Keresés korpuszban

PPKE ITK

Sass Bálint
Nyelvtudományi Kutatóközpont
sass.balint@nytud.hu

Keret

„számítógépes nyelvészet”

– nyelvészetet csinálunk informatikai tudással
– informatikát csinálunk nyelvészeti tudással
– informatikát csinálunk

Keret

„számítógépes nyelvészet”

– nyelvészetet csinálunk informatikai tudással ✓
– informatikát csinálunk nyelvészeti tudással
– informatikát csinálunk

Témák

NoSkE = NoSketchEngine – korpuszkezelő rendszer (← *lényeg !*)
Mtsz = Magyar történeti szövegtár – *elemzetlen*

MNSZ2 = Magyar Nemzeti Szövegtár – *elemzett*
Mazsola – igei bővítményszerkezet vizsgálatára

A korpuszkeresés elvei

Példák az MNSZ2 logból

0.
Korpusz

elemzetlen korpusz

szóalak

jelent
meg
először
,
hogy
Francis
Crick
és

elemzetlen korpusz = **tokenek sora**

elemzett korpusz

szóalak	szó/íj	szófaj	szótő	NE
jelent	w	ige	jelenik	–
meg	w	ik	meg	–
először	w	hsz	először	–
,	p	p	,	–
hogy	w	ksz	hogy	–
Francis	w	fn	Francis	`B-PERS`
Crick	w	fn	Crick	`I-PERS`
és	w	ksz	és	–

elemzett korpusz = **tokenek + annotáció** $\sim$ **táblázat**

elemzett korpusz

szóalak	szó/új	szófaj	szótő	jelleg
<s>				← struktúra
...				
jelent	w	ige	jelenik	–
meg	w	ik	meg	–
először	w	hsz	először	–
,	p	p	,	–
hogy	w	ksz	hogy	–
Francis	w	fn	Francis	B-PERS
Crick	w	fn	Crick	I-PERS
és	w	ksz	és	–
</s>				

elemzett korpusz = **tokenek + annotáció** $\sim$ **táblázat**

elemzett korpusz

szóalak	szó/új	szófaj	szótő	jelleg	
<s id="5">					← **struktúra-annotáció,** *metaadat*
...					
jelent	w	ige	jelenik	–	← **szó-annotáció**
meg	w	ik	meg	–	
először	w	hsz	először	–	
,	p	p	,	–	
hogy	w	ksz	hogy	–	
Francis	w	fn	Francis	B-PERS	
Crick	w	fn	Crick	I-PERS	
és	w	ksz	és	–	
</s>					

elemzett korpusz = **tokenek + annotáció** $\sim$ **táblázat**

1.
NoSkE + példa: Mtsz

NoSkE felület

Mtsz, egyszerű keresés: *de viszont* **(1. példa)**

Ami látszik:
- nagybetű/kisbetű nem számít – sőt: f
- strukturális információk (oldal, bekezdés, (vers)sor): zölddel
- találatok időrendben

Ami nem látszik:
- évszám katt = részletes bibliográfiai adatok
- találat katt = nagyobb kontextus

NoSkE funkciók

– alkorpuszok – *minden metaadatból automatikusan!* (Baróti, 1808)

– mentés – *összes találat!* (sorok max. száma)

– megjelenítés – struktúrák – `<oldal>`, . . . `<g>`; infó – szó sorszáma (Ctrl!)

– rendezés – *jobb* (vesszők)

– véletlen minta

– **szűrés** – *1..1* (vessző)

– **gyaklisták** – *szóalakok, évszámok,* *1R*

– kollokációk ($\rightarrow$ *se, sem, ne, nem, nincs, nélkül*)

– **CQL = Corpus Query Language – formális lekérdezőnyelv**

$\rightarrow$ használatával tárhatjuk fel a korpuszban rejlő teljes információt!

