

## PETIT MANUAL DE CQP (*Corpus Query Processor*)

### I. Cerques<sup>1</sup>:

#### 1. Per començar:

- Activar CQP: des del servidor, teclejar `cqp -e`
- \*0j0\* UN COP DINS DEL CQP, TOTES LES COMANDES HAN D'ACABAR AMB UN PUNT I COMA (;).
- Per veure quins corpus estan disponibles:  
`show;`
  - Per activar el corpus en què vulgueu fer la cerca:  
`NomCorpus; ---` Ex.: `BNC;` (per treballar amb el BNC)
  - Per veure els atributs que estan disponibles al corpus en què feu la cerca:  
`show cd;`

#### 2. Cerques amb expressions regulars:

- Per buscar paraules, s'han d'introduir entre cometes dobles:  
`"Paraula"; ---` Ex.: `"house";`
  - Si el que voleu són cadenes de paraules, simplement les escriviu en una sola línia, juxtaposades, amb un blanc entremig:  
`"Paraula1" "Paraula2" "Paraula3" (...) "Paraulan"; ---` Ex.: `"White" "House";`
- \*0j0\* El CQP cercarà tot el que poseu entre les cometes, o sigui que compte amb els blancs: si introduïu `"White"`, no trobarà res.
- \*0j0\* Les majúscules i minúscules són significatives: no us donarà els exemples de `"white house"` sense majúscula.
- Per cercar qualsevol paraula, utilitzeu `[]` (sense les dobles cometes!!!):  
Ex. `"give" [] "up";` us buscarà totes les ocurrències de `"give up"` amb una i només una paraula entremig: `"give it up"`, `"give him up"`, etc.
  - Si voleu més d'una paraula entremig, teniu diverses opcions:
    - a) repetir `[]`: `"give" [] [] [] "up";` *per obtenir tres i només tres paraules entremig.*
    - b) utilitzar l'estrella de Kleene o *wildcard* `*`: `"give" []* "up";` *us donarà totes les ocurrències de "give up" amb un nombre arbitrari de paraules entremig, inclòs el zero.*
    - c) utilitzar l'operador *plus* `+`: `"give" []+ "up";` *us donarà el mateix resultat que (b) però amb una paraula entremig com a mínim.*
    - d) podeu limitar el poder de (b) i (c) utilitzant l'opció *within*: `"give" []* "up" within 3 word;` *us donarà el mateix resultat que (b) però amb un nombre màxim de 3 paraules entremig de "give" i "up". També podeu fer servir la marca estructural de frase 's' (v. més avall): "give" []\* "up" within s; *us donarà tots els casos en què "give" i "up" apareixin a la mateixa oració.**
    - e) utilitzar l'operador de repetició `{m,n}` (*m* = nombre mínim, *n* = nombre màxim de paraules):  
`"give" []{0,3} "up";` *us donarà el mateix resultat que (d); "give" []{3} "up";* *el mateix que (a), i "give" []{1,} "up";* *el mateix que (c).*
- \*0j0\* Tots aquests operadors els podeu fer servir amb qualsevol element de cerca, no només amb `[]`.
- Disjunció: operador `|`:

---

<sup>1</sup> **Nota:** si no dic el contrari, totes les comandes han d'anar seguides d'un ENTER.

*Cadena1 | Cadena2 --- Ex.: "house" | "houses"; us donarà tots els casos de "house" en singular o en plural.*

- Opcionalitat: operador '?':

*Ex.: "houses?"; aquesta és una cerca equivalent a l'anterior, però més econòmica. L'operador '?' expressa opcionalitat de l'element immediatament anterior, per la qual cosa cal tenir-ne en compte l'àmbit; per exemple, si la cerca fos "houses"? el que seria opcional no seria la 's' sinó tota la paraula, "houses" (i, per tant, només tindria sentit si estés inclòs en una cerca més gran).*

- Comodí: operador '.': indica qualsevol caràcter

*Ex.: "house."; us donarà totes les paraules que comencin per "house" i acabin en qualsevol lletra, p. ex. "houses" o "housed". Si combineu el comodí amb l'operador d'opcionalitat ("house.?"), obtindreu també les ocurrences de "house".*

- Conjunció: operador '&' (s'exemplifica més avall)

