

# Extensions to the CQL & Sketch Engine infrastructure

Miloš Jakubíček

Lexical Computing Ltd.

`milos.jakubicek@sketchengine.co.uk`

SKEW2, 17. 3. 2011

# Obsah

**1** CQL extensions

**2** Sketch Engine infrastructure

# Today's Corpora in SkE

- **LARGE** (= billions of tokens, and it's going to be worse)
- complex multi-level multi-value annotation
- wide range of languages
- growing demand on complex searching – moving from morphology to syntax and semantics
- search API for automatic information retrieval and post-processing in particular applications needed

# CQL

- = Corpus Query Language (Christ and Schulze, 1994)
- positions and positional attributes: `[attr="value"]`
- structures and structural attributes: `<str attr="value">`
- example:

```
[word=".*ing" & tag="V.*"]  
  <doc id="20[5-9].*">
```

- established a within `<str/>` query:

```
[tag="N.*"]+ within <s/>
```

and alternative meet/union query:

```
(meet [lemma="take"] [tag="N.*"] -5 +5)  
  (union (meet ...) (meet ...))
```

# CQL in Manatee/Bonito

- enhancements and differences to the original CQL syntax
- within <query> and containing <query>
- meet/union (sub)query
- inequality comparisons
- frequency function

## within/containing queries

- searching for particles:

```
[tag="PR.*"] within [tag="V.*"] [tag="AT0"]?  
[tag="AJO"]* [tag="(PR.?|N.*)"] [tag="PR.*"]  
within <s/>
```

- searching for a Czech idiom “hnout někomu žlučí” (“to get somebody’s goat”):

word-by-word translated as:

*hnout* “move” [V, infinitive]

*někomu* “somebody” [N, dative]

*žlučí* “bile” [N, instrumental].

```
<s/> containing [lemma="hnout"] containing  
[tag=".*c3.*"] containing [word="žlučí"]
```

# within/containing queries

- structure boundaries: begin: `<str>`, whole structure: `<str/>`, end: `</str>`
- **changes**: within `<str>` not allowed anymore, use within `<str/>`

## meet/union queries

- combined with regular query: <s/>

```
containing (meet [lemma="have"] [tag="P.*"] -5 5)  
containing (meet [tag="N.*"] [lemma="blue"])
```

- **changes:** meet/union queries can be used on any position, they can contain labels and no MU keyword is required (and deprecated):

```
(meet 1:[] 2:[]) & 1.tag = 2.tag
```



# Inequality comparisons

- former comparisons allowed only equality and its negation:  
`[attr="value"]` `[attr!="value"]`
- inequality comparisons implemented: `[attr<="value"]`  
`[attr>="value"]` `[attr!<="value"]` `[attr!>="value"]`
- intended usage:

`[tag="AJ.*"]` `[tag="NN.*"]` within `<doc year>="2009">`

- sophisticated comparison performed on the attribute value:  
`<doc id<="CC20101031B">` matches e.g. BB20101031B,  
CC20091031B, CC20101030B CC20101031A.

# Fixed string comparisons

- normally the CQL values are regular expressions
- sometimes this is not desirable (batch processing needs escaping of metacharacters)
- new `==` and `!=` operator introduced for fixed strings comparison
- no escaping needed except for `"'"` and `'\"'`
- examples: `".", "$", " "` matches a single dot, dollar sign and tilda, respectively, `"\\n"` matches a backslash followed by the character `n`,

# Frequency function

- a frequency constraint allowed in the global conditions part of CQL:

`1:[tag="PP.*"] 2:[tag="NN.*"] & f(1.word) > 10`

# Performance evaluation

**Table:** Query performance evaluation – corpora legend: ○ BNC (110M tokens), ● BiWeC (version with 9.5G tokens), \* Czes (1.2G tokens)

query	# of results	time (m:s)
○ [lemma="time"]	179,321	0.07
○ [lemma="t.*"]	14,660,881	3.12
○ Ex: particles	1,219,973	33.36
● Ex: particles	97,671,485	32:26.48
* Ex: idioms	66	1:6.86
○ Ex: meet/union	3	8.47
● Ex: meet/union	1457	7:13.12

# Sketch Engine infrastructure

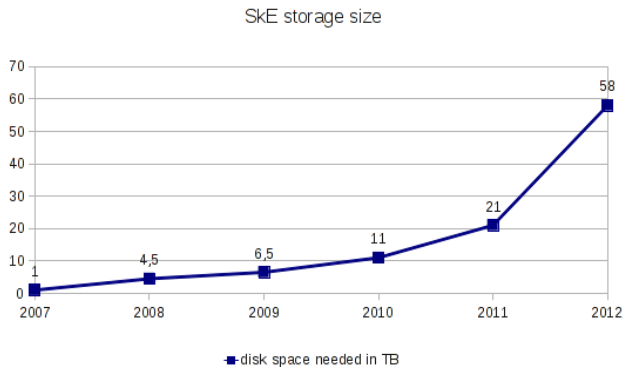
- = servers and their equipment
- what do we need?
- most companies either store lots of data, but don't need fast access (e.g. backups, logs) or store quite small amount of data accessible fastly (information systems, databases)
- we need both + lots of memory and fair number of CPU cores
- we need to manage concurrent access

# Sketch Engine in numbers

by 2011:

- > 8500 registered users
- 191 preloaded corpora, 38G tokens, 47 languages
- 4,335 user corpora, 2.5G tokens
- ca. 30,000 requests per day

# SkE storage size



# SkE 2007

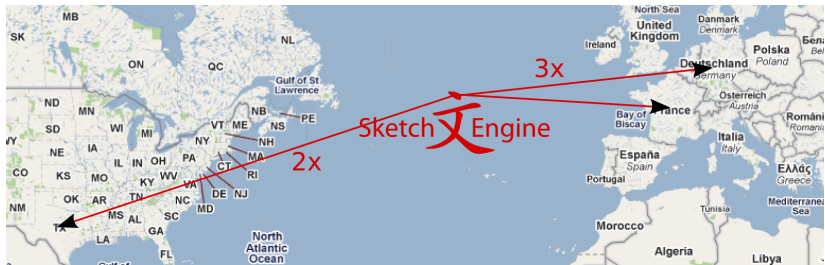




# SkE 2008



# SkE 2010





# SkE 2011



# Thank you!

# Thank you for your attention!