elemzett korpusznál is hasznos, de *elemzetlennél nagyon kell!*

az így megfogalmazott kérdésre alkalmazható az összes fenti funkció

Pozíciók szűréshez és gyaklistához

keresett kifejezés: *viszont*

	Ám	de	viſzont	hallá	,	hogy	majd	a	’	Trójai	vérből
szűrés ablak	-2	-1	0	1	2	3	4	5	6	7	8
gyaklista poz	2L	1L	[Node]	1R	2R	3R	4R	5R	6R	7R	8R

szűrés ablak (lehet több token):

-1..1 = de viſzont hallá

1..3 = hallá , hogy

1..1 = hallá

gyaklista pozíció (itt csak 1 token!):

1L = Ám

1R = hallá

CQL – reguláris kifejezések (regkif)

Bizonyos tulajdonságú karaktersorozatok megadására.

Speciális jelentésű karakterek:

`.`	tetszőleges karakter
`*`	a megelőző karakterből 0 vagy több
`+`	a megelőző karakterből 1 vagy több
`?`	a megelőző karakterből 0 vagy 1
`[ab]`	'a' vagy 'b' karakter
`[^ab]`	nem 'a' és nem is 'b' karakter
`r|s`	'r' vagy 's' reguláris kifejezés
`(..)`	egybefoglalás
`\`	a követő karakter „escape"-elése

(1) `alma`	(4) **`nélk[üűúű]l`**	(7) `alma|almá.*`
(2) `tejf.l`	(5) `.*`	(8) `\.`
(3) `mondjá(to)?k`	(6) `.*bb`	(9) **`([Aa]|[Aa]z|Ee]gy)`**

CQL (Corpus Query Language)

`[..]`	egy tokenre vonatkozó megkötések
`[..]`*op*	egy tokenre vonatkozó operátorok: *op* = *? + {n,m}
`x="y"`	*x* attrib értéke legyen *y* – Mtsz: csak 1 attrib van, a *word*
`x!="y"`	*x* attrib értéke *ne* legyen *y*
`&`	*és* kapcsolat megkötések között
`<s>`	strukturális elem: mondat eleje

(1) `[] []`

(2) `[word="majd"]`

(3) `"majd"`

(4) `[word!="a.*"]`

(5) `[]{0,5}`

(6) `<s> [word="[Nn]em"] [word="kellett"] [word="volna]? [word=".*ni"]`

Regkif 2 szinten: attribútumértéken belül + tokenek szintjén

CQL (Corpus Query Language)

```
[word="volna"]?
```

$\leftrightarrow$

```
[word="volna?"]
```

2. példa: tárgy + ige

Feladat. Keressünk ilyet: *tárgyesetű szó + múltidejű E/3 ige!*

2. példa: tárgy + ige

Feladat. Keressünk ilyet: *tárgyesetű szó + múltidejű E/3 ige!*

```
".+t"  ".+tt"
```

2. példa: tárgy + ige

Feladat. Keressünk ilyet: *tárgyesetű szó + múltidejű E/3 ige!*

```
".+t" ".+tt"
```

most itt –???

```
".+t" [word=".+tt" & word!="(itt|alatt)"]
```

2. példa: tárgy + ige

1. CQL: `".+t" ".+tt"`
2. Gyakoriságok / szóalakok
3. $p \rightarrow$ erőt vett
4. Milyen szó jön utána? $\rightarrow$ Gyakoriságok: 1R
5. $p \rightarrow$ rajta
6. Rendezés / jobb $\rightarrow$ hogy *mi* vesz erőt rajta

$\rightarrow$ félelem, féltékenység, habozás, kacagás, kishitűség, kíváncsiság ...

3. példa: alanyesetű melléknév

Nincs fogodzó …

3. példa: alanyesetű melléknév

Nincs fogodzó . . . *csak a kontextusban!*

-bAn = leggyakoribb esetrag: `".+b[ae]n"` → főnevek
(esetleg: -rA, -vAl ↔ nem jó: -t, -nAk)

1L gyaklista → nem valami jó . . .

