorisme capaç d'aprendre relacions morfològiques únicament a partir de textos sense anotar. Els programes que presentem en aquest apartat poden treballar tant amb textos sense anotar com amb llistes de formes. L'objectiu d'aquests programes és descobrir les *signatures*, és a dir el conjunt de sufixos que comparteixen una sèrie de bases, i el conjunt de bases que comparteixen aquests sufixos. El concepte de signatura és similar al de paradigma, però no és exactament igual, ja que en una signatura es poden barrejar processos flexius i derivatius.

**Exemple de signatura**

Donada la llista de formes catalanes *casa*, *cases*, *canto*, *cantes*, *canta*, *cantem*, *canteu*, *canten*, *cantaire*, *cantaires*, *maco*, *maca*, *macos* i *maques* volem que l'algorisme ens retorni el conjunt següent de signatures i arrels associades:

| | |
|---|---|
| a:es | cas |
| o:es:a:em:eu:en:aire:aires | cant |
| co:ca:cos:ques | ma |

En els subapartats 5.1. i 5.2. presentarem en detall el funcionament dels programes Automorphology i Linguistica i veurem el resultat dels programes per a un conjunt de formes russes. Concretament el conjunt de formes russes d'exemple està format per totes les formes de cinc substantius masculins (мост, 'pont'; завод, 'fàbrica'; гарнитур, 'assortiment'; клавицимбал, 'clavicèmbal'; меморандум, 'memoràndum'), de cinc substantius femenins (карта, 'mapa'; мера, 'mesura'; обуза, 'càrrega'; победа, 'victòria'; регата, 'regata'), de cinc substantius neutres (болото, 'pantà, fang'; регентство, 'regència'; самоуправство, 'arbitrarietat'; убийство, 'assassinat, crim, homicidi'; удобство, 'comoditat, confort'), de cinc adjectius (новый, 'nou'; новоиспеченный, 'de nova fornada'; юмористичный, 'humorístic'; абажурный, 'difuminat'; багажный, 'relatiu a l'equipatge') i de cinc verbs en present (делать, 'fer'; гаркать, 'cridar'; гонять, 'portar, fer córrer'; избавлять, 'alliberar, salvar'; одолеть, 'vèncer, superar'). El total de formes és de 235 i el conjunt de signatures i d'arrels associades que s'espera obtenir és el següent:

(1) terminacions: 0:а:ах:ам:ами:е:ом:ов:у:ы

arrels: меморандум:завод:клавицимбал:гарнитур:мост

(2) terminacions: 0:а:ах:ам:ами:е:ой:у:ы

arrels: мер:обуз:карт:побед:регат

(3) terminacions: 0:а:ах:ам:ами:е:о:ом:у

arrels: самоуправств:удобств:болот:регентств:убийств

(4) terminacions: ая:ое:ого:ой:ом:ому:ую:ые:ых:ый:ым:ыми

arrels: юмористичн:абажурн:багажн:нов:новоиспеченн

(5) terminacions: ю:ют:ем:ет:ете:ешь:ть

arrels: избавля:дела:гарка:гоня:одоле

La signatura (1) correspon als substantius masculins, la (2) als substantius femenins, la (3) als substantius neutres, la (4) als adjectius i la (5) als verbs.

## 5.1. Automorphology

El programa Automorphology accepta com a entrada un fitxer de text en una llengua determinada i en fa una anàlisi morfològica. Aquest programa requereix que el nombre mitjà de sufixos per paraula no sigui gaire elevat, de manera que pot funcionar relativament bé per a llengües indoeuropees, però no tan bé per a llengües d'altres famílies.

