Mémo CQL (langage d'interrogation pour TXM)

Introduction

CQL, CQP

CQL est l'acronyme de Corpus Query Language, c'est un langage d'expression de requêtes. Une expression CQL est une chaîne de caractères exprimant un motif linguistique (un mot, ou une suite de mots) à partir des valeurs de leurs propriétés (comme la catégorie grammaticale, le lemme, la forme graphique).
CQP est l'acronyme de Corpus Query Processor, c'est un module logiciel qui traite des requêtes : c'est un moteur de recherche qui permet de trouver toutes les occurrences correspondant à une équation CQL dans un corpus donné. Le moteur CQP a été développé à l'université de Stuttgart. Il est intégré à TXM où il assure les recherches d'occurrences et d'une façon générale toutes les opérations de sélection à l'intérieur du corpus. Il a été choisi pour l'excellent rapport de ses performances à la complexité des requêtes traitées.

Les requêtes dans TXM : requêtes simples, requêtes assistées, requêtes avancées

CQL est donc un langage formel, avec un lexique et une syntaxe d'opérateurs, qui forment un métalangage permettant de combiner des éléments pour la recherche de motifs structurés.
L'apprentissage du langage CQL n'est pas un passage obligé pour utiliser TXM, mais c'est en langage CQL qu'on a le mode d'expression de motifs le plus riche.
Si l'on saisit un mot dans la zone de requête, c'est interprété comme la recherche des mots présentant exactement cette graphie dans le corpus. Cela permet déjà un certain nombre de recherches simples. Mais on perçoit assez vite deux limites : d'une part, on reste à la « surface » du texte, on ne tire aucun parti des autres informations linguistiques encodées dans le corpus (lemme, catégorie grammaticale, etc). D'autre part, on est rivé à l'empan exact d'un mot : la formulation de la recherche ne peut se faire ni sur une partie du mot (son début par exemple), ni sur des expressions en plusieurs mots -alors que cela devient possible en utilisant CQL.
Le logiciel TXM comporte un assistant à l'écriture de requêtes, accessible via une icône « baguette magique » à gauche du champ de saisie de la requête. Cet assistant permet d'exprimer une recherche à l'aide de menus déroulants plus intuitifs si l'on est peu familier des langages de requête. En revanche, il ne permet pas d'exprimer autant de choses que le langage CQL, qui reste beaucoup plus souple et plus complet. La connaissance de CQL est donc utile pour avoir les possibilités d'expression les plus larges et les plus précises.
En pratique, on peut apprécier de combiner l'utilisation de l'assistant avec la connaissance du langage CQL. L'assistant peut faciliter l'écriture d'une première version de la requête. La connaissance de CQL permet ensuite de bien comprendre l'équation et de l'ajuster ou de l'affiner si nécessaire.

Dynamique de la construction d'une requête

Une requête se met au point : entre ce qu'on veut repérer (que l'on pense avoir exprimé dans la requête), et ce qu'on trouve effectivement dans le corpus, il y a souvent un écart qui demande à être corrigé. Il est de toutes façons toujours sage de vérifier la portée effective, dans le corpus choisi, de la requête utilisée, avant de l'utiliser pour un calcul statistique.
L'apprentissage et l'utilisation de CQL font donc un usage central de la fonctionnalité Index de TXM. La fonctionnalité Index permet de lister toutes les formes correspondant au motif dans le corpus. On peut les parcourir soit par importance quantitative décroissante (tri par fréquence décroissante, qui est la manière dont se présente le résultat par défaut), soit par ordre alphabétique, ce qui peut faciliter la lecture en regroupant les réalisations de forme proche.
Le parcours de cette liste des configurations trouvées met en évidence les formes indésirables ; en revanche il ne dit rien des formes qui seraient pertinentes mais qui, ne correspondant pas formellement à la requête, n'ont pas été repérées. Méthodiquement, on recommande donc toujours, quand on a un motif linguistique à recherche, de commencer par l'exprimer de façon très ouverte, de veiller à minimiser les a priori qui pourraient être réducteurs. L'examen des occurrences correspondantes trouvées guide alors sur la manière d'ajouter alors peu à peu des contraintes permettant de cibler les formes pertinentes et d'écarter les formes non voulues.

Utilisation pédagogique des exemples

Les exemples ci-après ont été choisis pour illustrer les possibilités de CQL qui nous paraissent les plus utiles : les soumettre avec la fonctionnalité Index pour bien voir leur effet. Ils ont été conçus pour être lancés sur le corpus Voeux (http://sourceforge.net/projects/txm/files/corpora/voeux/voeux-bin-0.6.zip/download). Le corpus Discours est quelquefois utilisé en complément si nécessaire. Les exemples sur fond gris sont plus complexes et peuvent être ignorés dans un premier temps.