3. példa: alanyesetű melléknév

Nincs fogodzó . . . *csak a kontextusban!*

-bAn = leggyakoribb esetrag: `".+b[ae]n"` → főnevek
(esetleg: -rA, -vAl ↔ nem jó: -t, -nAk)

1L gyaklista → nem valami jó . . .

szűrés: -2..-2 `"([Aa]z?|[Ee]gy)"`

1L gyaklista → egész jó
(1-2 birtokos: ember, világ, nm-k . . . kizárni hogy lehetne?)

- szomszéd – nem főnév, melléknév!
- mult – helyesírási hibás!

2.
MNSZ2, Mazsola

MNSZ2

A „mai magyar írott köznyelv reprezentatív korpusza” kíván lenni.

1,04 milliárd szövegszó (= Mtsz $\times$ 35) – v2.0.5

méretéből adódóan sok esetben lassú (gateway timeout! `"m.*"`)
ami gyors: szóalak, szótő, CQL $\leftrightarrow$ egyszerű keresést ne!
kisbetű/nagybetű eltér:
`[word="nem"]` $\leftrightarrow$ `[word="[Nn]em"]` $\leftrightarrow$ `[word="(?i)nem"]`

metaadatok kevésbé kidolgozottak

viszont: **elemzett!** = plusz attribútumok

(vö: Mtsz megjelenítés $\leftrightarrow$ MNSZ2 megjelenítés)

MNSZ2 – attribútumok

(1)	word	**szépet**
(2)	lemma	szép
(3)	msd	MN.ACC
(4)	ana	compound=n;;hyphenated=n;;stem=szép::MN;; morphemes=et::ACC;;mboundary=szép+et
(5)	word_cv	CNCNC
(6)	word_syll	2
(7)	lemma_cv	CNC
(8)	lemma_syll	1
(9)	word_phon	Sépet (←*!!!*)
(10)	lemma_phon	Sép

Mind ugyanúgy használható, mint az Mtsz-ben a *word*!

példa: `[lemma="szép"]` – *példa:* `[lemma_cv="CBCCNC"]`

(az attribútumoknak megfelelően vannak újabb gyaklista-típusok is, ana...)

MNSZ2 – igekötős attribútumok

form	lemma	xpostag	compound	prev	previd	prevpos
kapaszkodik	kapaszkodik	[/V][...]	kapaszkodik			
→						
kapaszkodik	kapaszkodik	[/V][...]	kapaszkodik			
eloldozódott	eloldozódik	[/V][...]	el#oldozódik			
→						
eloldozódott	eloldozódik	**[/Prev]**[/V][...]	el#oldozódik	**pfx**		
tér	tér	[/V][...]	tér			
vissza	vissza	[/Prev]	vissza			
→						
tér	**visszatér**	**[/Prev]**[/V][...]	**vissza#tér**	**sep**	**7**	**+1**
vissza	∅	[/Prev]	∅	**conn**	**7**	
haza	haza	[/Prev]	haza			
akarok	akar	[/V][...]	akar			
menni	megy	[/V][Inf]	megy			
→						
haza	∅	[/Prev]	∅	**conn**	**26**	
akarok	akar	[/V][...]	akar			
menni	**hazamegy**	**[/Prev]**[/V][Inf]	**haza#megy**	**sep**	**26**	**-2**

igekötős példák

1. igekötős ige összes találata:
   CQL: `[lemma="előjön"]`

2. elvált alakjai:
   CQL: `[lemma="előjön" & prev="sep"]`

3. összes igekötős ige:
   CQL: `[xpostag="\[/Prev\]\[/V\].*"]`

MNSZ2 – részletes keresés

plusz szolgáltatás

kattingatással állítjuk össze a kívánt lekérdezést
$\rightarrow$ a háttérben persze CQL lesz belőle

Az elemzésnek köszönhetően:

morfológia:
– körülültük, felszedegettük, elsimítottuk, végigcsináltuk, ...

fonológia:
– cél, csal, csaj, csel, dzsal, ...