### 5.1.1. Divisió inicial

La primera operació que es fa és convertir el corpus en una llista de paraules guardant-ne la freqüència absoluta d'aparició, és a dir, les vegades que apareixen en el corpus. En l'explicació es fa servir la notació `[W]` per a indicar el nombre de vegades que la paraula `W` surt en el corpus (freqüència absoluta), i `<W>` per a indicar la freqüència relativa de la paraula `W`, és a dir `[W]` dividit pel nombre total de paraules del corpus. A més, es fa servir `<<W>>` per a indicar `-log<W>`. Representarem com a `|W|` el nombre de caràcters de la paraula. Cal disposar d'una mesura de bondat per a cada una de les possibles divisions d'una paraula en arrel i afixos: la base `S` serà una bona candidata si i només si `S` és una bona candidata per a moltes altres paraules, i de la mateixa manera per als sufixos. Es pot arribar a la millor divisió de manera iterativa de la manera següent:

- Primer s'estableixen dues estructures (l'estructura de base i l'estructura de sufix) en les quals es posen les bases i sufixos candidats, i els associem un nombre per a estimar quantes vegades apareix en el corpus. En la primera

passada a través de la llista de formes es considera que totes les possibles divisions són igualment probables, i com que en general hi ha `|W|-1` maneres diferents de dividir una paraula en base i sufix, a la base se li assignen `[W]/(|W|-1)` ocurrències. Per exemple, si la paraula *taula* apareix 20 vegades en el corpus, a la base *t* se li assignen 5 ocurrències, de la mateixa manera que a les bases *ta*, *tau*, *taul*, *taul*. Es fa el mateix per als sufixos *aula*, *ula*, *la* i *a*, que s'emmagatzemen en l'estructura de sufixos. Aquesta operació es fa per a totes les paraules del corpus.

- En passades successives s'avalua la bondat d'una divisió mitjançant la mesura `V`, que es defineix com:

  ```
  V(Stem/Suffix)=|Stem|*<<Stem>>+|Suffix|*<<Suffix>
  ```

  Fent servir els valors de freqüència establerts en la iteració anterior, per a cada paraula considerem totes les divisions possibles i li assignem un valor de `V`.

- Un cop calculat el valor de la funció `V` per a totes les `|W-1|` divisions, distribuïm `[W]` entre les divisions d'una manera lineal basada en `V`. Per exemple, per a una paraula `W` de 7 lletres, `V*` és la suma sobre totes les possibles divisions $P_i$ de `V(`$P_i$`)`, i si `Stem(3)` són les 3 primeres lletres de la paraula `W` i `Suffix(4)` són les 4 darreres, llavors s'assigna una porció concreta de les ocurrències de `W` (aquesta base i aquest sufix apareixeran també en altres paraules del corpus), aquesta porció es calcula com:

  ```
  [W]*V(Stem(3)-Suffix(4))
  ```

  és a dir:

  ```
  [W]*(|Stem(3)|*log(<Stem(3)>)+ |Suffix(4)|*log(<Suffix(4)>))
  =[W]*(3*<<Stem(3)>>+4*<<Suffix(4)>>)
  ```

  Per a cada paraula la millor divisió serà aquella que tingui el millor valor de la funció V.

- El procés es repeteix iterativament fins que cap paraula canviï la seva millor divisió. S'ha determinat empíricament que el nombre d'iteracions necessàries és inferior a 5.

En aquest punt es disposa d'una divisió inicial en base i sufix per a totes les paraules. Ara cal determinar quines divisions s'han de mantenir, quines eliminar i quines modificar.

### 5.1.2. Determinació de les signatures

Com a heurística inicial, tot i que posteriorment es corregirà, s'eliminen totes les signatures que estan associades a una única base i totes les signatures amb un únic sufix. Les signatures que resten les anomenem **signatures estrictament regulars** i els sufixos continguts en aquestes signatures **sufixos regulars**. Aquests sufixos regulars no són exactament els que voldríem considerar per a la llengua tractada, però sí que són una bona aproximació i constitueixen un bon començament. Un cop s'han determinat els sufixos regulars de la llengua, es torna a analitzar tot el corpus, i permetem únicament les divisions que portin a sufixos regulars. Un cop es torna a analitzar el corpus, encara trobem alguns problemes per solucionar en algunes signatures:

- Tots els sufixos d'una signatura comencen per la mateixa lletra o lletres, i aquesta lletra o lletres haurien de formar part de les bases.