Recherche simple [niveau 1 (infralexical) : les valeurs]

Recherche d'un mot

`bonheur`
`l'amitié`
`l'`
`amitié`
`aujourd'hui`
`parce que`
	Pour chercher un mot donné il suffit de saisir sa graphie.
	L'expression CQL doit correspondre exactement à une unité telle que découpée par la segmentation lexicale, une unité lexicale n'est pas forcément une chaîne de caractères entre deux blancs. Voir par exemple aussi les différences entre Voeux et Discours pour les unités ci-contre.

`ami` `amiti`	*Une partie d'un mot ne rapporte aucun résultat, l'expression doit correspondre à un mot entier attesté dans le corpus.*

`bonheur`
`"bonheur"`
`[word="bonheur"]`
	Trois façons équivalentes d'exprimer une recherche sur une graphie :
	- la graphie telle quelle
	- la graphie entre guillemets doubles droits
	- l'usage des crochets et du mot réservé « word ».
	Les moyens les plus verbeux montreront leur utilité dans des cas plus complexes.

`[word="parce que"]` `[word=" bonheur "]`	*Un blanc à l'intérieur des guillemets est significatif (partie intégrante de la graphie). Le guillemet doit être collé à la graphie cherchée (sans espace supplémentaire).*
`[ word = "bonheur" ]`	*Les blancs à l'extérieur des guillemets sont non significatifs et peuvent être utilisés pour faciliter la lecture.*

Variantes d'écriture

`"gouvernement"%c`	*Neutralisation de la casse (majuscules/minuscules). Les guillemets sont obligatoires.*
`"Etat"%d`	*Neutralisation des signes diacritiques (accents, cédille, etc.).*
`"franc.*"%cd`	*Les deux neutralisations peuvent être cumulées.*

Troncature et joker

`libertés?`
`âgé?e?s?`
`"premiere?s?"%d`
	Le point d'interrogation porte sur le caractère qui précède et signifie qu'il est facultatif (0 ou 1 fois). Il peut se placer n'importe où. C'est utile notamment quand le corpus n'est pas lemmatisé, ou que la qualité de la lemmatisation est insuffisante.

`nation.*`	*Point étoile à la fin = « mot qui commence par … » . Point = « un caractère, n'importe lequel ».*
`.*patri.*`	*Etoile = « 0 à n fois, n aussi grand qu'on veut ». Utile pour chercher un radical.*
`.+patri.*`	*Signe plus = « 1 à n fois ». Ici on impose qu'il y ait un préfixe.*
`.*ables?` `in.*ables?`	*Ces opérateurs se plaçant n'importe où, on peut chercher des mots partageant les mêmes affixes, le radical variant librement.*
`"i[mn].*ables?"`	*Les crochets sont pratiques pour indiquer l'ensemble des lettres possibles, une seule devant être choisie.*
`.*`	*Zéro à n caractères, n'importe lesquels. Cette expression attrape tous les mots.*
`.* .*`	(dans Discours) *Graphies incluant un blanc (au moins).*
`.`	*Mots formés d'un seul caractère.*
`…`	*Mots de longueur trois.*

Ponctuations

`\.`	*Les caractères spéciaux (opérateurs), doivent être « endormis » en les précédant d'une barre oblique descendante, si on veut pouvoir les considérer eux-mêmes comme des caractères que l'on recherche. Ce n'est pas le cas de toutes les ponctuations : ex. ici mots terminés par une apostrophe.*
`\?`	
`.*'`	

Classes de caractères

`.+\p{P}`	*Mot terminé par une ponctuation : permet d'attraper aussi les apostrophes obliques (souvent originaires de Word et qu'on ne peut pas saisir facilement au clavier dans TXM).*
`\p{Lu}+`	*Mot composé de majuscules (y compris diacritiques). Voir FAQ pour autres classes.*

Alternative

`paix|guerre`	*OU, alternative non exclusive. Élargit la recherche à des variantes de formulation.*
`(inter|supra)nation.*`	*Peut s'utiliser à l'intérieur du mot, avec des parenthèses pour délimiter sa portée.*
`(inter|supra)?nation.*`	*Des opérateurs de facultativité ou répétition peuvent porter sur la parenthèse.*

Recherche sur les propriétés [niveau 2 (lexical) : les propriétés]