Részletes kereséssel is lehet szűrni!

kiss ottó: lepereg

messzire távoli távoli senkije
távoli semmibe csillaga rózsafa
kellene hallani zongora belseje
gyermeki nagymama tartani kellene

messzire mondani mennyire bökdösi
kezdeni kellene mennyire belseje
mennyire mondani mennyire holmira
messzire hordani hajnali városi

csuklani dallama dallama nápolyi
tartani nénire lakhelye semmire
mennyire hajdani hajdani démoni
fölveszi mesteri gyűlöli majdani

„Automatikus” versírás

részletes keresés / fonetikai tulajdonságok használata

szóalak =
`{con}(({lng}|{sht}{con}){con}{sht}{con}{sht}`

Mazsola

igék bővítményszerkezetének vizsgálatára

reprezentáció:
A lány vállat vont. → `ige=`von `alany=`lány `tárgy=`váll

felület . . .

példák:

- *eszik -t*
- *hagy -t*
- *hideg hátán* – „kifordított” keresés: igére
- *erőt vesz rajta vmi* – csináljuk meg jobban!:)

3.
A korpuszkeresés elvei

A korpuszkeresés elvei #1

1. Nyelvi példákat korpuszból!
   Korpusz = élő, valódi nyelvhasználat.

2. Minden találat kell!

3. Ne bízzunk vakon az annotációban!

4. „Alap” korpuszkészlet.

5. Nézzük meg korpuszban!

Nyelvi példákat korpuszból 1.

konstruált példa $\leftrightarrow$ *élő példa:*

két ló húzza a szekeret
mint a hogy húzza a vetőgépet a ló, és a jármot az ökör

a Győr-Moson-Sopron megyeiek tettek bele rendkívül sok pénzt
olcsó az alma, rendkívül sok termett

Nyelvi példákat korpuszból 1.

konstruált példa $\leftrightarrow$ *élő példa:*

két ló húzza a szekeret (ÉKSz)
mint a hogy húzza a vetőgépet a ló, és a jármot az ökör (Mtsz)

a Győr-Moson-Sopron megyeiek tettek bele rendkívül sok pénzt (MNSZ2)
olcsó az alma, rendkívül sok termett (0...)

„Minden találat kell!” elv 2.

a korpuszlekérdezők célja: hogy a felhasználó az *összes* találatot megkapja arra a kérdésre, amire a felület használata közben *gondolt*.:)

másképp: magas fedés kell! $\leftrightarrow$ alacsony pontosság nem annyira gond

- *tejföl* (3245) $\rightarrow$ visszaadjuk a *tejfel*-t is (474 = 12%)?
- *hogy* esetén: *hoyg* (1393)?
- ómagyar: *majd* $\rightarrow$ *maÿd* biztosan kell. Kérdés: *majdan*?
- *bokor* $\rightarrow$ *bokrok*?

Mit szeretne a felhasználó?

Legyen külön kapcsoló minden jelenségre?
e/ö, helyesírási hibák, régies alak, ragozott alak ...
Nagyon sok kapcsoló lenne.

„Minden találat kell!” elv 2.

Megoldás lenne elvben: **normalizálás**

$\sim$ vö: kitalálni, amit a felhasználó látni szeretne.

A normalizálás arra szolgál, hogy a lekérdezésre vetítse
az összes olyan korpusz-tokent, ami rá illik/illeszthető.

Hogy találjuk ki mit szeretne a felhasználó?

ötlet: „nyelvészetileg” releváns-e az adott különbség vagy nem?
$\rightarrow$ Ha nem, akkor normalizáljuk = azonos alakra hozzuk!
De el lehet-e ezt dönteni?

Az *eredeti* felszíni alak biztosan meghagyandó.

A nem tökéletes annotáció elve 3.

Annotáció és fedés

gond: ha hibás az annotáció → csökken a fedés (pl.: *szomszéd*)

Ne bízzunk vakon a korpusz annotációjában, tartalmazhat hibákat.
Tudatosítsuk, hogy konkrétan mennyire bízhatunk benne.