  Per exemple, la signatura

  .t.ted.tions.ts

  s'hauria d'analitzar com a

  .NULL.ed.ions.s.

  amb la *t* afegida al final de la base o bases.

- Tots els sufixos d'una signatura comencen amb el mateix morfema que hauria d'haver estat analitzat com un altre morfema, entre la base i el sufix real.

  Per exemple, per a les formes angleses *worker* i *workers* el programa detectaria la signatura "er.ers" amb la base associada *work*.

- No és possible detectar casos d'al·lomorfisme de la base.

### 5.1.3. Agrupació de signatures

En casos com els de les formes angleses *worker* i *workers* convindria agrupar la signatura detectada ("er.ers") amb la signatura "NULL:s", afegint la part inicial de la signatura a la base, amb l'obtenció de la nova base *worker*. Per a fer això s'aplica un procés d'eliminació de tots els pseudoprefixos de la signatura, que consisteix a eliminar els pseudosufixos i afegir-los a la base en el cas que existeixi una seqüència de dues o més lletres que comparteixin tots els elements d'una signatur; si s'eliminen, es converteix aquesta signatura en una signatura ja existent.

### 5.1.4. Detecció de prefixos

L'algorisme explicat per a la detecció dels sufixos es pot aplicar també per al reconeixement dels prefixos.

### 5.1.5. Determinació dels paradigmes

Per al descobriment dels paradigmes, Goldsmith inicialment fa servir una heurística molt simple, i defineix un paradigma com una signatura *S* amb tres o més bases associades que té la propietat següent: si *S* està composta per *n* sufixos, llavors totes les *n* subsignatures de *n* - 1 també es trobaran en el corpus. A més, cap subsignatura d'un paradigma no pot ser un paradigma. Aquesta heurística no dóna resultats plenament satisfactoris. Existeixen dos problemes principals:

- per una banda molts paradigmes (sobretot en llengües altament flexives) comparteixen una quantitat important de sufixos; i

- per l'altra banda, en un corpus podem trobar paradigmes incomplets, és a dir, no sempre trobarem totes les formes d'un determinat lema. Aquest segon problema també és més important en llengües altament flexives.

Goldsmith ha provat altres tècniques per a solucionar els problemes d'aquesta heurística simple. Per a cada parell de signatures (amb més d'un sufix i amb més d'una base) es calcula l'augment de variància que s'aconsegueix si s'uneixen aquestes dues signatures en una de sola, seleccionant en cada iteració el parell de signatures en les quals la variància ha augmentat menys, i repetint aquesta operació fins que s'assoleix un nivell determinat de confiança.

### 5.1.6. Resultats del programa Automorphology sobre el corpus de prova

Si analitzem el conjunt de formes de prova amb el programa Automorphology obtenim la sortida que podem observar a la taula 4.

La primera xifra indica el nombre de signatures (o paradigmes) detectats, en aquest cas 7. Al costat de cada signatura, el programa ens ofereix dues xifres, la primera indica el nombre de bases associades a la signatura i la segona és el nombre de formes representades, xifra que s'obté de la multiplicació del nombre de terminacions de la signatura pel nombre de bases associades. Com podem observar en els resultats, els paradigmes corresponents als substantius (1, 2 i 3) han estat detectats correctament. Les signatures (6) i (7) són incorrectes. El (6) representa algunes formes adjectives i algunes formes verbals. Hem d'observar, però, que en aquesta signatura es representen unes formes no existents en el conjunt de formes d'entrada, concretament юмористично,