Introduction

Jusqu'alors, les recherches effectuées portaient sur la forme graphique des mots, qui est enregistrée dans la propriété *word* : [word="bonheur"] signifie qu'on recherche la valeur *bonheur* de la propriété *word*, correspondant à la forme graphique. Mais, lorsque le corpus est enrichi, les mots portent d'autres informations que leur seule graphie, sous la forme d'autre propriétés. Les requêtes peuvent alors porter sur d'autres propriétés des mots (et les combiner).
La graphie étant une propriété (presque) comme les autres, tout ce qu'on a vu dans la section précédente s'applique aux valeurs de propriété quelle que soit la propriété, sauf l'écriture simplifiée.
Pour interroger sur les propriétés il faut connaître leur nom et leurs valeurs. En effet, le nom des propriétés dépend de l'import du corpus : dans tel corpus la propriété qui enregistre le lemme est *lemma*, dans tel autre *frlemme*, dans tel autre encore *ttlemme*, etc. De même, les valeurs des catégories grammaticales dépendent du jeu d'étiquettes utilisé. Dans TXM en version locale, la fonction Description montre quelles propriétés sont disponibles et donne pour chacune d'elle un aperçu de quelques valeurs attestées (sur les premières occurrences du corpus). La fonction Lexique permet de lister exhaustivement les valeurs d'une propriété attestées dans le corpus. Dans la version locale, un double-clic sur une de ces valeurs permet de voir son usage en contexte (dans une concordance). Ceci étant il est utile d'avoir les tables descriptives des jeux de catégories utilisés pour le corpus sur lequel on travaille.

Recherche sur une propriété

`[frlemma="beau"]`	*Rechercher un lemme permet de désigner un mot sous ses formes (très) variables. Il faut expliciter sur quelle propriété on travaille, la formulation à crochets devient nécessaire.*
`[frlemma="faire"]`	
`[frlemma="je"]`	*Le lemme « je » recouvre ici ses formes élidées ou avec majuscule initiale.*
`[frpos="ADV"]`	*De même, on peut chercher sur d'autres propriétés, comme la catégorie grammaticale.*
`[frpos="VER.*"]`	*La valeur que prend la propriété peut utiliser les mêmes opérateurs que précédemment, par ex. pour reconstruire des catégories en regroupant des étiquettes.*
`[frpos="NOM|NAM|VER.*|ADJ"]`	
`[frlemma=".*|.*"]`	*Ici la barre verticale fait partie intégrante de l'étiquette (ambiguïtés non résolues par* TT*).*

Alternative (2)

`[frpos="NAM|NOM"]`	*Il y a plusieurs manières d'exprimer l'alternative, plus ou moins factorisées.*
`[frpos="N(A|O)M"]`	*La barre verticale est l'opérateur le plus général, sa portée peut être ciblée*

	par des parenthèses.
`[frpos="N[AO]M"]`	*Les crochets ne sont utilisables que pour une alternance sur un seul caractère, mais facilitent l'expression d'un large choix*
`"[aeiouy]+"`	(dans Discours) *ou d'une gamme.*
`[pos=".*[1-3].*"]`	(dans Discours) *Le chapeau est une négation : ensemble des caractères interdits sur la position.*
`[pos="[^12]*"]`	
`[frpos="VER:(futu|cond|subi)"]`	*Alternance sur des séquences de caractères (de longueurs identiques ou non) : seule la barre verticale est utilisable.*

Combinaison d'informations

`[frlemma="pouvoir" & frpos="NOM"]`	*Désambiguïsation catégorielle d'un lemme.*
`[frpos="ADV" & word=".*ment"]`	*Croisement d'une catégorie et d'un trait morphologique.*
`[frlemma="liber.*"%d & frlemma!="libéral"]`	*Exclusion de cas non souhaités.*
`[frpos="NOM" & word!=".*\p{P}"]`	*Post-taitement des erreurs de segmentation.*
`[pos!="NA|pon" & pos!=fropos]`	(dans la BFM) *Comparaison directe à une autre propriété.*

Recherche d'un motif de plusieurs mots [niveau 3 (supralexical) : séquences d'unités lexicales]

Succession de mots

`[word="réduction"] [word="du"] [word="temps"] [word="de"] [word="travail"]`
Paire de crochets = mot.

```
"réduction" "du" "temps" "de" "travail"
[frlemma="réduction"] "du" "temps" "de" "travail"
```
Notation allégée possible si l'on ne travaille que sur des graphies. Mélange possible.

`[frpos="NOM"][frlemma="de"][frpos="NOM"]`
Usage avec des catégories (patron).

```
[frpos="NOM"][frlemma="de"][frlemma="le"]?[frpos="NOM"]
[frpos="NOM"]([frlemma="de"][frlemma="le"]|[frlemma="du"])[frpos="NOM"]
[frpos="DET.*"][frpos="ADV"]?[frpos="ADJ"]+ [frlemma="année"]
```
On retrouve à ce niveau 3 les opérateurs vus au niveau 1, pour gérer les variations.