El kell gondolkodni azon, hogy adott kérdésre az annotáció választ tud-e adni.
Ha embernek is nehéz eldöntenie, akkor a géptől se nagyon várjuk.
Adott esetben akár hagyjuk figyelmen kívül az annotációt!

pl.: elkészített – melléknévi igenév *vs.* múlt idejű ige, vagy: *terem*

> **Ne várjuk, hogy a korpusz annotációja tökéletes lesz.**
> **Ne várjuk, hogy pont az aktuális kutatási kérdésünket fogja automatikusan megválaszolni.**

Használjuk a meglévő annotációt kreatívan!

Nemzeti Korpuszportál (NKP)

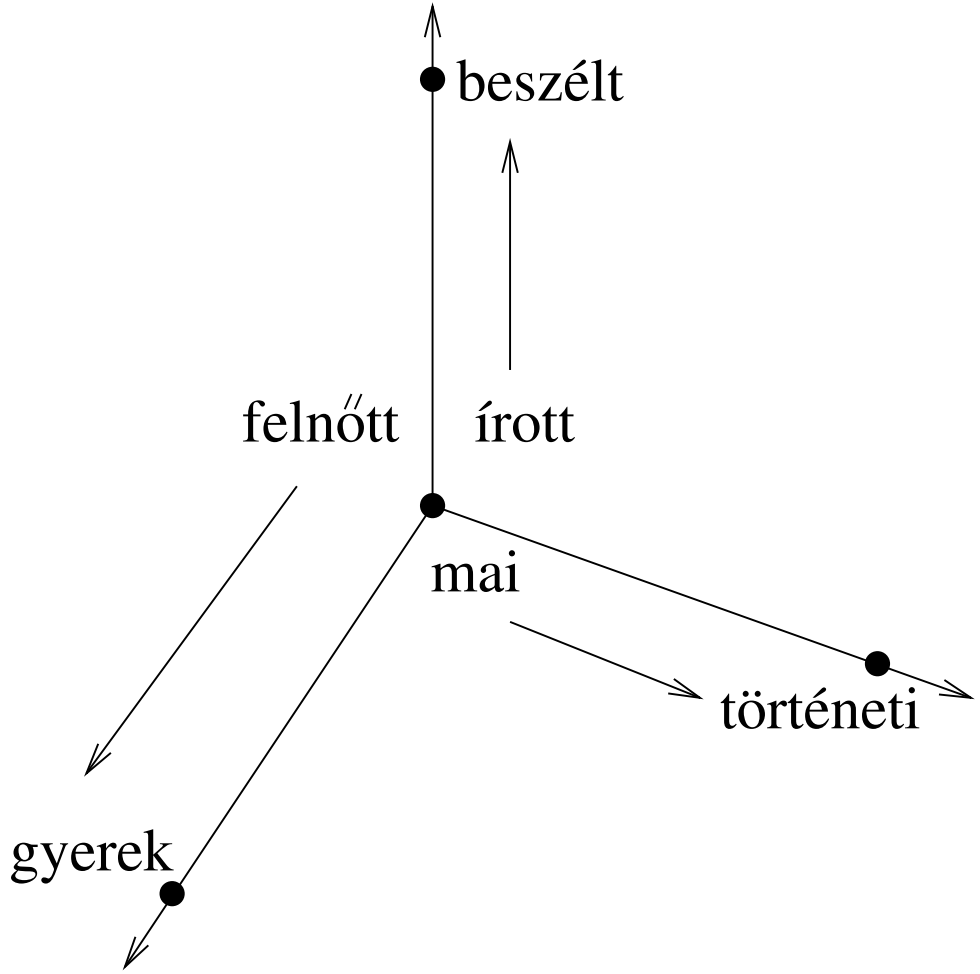
Együtt, egy helyen minél több meglévő…
- magyar nyelvű, online lekérdezhető korpusz
- korpuszlekérdező funkció