Taula 4. Anàlisi del programa Automorphology per al conjunt de formes de prova

| | |
|---|---|
| 7 | |
| (1) | .NULL.а.ах.ам.ами.е.ом.ов.у.ы. 5 50 |
| | мост, клавицимбал, меморандум, завод, гарнитур |
| (2) | .NULL.а.ах.ам.ами.е.о.ом.у. 5 45 |
| | тырл, регентств, самоуправств, удобств, болот |
| (3) | .NULL.а.ах.ам.ами.е.о.у.ы. 5 45 |
| | мер, карт, регат, обуз, побед |
| (4) | .ая.о.ого.ом.ую.ы.ых.ым.ыми. 5 45 |
| | нов, багажн, новоиспеченн, абажурн, юмористичн |
| (5) | .ю.ют.ем.ет.ешь.ть. 5 30 |
| | избавля, гарка, одоле, гоня, дела |
| (6) | .NULL.е. 15 30 |
| | гаркает, юмористично, юмористичны, избавляет, багажно, багажны, одолеет, новоиспеченно, новоиспеченны, ново, новы, делает, абажурно, абажурны, гоняет |
| (7) | .NULL.у. 5 10 |
| | юмористичном, новом, багажном, новоиспеченном, абажурном |

юмористичны, багажно, багажны, новоиспеченно, новоиспеченны, ново, новы, абажурно i абажурны, és a dir, les pseudobases de tipus adjectival més la terminació NULL. La signatura (7) representa formes adjectivals però és incorrecta. Atesos aquests errors, les signatures (4), de tipus adjectival, i la (5), de tipus verbal, són correctes però incompletes.

## 5.2. Linguistica

**MDL és l'abreviatura de *minimum description length*.**

En el programa Linguistica, Goldsmith fa una aproximació per longitud de descripció mínima (MDL). Aquesta aproximació es basa en l'observació que el nombre de lletres en una llista de paraules és més gran que el nombre de lletres en una llista d'arrels i afixos presents en la llista original. La idea central de l'anàlisi per longitud de descripció mínima (Rissanen, 1989) és compon de quatre parts:

**1)** Un model d'un conjunt de dades assigna una distribució de probabilitats a l'espai mostral del qual se suposa que s'extreuen les dades.

**2)** El model es pot fer servir per a assignar una longitud comprimida a les dades, fent servir nocions de teoria de la informació.

**3)** Es pot assignar una longitud al model mateix.

**4)** L'anàlisi òptima és aquella que té la suma de la longitud de les dades comprimides i la longitud del model més petita.

És a dir, es busca una especificació compacta mínima tant del model com de les dades, simultàniament. Goldsmith proposa dues heurístiques per a fer una anàlisi morfològica inicial que pugui servir com a punt d'inici de recerca de la descripció global més curta de la morfologia.

### 5.2.1. Heurístiques per a la segmentació de les paraules

El problema de la segmentació de les paraules és apropiat per a ser tractat amb *expectation-maximization* (EM). Cada paraula $w$ de longitud $N$ es pot analitzar de $N$ maneres diferents, tallant la paraula en base i sufix després de $i$ lletres, en què $1 \leq i \leq N$. A cada una d'aquestes anàlisis se li assigna una massa de probabilitat, que se suma per a tot el conjunt resultant de bases i sufixos. En cada iteració successiva, cada un dels $N$ talls en base i sufix es pondera per a aquesta probabilitat. La massa de probabilitat per a la base i el sufix en cada tall s'augmenta per una quantitat igual a la freqüència de la paraula $w$ per la probabilitat del tall. Després d'algunes iteracions, aproximadament quatre, les probabilitats estimades s'estabilitzen i cada paraula s'analitza d'acord amb el tall amb la probabilitat més alta. Aquesta aproximació, però, falla perquè sempre es prefereixen les anàlisis amb bases o, més sovint encara, sufixos formats per una sola lletra.

Per aquest motiu Goldsmith presenta dues heurístiques per a produir una anàlisi morfològica inicial. Aquesta anàlisi inicial ha de servir per a aplicar posteriorment el model d'MDL.

- **Primera heurística.** Goldsmith anomena aquesta primera heurística *take-all-splits* ('agafa totes les divisions'). Considera tots els talls d'una paraula de longitud $l$ en base més sufix $w_{1,i} + w_{i+1,l}$ en què $1 \leq i < l$.