Traitement des insertions

`[frlemma="il"][][frlemma="y"][frlemma="avoir"]`	*Une unité lexicale quelconque (joker de mot).*
`[frlemma="il"][]?[frlemma="y"][frlemma="avoir"]`	*Insertion facultative.*
`[frlemma="il"][][][][frlemma="y"][frlemma="avoir"]`	*Distance de trois unités lexicales.*
`[frlemma="il"][]{0,3}[frlemma="y"][frlemma="avoir"]`	*Distance de zéro à trois.*
`[frlemma="paix"][]{0,10}[frlemma="monde"]`	*Distance de 0 à 10, deux formulations équivalentes.*
`[frlemma="paix"][]*[frlemma="monde"] within 10`	

Si l'on utilise []* il faut absolument borner l'expansion.

```
[frlemma="je"][frpos!="V.*"]*[frlemma="souhaiter"][frpos!="V.*"]*[frlemma="année"] within 25
```
Distances avec mots exclus, contrôle davantage syntaxique.

```
[lemma="je"][pos!="V.*"]*[lemma="souhaiter"][pos!="V.*"]*[lemma="année"] within s
```
(dans Discours) *Empan sur structure (si disponible)*

`[lemma="République"][]*[lemma="France"] within 2s` *(dans* Discours*) Structure multipliée.*

Etude distributionnelle

`[frlemma="très"][]`	*On prend un motif (contexte), et on rend variable une place, soit complètement librement,*
`[frpos="NOM"][frlemma="français"]`	*soit avec une indication de catégorie.*
`[frlemma="ne"][frpos="VER.*"]`	*Recherche des verbes avec négation.*

```
[frlemma="ne"]([frpos!="VER.*|NOM|ADJ"]|[frlemma="être|avoir"])*[frpos="VER.*" &
```

```
frlemma!="être|avoir"] within 10
```
Idem, plus affinée.

Alternatives

```
([word="président"%c][][][word="république"%c]|[word="chef"%c][][][word="état"%cd])
```
Expressions.

```
([frlemma="paix"][]*[frlemma="monde"]|[frlemma="monde"][]*[frlemma="paix"]) within 10
([frlemma="travail.*"][]*[frlemma="famil.*"]|[frlemma="famil.*"][]*[frlemma="travail.*"]) within 20
```
Cooccurrences.

Informations contextuelles

Utilisation des structures

`<s>[pos="V.*"]`	(dans Discours) *Verbes qui commencent une phrase.*
`<s>[pos="V.*"] expand to s`	(dans Discours) *Phrases qui commencent par un verbe.*
`<s>[]{1,5}</s>`	(dans Discours) *Phrases d'au plus cinq mots.*
`[pos="Vmsm.*"] expand to s`	(dans Discours) *Phrases contenant un motif donné (ici subjonctif imparfait).*

Utilisation d'une propriété sur une structure

`[word="Algérie" & _.text_loc!="dg"]` *« Algérie » dans un texte dont le locuteur n'est pas De Gaulle.*

Lien entre deux mots

```
a:[frpos="NAM|NOM|ADJ|VER.*" & word!=".*\p{P}"][]*[word=a.word] within 10
```
Répétition, accord,...

Lien d'alignement entre corpus parallèles

On dispose d'un corpus latin CorpusLAT aligné avec un corpus d'ancien français CorpusFRO (textes existant dans les deux langues, en relation de traduction). Les requêtes suivantes sont effectuées sur CorpusLAT.

`[lemme="HIC"] :CorpusFRO [lemme="CIST"]`	*Occurrences du lemme HIC pour lesquelles on trouve le lemme CIST dans le passage aligné en ancien français.*
`[lemme="HIC"] :CorpusFRO ![lemme="CIST"]`	*Occurrences du lemme HIC pour lesquelles on ne trouve pas le lemme CIST dans le passage aligné en ancien français.*
`[lemme="HIC"] expand to seg :CorpusFRO [lemme="CIST"]`	*Segments contenant le lemme HIC et pour lesquels on trouve le lemme CIST dans le segment aligné en ancien français.*
`[] expand to seg :CorpusFRO [lemme="CIST"]`	*Segments latins alignés avec ceux contenant le lemme CIST en ancien français (construction d'un sous-corpus pour calcul de résonance).*
`<seg>[lemme!="HIC"]*<seg> :CorpusFRO [lemme="CIST"]`	*Segments ne contenant pas le lemme HIC et pour lesquels on trouve le lemme CIST dans le passage aligné en ancien français.*