`http://corpus.nytud.hu/nkp`

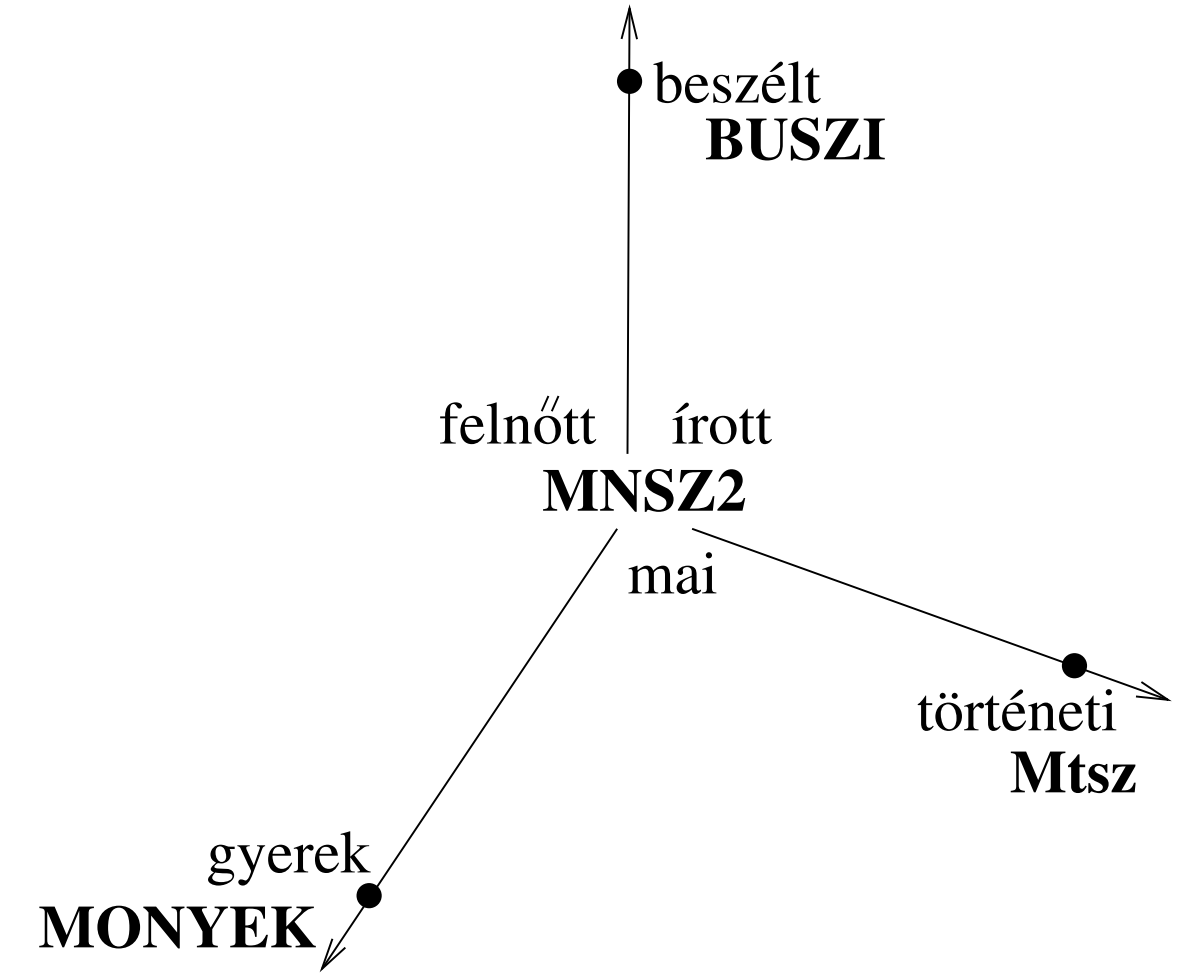
Cél: a korpuszok népszerűsítése a szakma és a nagyközönség felé

Távlati cél: egységesítés, automatizálás

„Alap” korpuszkészlet 4.



„Alap” korpuszkészlet 4.



Nézzük meg korpuszban! 5.

A korpuszok a nyelvi adatok forrásaként arra szolgálnak,
hogy segítségükkel nyelvészeti kérdésfelvetéseket, hipotéziseket
alátámasztani vagy cáfolni lehessen.