- **Segona heurística.** Com que l'objectiu és identificar sufixos finals de paraules, Goldmith assumeix per convenció que totes les paraules acaben amb un símbol de final de paraula ("#"), i avalua els comptatges de tots els $n$-grames de longitud entre 2 i 6 lletres que apareguin a final de paraula. Com que suposa que els morfemes gramaticals no requereixen més de 5 lletres calcula fins el 6-grama. D'aquests $n$-grames calcula la informació mútua ponderada. Aleshores escull els 100 $n$-grames que presenten aquest paràmetre més alt com a conjunt de sufixos candidats.

### 5.2.2. Determinació de les signatures

En aquest punt totes les formes tenen assignat un tall òptim en base i sufix per a l'heurística inicial escollida. Es retenen únicament aquelles bases i sufixos que són òptims com a mínim per a una paraula. A cada base se li assigna la llista ordenada alfabèticament dels sufixos que apareixen amb aquesta: aquesta llista és la signatura. Posteriorment es fa una llista de totes les signatures amb les diferents bases associades.

### 5.2.3. Resultats del programa Linguistica sobre el corpus de prova

Si analitzem el conjunt de formes de prova amb el programa Linguistica obtenim la sortida que podem observar a la taula 5.

Taula 5. Anàlisi del programa Linguistica per al conjunt de formes de prova

| | |
|---|---|
| 3 | |
| (1) | ая.ое.ого.ой.ом.ому.ую.ые.ых.ый.ым.ыми 5 60 |
| | юмористичн абажурн багажн нов новоиспеченн |
| (2) | NULL.а.ах.ам.ами.е.ом.ов.у.ы 5 50 |
| | гарнитур клавицимбал меморандум мост завод |
| (3) | NULL.а.ах.ам.ами.е.ой.у.ы 5 45 |
| | карт мер обуз побед регат |

En aquest cas s'han derivat 3 signatures. En els tres casos s'han detectat correctament tots els sufixos i s'han assignat també correctament totes les bases. La signatura (1) correspon als adjectius, la (2) als substantius masculins i la (3) als substantius femenins. No s'han derivat les signatures corresponents al substantius neutres ni als verbs.

## Resum

En aquest mòdul hem fet una introducció als conceptes, metodologies i tècniques relacionades amb la morfologia computacional. També hem pogut treballar en l'àmbit pràctic amb un dels formalismes exposats: el formalisme de descomposició morfològica, ja que hem desenvolupat alguns programes en Python que demostren l'ús d'aquest formalisme.

Aquells alumnes interessats en el formalisme morfològic de dos nivells poden treballar amb el programa PC-KIMMO, que es pot descarregar de l'adreça *http://www.sil.org/pckimmo*.

# Glossari

**afix** *m* És un morf travat que es realitza com a una seqüència de fonemes o de grafemes.

**arrel** *f* És el morfema lèxic, lliure o lligat, comú a tot un paradigma flexiu o derivatiu un cop eliminats tots els afixos.

**base** *m* És el mot simple o el morfema lèxic al qual afegim afixos per a obtenir un paradigma flexiu o per a formar nous mots.

**fonema** *m* Element mínim del sistema sonor d'una llengua, desproveït de significat, però que té un caràcter constrastiu, és a dir, que són capaços de contrastar i distingir significats.

**grafema** *m* És la representació gràfica, és a dir, escrita, d'un morfema.

**lema** És la forma de referència d'un determinat paradigma.

**morf** *m* És la representació fònica o gràfica d'un morfema. Un morfema pot estar representat per un sol morf o per més d'un. En aquest segon cas parlem d'*al·lomorfs*.

**morfema** *m* És la unitat mínima recurrent amb significat, no descomponible en elements menors portadors de significat lèxic o gramatical.