- Negació: operador '!' (s'exemplifica més avall)

*\*0j0\* Si combineu operadors, heu de fer servir parèntesi per indicar l'ordre d'aplicació dels mateixos: no és el mateix la cerca "their" ("house" | "houses") que ("their" "house" | "houses").*

### 3. Operacions amb atributs:

- Fins ara, només hem considerat cerques amb paraules. Això no obstant, una de les utilitats dels corpus etiquetats (morfològicament i per lema, com el BNC, o també sintàcticament, com d'altres corpus) és el fet que es poden efectuar cerques pel que s'anomenen 'atributs', és a dir, per exemple, per la categoria morfològica o el lema. P.ex., si cerquem la paraula "surprise" ens sortiran tant verbs com noms. Si ho volem evitar, caldrà que li especifiquem quina categoria ("pos", *part of speech*) volem:

*Ex.: [word = "surprise" & pos = "V.\*"]; això vol dir que el que busquem és la paraula "surprise" quan és un verb (de fet, el que li diem és que l'etiqueta de 'pos' comenci per V, noti's la combinació del comodí amb l'estrella de Kleene); observem que la paraula també està tractada com un atribut, en aquest cas l'atribut 'word'; quan un atribut està marcat com a "default", es pot fer la cerca sense indicar explícitament de quin es tracta (més avall s'explica la manera de canviar l'atribut "default").*

- Una cerca amb uns resultats equivalents als anteriors, però fent servir l'operador de negació introduït més amunt, seria [word = "surprise" & pos != "N.\*"]

- Imaginem que volem buscar els casos en què aparegui el verb 'surprise' (en qualsevol de les seves formes): una manera de fer-ho és demanar

*[word = "surpris.\*" & pos = "V.\*"]; aquesta cerca ens dona totes les paraules que comencen per "surpris" i acaben en un número indefinit de lletres i que són verb.*

*Això no obstant, una manera més econòmica i més segura d'obtenir un bon resultat és fer la cerca per lema: [lemma = "surprise" & pos = "V.\*"];*

- Per visualitzar altres atributs (p.ex. categoria morfològica, "pos", i lema, "lemma") al resultat de la nostra cerca, hem d'introduir abans de fer-la o de veure-la, respectivament,

`show +pos; o show +lemma;`

*També es poden visualitzar altres atributs estructurals, com les marques de principi i final de l'oració, <s> i </s> (al BNC, aquestes marques corresponen a tot el que hi ha entre dos punts), de manera anàloga:*

`show +s;`

- Si treballem molt amb atributs que no són 'word' (p. ex., amb categories morfològiques, 'pos') podem canviar el *default* a aquest atribut:

`set DefaultNonBrackAttr NomAtribut;` (versió curta, equivalent: `set da NomAtribut;`)

*Ex.: set da pos; això vol dir que el que s'introduirà entre cometes a les cerques, sense explicitar [pos = ...], seran categories morfològiques (v. etiquetari del BNC).*

- Per veure totes les opcions que es poden parametritzar amb “set”, cal picar **set**; (també en mostra la versió curta)
- Per veure els atributs que estan codificats a un corpus en particular, cal picar **show cd**; (cd = context descriptor)

#### 4. Operacions amb els *matches*:

- Per limitar el nombre de *matches* que ens retorna el CQP, hi ha dues opcions:

a) l'operador 'cut':

*Cerca cut n; (n = nombre de matches)*

Ex.: "Buckingham" "Palace" cut 20; *ens donarà els primers 20 casos de "Buckingham Palace" que trobi al BNC.*

b) l'operació 'reduce' (amb aquesta ordre, cal fer-ho en dos passos)

*Cerca;*

reduce Last to *n*; (*n* = nombre de *matches*)

Ex.: "Buckingham" "Palace";

reduce Last to 20; *ens seleccionarà 20 casos de manera aleatòria.*

- Si es vol aplicar (b), el millor és fer que els resultats no apareguin per pantalla, cosa que farà guanyar molt de temps al CQP. Aquesta opció s'activa així:

set AutoShow no; (versió curta: set as no;)

- Si volem ampliar el nombre de paraules que entren dins el *match* (no el context!), hem d'utilitzar l'opció 'expand':