Ha szembetalálkozunk egy nyelvészeti állítással, akkor
ha rendelkezésre áll a megfelelő korpusz,
azonnal ellenőrizhetjük az állítás igazságtartalmát, megfelelőségét.

Kialakítható egy olyan hozzáállás, gondolkodásmód,
hogy amikor felmerül egy ilyen állítás vagy kérdés, akkor
készségszinten, természetes módon nyúljunk a korpuszhoz,
és ott keressünk választ.

Korpuszok együttműködése: cigány eredetű szavak (1/2)

szavak:

csaj, csávó, csór, gádzsó, gizda,
góré, kaja, kéró, lóvé, nyikhaj,
pia, pimasz, séró, verda

$\rightarrow$ Melyik a kakukktojás?

Korpuszok együttműködése: cigány eredetű szavak (1/2)

szavak:

csaj, csávó, csór, gádzsó, gizda,
góré, kaja, kéró, lóvé, nyikhaj,
pia, pimasz, séró, verda

$\rightarrow$ Melyik a kakukktojás?

első előfordulás az **Mtsz**-*ben:*

csaj – 1963, csávó – 1971, csór – 1913, gádzsó – $\emptyset$, gizda – $\emptyset$,
góré – 1965, kaja – 1948, kéró – $\emptyset$, lóvé – 1968, nyikhaj – 1978,
pia – 1954, pimasz – 1785, séró – 2003, verda – 2004

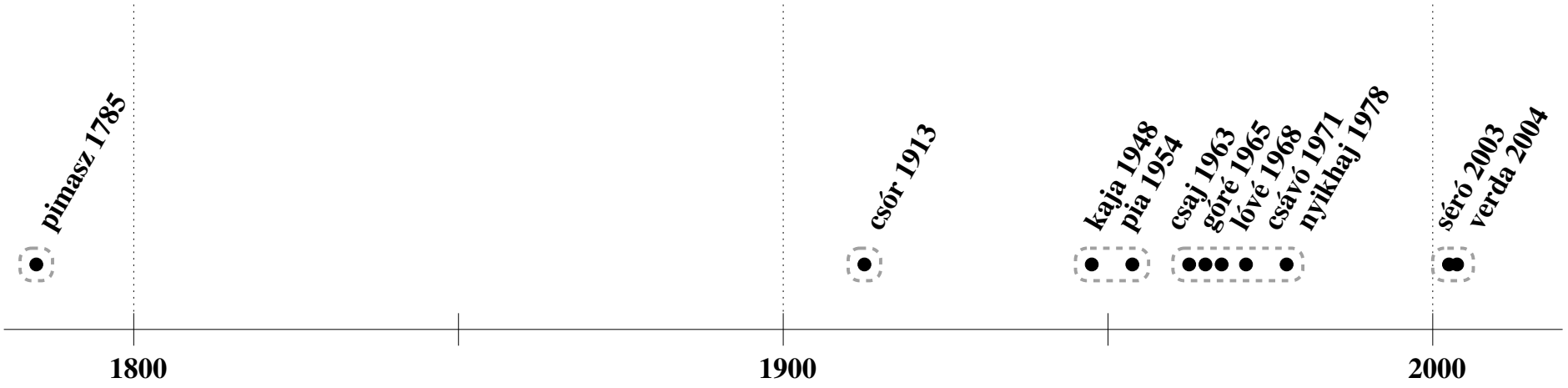
Korpuszok együttműködése: cigány eredetű szavak (1/2)

szavak:

csaj, csávó, csór, gádzsó, gizda,
góré, kaja, kéró, lóvé, nyikhaj,
pia, pimasz, séró, verda

→ Melyik a kakukktojás?

első előfordulás az **Mtsz**-*ben:*



→ a *pimasz* régi magyar szó!:)

Korpuszok együttműködése: cigány eredetű szavak (2/2)

Mennyire köznyelvi?

ötlet: gyakoriság közeli szinonimával összevetve: **MNSZ2**

lány	198000	csaj	10000	×**20**
szemtelen	1976	pimasz	1825	**=**

→ *pimasz* teljesen köznyelvi
→ *csaj* kevésbé köznyelvi, van stílusértéke!