**morfema gramatical o semàntic categorial** *m* Aquests morfemes es poden dividir en flexius i derivatius.

**morfema lèxic** *m* És un morfema amb significat lèxic, les arrels dels mots.

**morfema lligat** *m* És un morfema que ocorre només en combinació amb altres morfemes.

**morfema lliure** *m* És un morfema que pot constituir una paraula.

# Bibliografia

**Alegría, I.; Artola, X.; Sarasola, K. i Urkia, M.** (1996). «Automatic morphological analysis of Basque». A: *Literary & Linguistic Computing*, volum 11(4).

**Alshawi, H.** (ed.) (1992). *The Core Language Engine*. MIT Press.

**Badia, T.; Egea, A. i Tuells, A.** (1997). «CATMORF: Multi two-level steps for Catalan morphology». A: «ANLP 97».

**Beesley, K. i Karttunen, L.** (2000). «Finite-State Non-Concatenative Morphotactics». A: «SIGPHON-2000. Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology», (pàgs. 1–12). Luxembourg.

**Black, A.; Ritchie, G.; Pulman, S. i Russell, G.** (1987). «Formalisms for Morphographemic Description». A: «Proceedings of the 3rd. European ACL», (pàgs. 11–18). ACL.

**van den Bosch, A. i Daelemans, W.** (1999). «Memory-Based morphological analysis». A: «Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics», (pàgs. 285–292).

**Brent, M.** (1993). «Minimal generative models: A middle ground between neurons and triggers». A: «Proceedings of the 15th Annual Conference of the Cognitive Science Society», (pàgs. 28–36). Hillsdale, NJ: Lawrence Erlbaum Associates.

**Chomsky, N. i Hall, M.** (1968). *The Sound Patterns of English*. New York: Harper and Row.

**Clark, A.** (2001). *Unsupervised Language Acquisition: Theory and Practice*. Tesi Doctoral, University of Sussex.

**Clark, A.** (2002). «Memory-Based Learning of Morphology with Stochastic Transducers». A: «Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)», (pàgs. 513–520).

**Creutz, M. i Lagus, K.** (2002). «Unsupervised Discovery of Morphemes». A: «Workshop on Morphological and Phonological Learning», Philadelphia, PA: Association for Computational Linguistics.

**Déjean, H.** (1998). «Morphemes as necessary concept for structures discovery from untagged corpora». A: «Workshop on Paradigms and Grounding in Natural Language Processing», (pàgs. 295–299).

**Dzerovski, S. i Erjavec, T.** (1997a). «Induction of Slovene nominal paradigms». A: «Inductive Logic Programming, 7th International Workshop, ILP-97», volum 1297 de *Lecture Notes in Computer Science*, (eds.) Lavrac, N. i Dzeroski, S., (pàgs. 17–20). Berlin: Springer.

**Dzerovski, S. i Erjavec, T.** (1997b). «Learning Slovene Declensions with FOIDL». A: «Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks», (eds.) Daelemans, W.; van den Bosc, A. i Weijters, A., (pàgs. 49–60). Praga.

**Fisher, R.** (1925). «Theory of statistical estimation». A: «Proceedings of the Cambridge Philosophical Society», (pàgs. 700–725).

**Flenner, G.** (1994). «Ein quantitatives Morphsegmentierungssystem für spanishe Wortformen». A: «Computatio Linguae II», (ed.) Klenk, U., (pàgs. 31–62). Stuttgart: Steiner Verlag.

**Fromkin, V. i Rodman, R.** (1988). *An introduction to language*. Hot, Rinehart and Winston, Inc.

**Gaussier, E.** (1999). «Unsupervised learning of derivational morphology from inflectional lexicons». A: «Proceedings of the Workshopon Unsupervised Learning in Natural Language Processing», (pàgs. 24–30). Association for Computational Linguistics.

**Golding, A. i Thompson, H.** (1995). «A morphology component for language programs». A: *Linguistics*, (pàgs. 263–284).

