Lecture 9 CQPweb: searching the CRPC 21 Nov 2012

Simple Query Language

Regular expressions

A regular expression is a way of characterizing a string, you can view it as a pattern or a template in which you use wildcards to leave certain characters unspecified.

Wildcards	*example*	*matches with*
? a single arbitrary character	gat?	gato, gata
* zero or more characters	*mente	mente, absolutamente, provavelmente, etc.
+ one or more characters	+mente	absolutamente, provavelmente, etc. (but not: *mente*)
	????+ato	candidato, colonato
	*ou+ou	roupousou, roubou

Simple Query Syntax uses a set of characters as meta-characters:

? * + [ ] ( ) { } , : @ / _ - < >

To query for the literal meaning of these characters, use a backslash in front.
E.g. to look for a question mark, type: \?

Query type	*example*	*matches with*
Alternatives: between square brackets	lind[o,a]	lindo, linda
Two alternatives followed by exactly 1 character	lind[o,a]?	lindos, lindas
Two alternatives followed by: 's' or nothing	lind[o,a][s,]	lindo, linda, lindos, lindas
Two alternatives followed by zero or more characters	lind[o,a]*	lindo, lindos, lindamente, lindinho, lindoso, lindano, etc.
Ignore accents (:d)	e:d	e, é

Part-of-speech tags

You can search for a part-of-speech (POS) tags, or a combination of a word with a POS tag using '_'. The list of tags can be found on the last page of this manual.

Query description	*example*	*matches with*
Word - POS combination	desse_V	(verb) desse
Only POS	_IND	(indefinites) algo, nada, ninguém, outras, etc
Word string followed by zero or more character combined with POS	ante*_V	antecipar, antedatar, etc
	_INF*	INF, INFAUX

Notice that the tags have been assigned automatically and the corpus may contain errors.

Lemmas

You can search for a lemma or root form of a word by using curly brackets.

Query description	*example*	*matches with*
lemma	{poder}	poder, posso, podes, podia, etc
lemma with POS tag	{poder}_CN	poder, poderes
Word string followed by zero or more character combined with POS	ante*_V	anteceder, antecipar, antedatar, etc

Notice that the lemma tags have also been assigned automatically and the corpus may contain errors.

Word sequences

You can also search for multiple words. Notice that:

- punctuation marks are split from words and are separate tokens
- special characters need a backslash
- you can combine + and * to define a sequence of arbitrary words in your query. E.g. the pattern +** represents a sequence of one to three tokens.

Query description	*example*	*matches with*
Adjective followed by the lemma of the noun 'jantar'	*_ADJ {jantar}_CN	célebre jantar, breve jantar, grandes jantares, bom jantar, etc.
The word 'se' followed by an optional word and a comma	se * \,	'se trata ,', 'se ,', 'se vê ,', 'se calhar ,' etc
The lemma 'célebre' followed by the lemma of the noun 'jantar'	{célebre} {jantar}_CN	célebre jantar, célebres jantares
	{de} +** jantar	de estar presente num jantar ,de fazer um jantar, de nosso jantar, etc

Search of contracted elements (no, naquele, do, etc…)

Contractions of two words are annotated with double POS tags and lemmas. For example "no" has POS-tag "PREP+DA" and lemma "em+o". Below are some examples of how to search for these particular words, the '+' character is a meta- character, therefore you need to use a backslash.

Wildcards	*example*	*matches with*
To search for a contracted form, use '\+*'	{em\+*}	no,nas, naquele, etc.
Contracted forms followed by 'o'	{em\+o}	no, nos
To find both contracted and uncontracted forms, the '|' means "or".	({em\+}|{em})	em, no, nos, na, naquele, etc.

Advanced lexico-grammatical patterns

Use regular expression notation for alternatives, optional elements and repetition within a sequence:

(_ADJ)? optional adjective — e.g. **o (_ADJ)? cão**
(_ADJ)* zero or more adjectives (optional)
(_ADJ)+ one or more adjectives (non-optional)
(_ADJ){2,4} between two and four adjectives
(_ADJ){2,} at least two adjectives

(…|…|…) matches one of the alternatives indicated by …
(…|…|…)* alternatives with repetition (optional)
(…|…|…)+ alternatives with repetition (non-optional)
(…|…|…){2,4} between 2 and 4 repetitions of the given alternatives (may be mixed in any order)

• Regular expression notation can be nested to match complex patterns:
    the (most _AJ0 | _AJS) {man}
        e.g. the biggest men, the most attractive man, …
    the (most (_AV0)? _AJ0 | (_AV0)? _AJS) {man}
        e.g. the very richest men, the most supremely stupid men, …
• Complex syntactic patterns can be formed, e.g. for a prepositional phrase:
    _{PREP} (_{ART})? ((_{ADV})? _{A})* _{N}
    "a preposition; followed by an optional article; followed by any number of adjectives (zero or more), each of which may optionally be preceded by an adverb; followed by a noun"

Sentences

Sentence beginning with *o cão*	<s> o cão
Sentence ending with *o cão*	o cão _PNT </s>
List of all sentences	<s> (+)+ </s>

Proximity queries

cão <<s>> gato cão and gato in the same sentence
cão <<3>> gato cão and gato within range of 3 tokens
cão <<5<< gato gato … cão (within 5 tokens)
cão >>5>> gato cão … gato (within 5 tokens)

cão <<5>> (gato <<3>> leão)

CQP syntax

Noun Phrases

Version 2.2 of the CRPC has been tagged with Noun Phrases (NPs). You can query those NPs provided you use the CQP syntax. Here are a few examples:

All NPs: (this will take a very long time!): **<np> []* </np>;**

Only the word *gato* (no case distinction) as NP: **<np> [word="gato"%c] </np>;**

NPs with exactly 3 words: **<np> []{3} </np>;**

V at the start of a NP: **<np> [pos = "V"];**

Eles at the start of a NP: **<np> [word = "Eles"];**

The lemma *comer* at the start of a NP: **<np> [lemma = "comer"];**

V at the end of a NP: **[pos = "V"] [pos = "PNT"]? </np>;**

NP with at least 3 adjectives: **<np> []* ([pos="ADJ.*"] []*){3,} </np>;**

Sentences that start and end with a NP: **<s><np>[]*</np> []* <np>[]*</np></s>;**

CN that is not contained in a noun phrase: **[(pos = "CN") & !np];**

Sequence of two singular nouns within the same NP: **[pos="CN"] []* [pos="CN"] within np;**