A korpuszkeresés elvei #2

6. A szűrés elve.
7. A fókusz elve.
8. A kontextusalapú keresés elve.

A szűrés elve 6.

Lekérdezés finomítása.

Törekedjünk rá, hogy minél kevesebb találatot kapjunk. :)

Ha `DET + MN.NOM`-ra keresünk,
akkor először `MN.NOM`,
aztán szűrés: `DET`.

A fókusz elve 7.

mire vagyok kíváncsi = miből szeretnék gyakorisági listát készíteni

4. példa volt: Készítsünk gyakorisági listát
a *fog*-tól jobbra 1, 2 vagy 3 szó távolságban lévő FNI-kből.

megoldás volt: 1. FNI + 2. *fog*

A fókusz elve 7.

Többszavas lekérdezés vagy szűrés?

Ha többszavasra keresünk:
annak a részeiből nem tudunk gyaklistát készíteni (*Node*).
De az egészből és a hozzá képest vett n-edik szóból igen.

Ha egy szóra keresünk + szűrés:
csak az első szóhoz képest n-edik szóból tudunk gyaklistát készíteni.
Az itt-ott megjelenő „szűrésből kijött” szavakból nem.

Mindig végig kell gondolni: éppen melyik megközelítés a hasznos.

Lehetőség: többszavast így felépíteni: egy szó + 1..1, 2..2 szűrés (!)
$\rightarrow$ és akkor lehet gyaklistát csinálni a részeiből.

A kontextusalapú keresés elve 8.

alap: a keresett szó (formai/elemzési) tulajdonságai alapján keresünk

ha ez nem megy: megpróbálhatunk a *kontextus* (formai/elemzési) tulajdonságai alapján keresni!

- *3. példa volt:* alanyesetű melléknév?
  ezt alkalmaztuk: főnév *előtti pozícióban* fogjuk megtalálni
- keressünk ilyet: *villany lekapcsol, ajtó becsuk, ...*
  tipp: ez felsorolásként jelenik meg
  („halmozódás elve”)
- keressünk pozitív értékelőket
  tipp: (*jó* utáni szavak – *rossz* utáni szavak) előtt mi van!
  => különleges, szülőkímélő, kihagyhatatlan
  („áttételes keresés elve”)

4.
Példák az MNSZ2 logból

Példák az MNSZ2 logból

1. érdekes/értelmes lekérdezések
2. hibák
3. „ilyet ne”

a) `"tudatjuk" "mindazokkal"`

b) `[lemma="felkap"] [lemma="a"] [lemma="víz"]`

c) `[word="."]`

d) `[lemma = "k[K]andeláber"]`

e) `ama i*`

Példák az MNSZ2 logból

1. érdekes/értelmes lekérdezések
2. hibák
3. „ilyet ne”

f) `[word="\."] [word="[Mm]indig"] [word="\."]`

g)
```
[msd="Det.*"] [msd="FN.PSe2.*"]
[lemma="fog"] [word=".*ni" & pos="V.*"]
```

h) `".*"`

i) `[][][][]*`

j) `[word = "elé"] [word = "a.?"] [word = ".+n(a|e)k"]`

Összefoglalás

- az elemzett korpusz = egy **táblázat**
- NoSkE: **szűrés 1..1** ***és*** **gyakorisági lista 1R**
- NoSkE: **regkif + CQL = `".+t" ".+tt"`**
- Mtsz: elemzetlen ↔ MNSZ2: elemzett!
- **elvek:** „Példák korpuszból” / „Minden találat kell!”
  „Ne bízzunk vakon az annotációban!”
  „Nézzük meg korpuszban!” / „Kontextusalapú keresés”
- MNSZ2 példák: többszavas lekérdezés ↔ szűrés

olvasnivaló: Sass Bálint: Principles of Corpus Querying.
In: *Acta Linguistica Academica.* **69**(4): 599-614.

Sass Bálint – `sass.balint@nytud.hu`