**Goldsmith, J.** (1999). «Unsupervised Learning of the Morphology of a Natural Language». http://humanities.uchicago.edu/faculty/goldsmith/Automorphology.

**Goldsmith, J.** (2001). «Unsupervised Learning of the Morphology of a Natural Language». A: *Computational Linguistics*, volum 27(2): pàgs. 153–198.

**Hafer, M. i Stephen, F.** (1974). «Word segmentation by letter successor varieties.» A: *Information Storage and Retrieval*, volum 10.

**Harris, Z.** (1955). «From phoneme to morpheme». A: *Language*, volum 31: pàgs. 190–222. Reprinted in Harris 1970.

**Jacquemin, C.** (1997). «Guessing morphology from terms and corpora». A: «Proceedings of SIGIR 97», (pàgs. 156–165). Philadelphia: ACM.

**Janssen, A.** (1992). «Segmentierung französischer Wortformen in Morphe ohne Verwendung eines Lexikons». A: «Computatio Linguae», (ed.) Klenk, U., (pàgs. 74–95). Stuttgart: Steiner Verlag.

**Johnson, C.** (1972). *Formal aspects of Phonological Description*. Mouton, The Hague.

**Kaplan, R. i Kay, M.** (1981). «Phonological rules and finite-state transducers». A: «Linguistic Society of America Meeting Handbook. 56th Anual Meeting.», New York.

**Karlsson, F. i Karttunen, L.** (1997). «Subsentencial Processing». A: «Survey of the State of the Art in Human Language Technology», (ed.) Cole, R. Giardini Editori e Stampatori.

**Karttunen, L.** (2000). «Applications of Finite-State Transducers in Natural Language Processing». A: «Proceedings of CIAA-2000. Lecture Notes in Computer Science».

**Karttunen, L. i Beesley, R.** (2001). «A Short History of Two-Level Morphology». http://www.ling.helsinki.fi/ koskenni/esslli-2001-karttunen/.

**Karttunen, L.; Chanod, J.-P.; Grefenstette, G. i Schiller, A.** (1997). «Regular Expressions for Language Engineering». A: *Journal of Natural Language Engineering*, volum 2(4): pàgs. 307–330.

**Kazakov, D.** (1997). «Unsupervised learning of naïve morphology with genetic algorithms». A: «Workshop Notes of the ECML/Minet Workshop on Empirical Learning of Natural Language Processing Tasks», (eds.) Daelemans, W.; van den Bosch, A. i Weijtera, A.

**Kazakov, D. i Manandhar, S.** (1998). «A hybrid approach to word segmentation». A: «Proceedings of the 8th International Workshop on Inductive Logic Programming (ILP-98)», volum 1446 de *LNAI*, (ed.) Page, C. D. Madison, Wisconsin, USA: Springer-Verlag.

**Kazakov, D. i Manandhar, S.** (2001). «Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming». A: *Machine Learning*, volum 43: pàgs. 121–162.

**Koskenniemi, K.** (1983). *Two-level morphology: a general computational model for word recognition and production*. Número 11 A Publications. University of Helsinki, department of General Linguistics.

**Ling, C.** (1994). «Learning the past tense of English verbs: the symbolic pattern vs. connectionist models». A: *Journal of Artificial Intelligence Research*, volum 1: pàgs. 209–229.

**Ling, C. i Marinov, M.** (1993). «Answering the connectionist challenge: a symbolic model of learning the past tense of English verbs». A: *Cognition*, volum 49: pàgs. 235–290.

**Manandhar, S.; Dzerovski, S. i Erjavec, T.** (1998). «Learning Multilingual Morphology with CLOG». A: «Lecture Notes in Artificial Intelligence», (ed.) Page, D., (pàgs. 135–144). Springer.

**Manning, C.** (1998). «The segmentation problem in morphology learning». A: «NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Language Learning», (ed.) Powers, D., (pàgs. 299–305). ACL.