*Cerca expand to n word; ens ampliarà el match tant per l'esquerra com per la dreta al nombre n de paraules.*

*Cerca expand left to n word; només ens l'ampliarà per l'esquerra*

També podeu fer servir aquesta opció amb qualsevol etiqueta estructural. P. ex.:

*Cerca expand right to 1 s; ens ampliarà el match a tota l'oració que contingui l'element de la cerca, si l'atribut 's' està definit al corpus en qüestió.*

#### II. Opcions de presentació:

- Per veure el resultat de l'última cerca feta, cal introduir

cat; o cat Last;

- L'opció ja citada de no veure el resultat de la cerca per pantalla es desactiva en sortir i tornar a entrar al CQP o bé amb la instrucció inversa: set AutoShow yes;

- Podem definir el context de les cerques que volem veure per pantalla o que volem exportar a un arxiu de text:

set Context *n word*; (versió curta: set c *n word*) *ens ampliarà el context, tant per l'esquerra com per la dreta, al nombre n de paraules.*

set LeftContext *n word*; *només ens l'ampliarà per l'esquerra.*

set RightContext *s*; *ens ampliarà el context de la dreta a l'oració (si l'atribut 's' està definit al corpus en qüestió)*

#### III. Treballar amb subcorpus:

- Quan fem cerques, el que estem fent en realitat és crear subcorpus del corpus més gran. Per defecte, CQP assigna el nom "Last" al resultat de cada cerca que fem, i per tant sobreescrui el subcorpus cada cop. Això no obstant, podem salvar les nostres cerques per reutilitzar-les de diferents maneres. El primer que cal fer és donar nom al subcorpus que hem creat, cosa que es pot fer per dos camins:

a) *NomSubCorpus = Cerca*; --- Ex.: *prova1 = "White" "House" cut 20;*

b) *Cerca*;

*NomSubCorpus = Last*;

Ex.: "Buckingham" "Palace" cut 20;

```
prova2 = Last;
```

Un cop els hem donat nom, podem veure'ls amb la instrucció 'cat' ja introduïda, seguida pel nom del corpus: `cat prova1`;

- Si volem fer servir els resultats una altra sessió, a continuació hem de salvar-los (si no, s'esborren en sortir de CQP):

```
save NomSubCorpus; --- Ex.: save prova1;
```

- De vegades va bé, per guanyar temps, construir-se un subcorpus per fer-hi cerques específiques. Per exemple, la cerca NP ens donaria bona part dels sintagmes nominals amb determinant que apareixen al BNC (per fer aquesta cerca cal canviar l'atribut *default* a 'pos', com hem vist més amunt):

```
NP = ("AT.*"|"D.*") ("AV0"* ("AJ.*" | "V.G" | "V.N") | "N.*")*  
"N.*";
```

Un cop tinguem els resultats, podem efectuar-hi cerques simplement cridant el subcorpus, igual com cridem el BNC, cosa que farà canviar el "prompt":

```
BNC> NP;
```

```
BNC:NP> "their" "house" cut 10;
```

#### IV. Treballar amb arxius:

- Per exportar el resultat d'una cerca (i.e., un subcorpus) a un arxiu de text cal fer:

```
cat NomSubCorpus > "NomArxiu"; (aquest subcorpus pot ser simplement 'Last')
```

```
Ex.: cat prova1 > "prova1.txt"
```

\*0j0\* L'arxiu 'prova1.txt' es crearà al directori des d'on haguem cridat el CQP.

- Per afegir (*append*) els resultats d'una altra cerca al mateix arxiu, es fa igual però amb '>>' en lloc de '>': `cat prova2 >> "prova1"`;
- En lloc de treballar directament amb CQP, és molt útil fabricar-se un "batchfile", és a dir, un arxiu amb comandes que CQP llegirà i executarà seqüencialment. Un cop creat aquest arxiu (en format ASCII), es crida (des del servidor!) de la manera següent:

```
cqp -f NomArxiu
```

#### Apèndix. Informació general sobre el CQP:

(c) 1993-98 University of Stuttgart, IMS

cqp developed by Bruno Schulze 93-94 and Oliver Christ 94-96

further work by Stefan Evert 97- and Arne Fitschen 97-

<http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/>