**de Marcken, C.** (1995). *Unsupervised Language Acquisition*. Tesi Doctoral, MIT, Cambridge, MA.

**Martí, M.** (1988). *Processament Informàtic del llenguatge natural: un sistema d'anàlisi morfològica per ordinador*. Tesi Doctoral, Departament de Filologia Romànica. Facultat de Filologia de la Universitat de Barcelona.

**Matheson, C.** (1995). «Computational Morphology. An introduction to ALE-RA.» http://www.ltg.ed.ac.uk/projects/ledtools/ale-ra/ale-ra.html.

**Mitkov, R.** (ed.) (2003). *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

**Mooney, R. i Califf, M.** (1995). «Induction of first-order decision lists: Results on learning the past tense of English verbs». A: *Journal of Artificial Intelligence Research*, volum 3: pàgs. 1–24.

**Mooney, R. i Califf, M.** (1996). «Learning the past tense of English verbs using inductive logic programming.» A: «Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing», (eds.) Wermter, S.; Riloff, E. i Scheler, G. Springer Verlag.

**Moreno, J.** (1994). *Curso universitario de lingüística general. Tomo II: Semántica, pragmática, morfología y fonología*. Editorial Sintesis.

**Neuvel, S. i Fulop, S.** (2002). «Unsupervised Learning of Morphology Without Morphemes». A: «Proceedings of the Workshop on Morphological and Phonological Learning», Association for Computational Linguistics.

**Pereira, F. i Warren, D.** (1980). «Definite Clause Grammars for Language Analysis - a Survey of the Formalism and a Comparision with Transition Networks». A: *Artificial Intelligence*, volum 13: pàgs. 233–278.

**Plunkett, K. i Nakisa, R.** (1997). «A connectionist model of the Arabic plural system». A: *Language and Cognitive Processes*, volum 12(5/6): pàgs. 607–836.

**Quinlan, J.** (1979). «Discovering rules by induction from large collections of examples». A: «Expert Systems in the Micro Electronic Age», (ed.) **Michie, D.**, (pàgs. 168–201). Edinbourgh University Press.

**Radford, A.; Atkinson, M.; Britain, D.; Clahsen, H. i Spencer, A.** (2000). *Introducción a la lingüística*. Cambridge University Press.

**Rissanen, J.** (1978). «Modelling by shortest data description». A: *Automatica*, volum 14: pàgs. 465–471.

**Rissanen, J.** (1989). *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Co.

**Ruessink, H.** (1989). «Two level formalisms». A: «Utrecht Working Papers in NLP».

**Schone, P. i Jurafsky, D.** (2000). «Knowledge-free induction of morphology using latent semantic analysis». A: «Proceedings of the CoNLL-2000 and LLL-2000», (pàgs. 67–72). Lisboa.

**Schützemberger, M.** (1961). «A remark on finite state transducers». A: *Information and control*, volum 4: pàgs. 185–196.

**Snover, M. i Brent, M.** (2001). «A Bayesian Model for Morpheme and Paradigm Identification». A: «Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics», (pàgs. 490–498).

**Theron, P. i Cloete, I.** (1997). «Automatic acquisition of two-level morphological rules». A: «Proceedings of the Fifth Conference on Applied Language Processing», (pàgs. 103–110).

**Trost, H.** (2003). *The Oxford Handbook of Computational Linguistics*, capítol 2. Morphology. Oxford University Press.

**Tuson, J.** (ed.) (2000). *Diccionari de lingüística*. Vox.

**Viterbi, A.** (1967). «Error bounds for convulutional codes and an asymptotically optimum decoding algorithm.» A: *IEEE Transactions on Information Theory*, volum IT-13: pàgs. 1260–1269.

**Westermann, G. i Goebel, R.** (1995). «Connectionist rules of language». A: «Proceedings of the 17th annual conference of the Cognitive Science Society», (pàgs. 236–241).

**Yarowski, D. i Wicentowski, R.** (2000). «Minimally supervised morphological analysis by multimodal alignement». A: «Proceedings of the ACL», (pàgs. 207–216). Hong Kong.