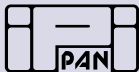


Poliqarp2 na ostatniej prostej

Aleksander Zabłocki, Bartosz Zaborowski



INSTITUTE OF COMPUTER SCIENCE
POLISH ACADEMY OF SCIENCES
ul. Jana Kazimierza 5, 01-248 Warszawa

21 marca 2016

Plan



- 1 Wstęp
 - Założenia projektu
- 2 Przypomnienie modelu danych
- 3 Język zapytań w działaniu
- 4 Implementacja
- 5 Na zakończenie

Założenia dla Poliqarpa 2 – funkcjonalności



- Następca Poliqarp 1.x – cała funkcjonalność zachowana, szczególnie niejednoznaczność i wyrażenia regularne na segmentach,
- obsługa wielu poziomów anotacji (anotacje z NKJP i nie tylko),
- wsparcie dla innych korpusów/banków (korpusy historyczne, grafy składniowe/zależnościowe, struktury LFG...).

Założenia dla Poliqarpa 2 – techniczne



- Obsługa dużych zbiorów tekstów (miliardy segmentów),
- rozproszenie i zrównoleglenie obliczeń,
- stabilność działania,
- graficzny interfejs z wizualizacją wyników.

Plan



- 1 Wstęp
- 2 Przypomnienie modelu danych**
- 3 Język zapytań w działaniu
- 4 Implementacja
- 5 Na zakończenie

Typy danych



- typy proste (int, float, bool, string, enum)
- struktury atrybutowe (np. węzeł, krawędź, interpretacja, F-struktura)
- lista
- lista wieloznaczna (np. wszystkie interpretacje segmentu)

Rodzaje struktur



- wyróżnione rodzaje struktur: węzły wielkoskalowe, węzły zwykłe i krawędzie
- 2 rodzaje krawędzi: pierwotne i wtórne
- węzły i krawędzie pierwotne tworzą DAG (składnia)
- krawędzie wtórne bez ograniczeń (dowolny graf)
- wyróżniony atrybut `type` – nadaje sens poszczególnym strukturom
 - zagnieżdżanie (konwencja), np. `syngtr:PrepNG`,
`named:person:surname`
 - wyróżnione wartości: `seg`, `doc`, `sub`

Rodzaje struktur



- wyróżnione rodzaje struktur: węzły wielkoskalowe, węzły zwykłe i krawędzie
- 2 rodzaje krawędzi: pierwotne i wtórne
- węzły i krawędzie pierwotne tworzą DAG (składnia)
- krawędzie wtórne bez ograniczeń (dowolny graf)
- wyróżniony atrybut **type** – nadaje sens poszczególnym strukturom
 - zagnieżdżanie (konwencja), np. `syngtr:PrepNG`,
`named:person:surname`
 - wyróżnione wartości: `seg`, `doc`, `sub`

Dodatkowe założenia



- węzły wielkoskalowe i krawędzie zwykłe między nimi tworzą drzewo (segmentacja wielkoskalowa)
- rozróżniamy porządek krawędziowy (dzieci węzła) i segmentowy (kolejne segmenty, niekoniecznie liniowy)
- dla LFG i podobnych: warianty DAGu (składni)
 - każda struktura (węzeł, krawędź) wie w których wariantach występuje,
 - część wariantów wyróżniona (np. poprawny rozbiór),
 - warianty globalne w skali dokumentu

Dodatkowe założenia



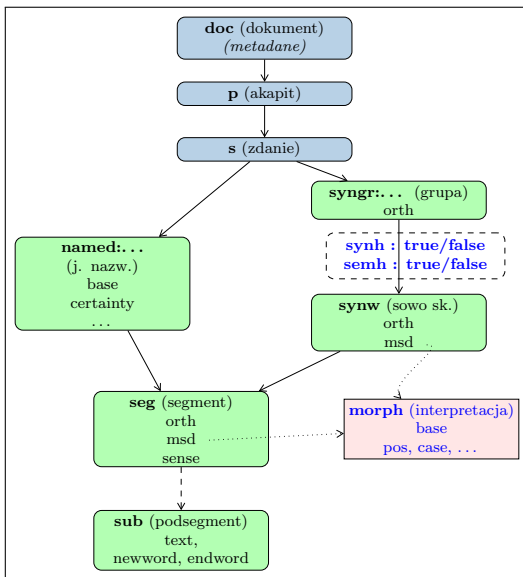
- węzły wielkoskalowe i krawędzie zwykłe między nimi tworzą drzewo (segmentacja wielkoskalowa)
- rozróżniamy porządek krawędziowy (dzieci węzła) i segmentowy (kolejne segmenty, niekoniecznie liniowy)
- dla LFG i podobnych: warianty DAGu (składni)
 - każda struktura (węzeł, krawędź) wie w których wariantach występuje,
 - część wariantów wyróżniona (np. poprawny rozbiór),
 - warianty globalne w skali dokumentu

Przykład modelu NKJP



- DAG: segmentacja wielkoskalowa + (grupy + słowa składniowe) + morfoskładnia:
 $\text{doc} \rightarrow \text{p} \rightarrow \text{s} \rightarrow \text{syng}r:\dots \rightarrow \text{syn}w \rightarrow \text{seg} (\rightarrow \text{sub})$
- metadane w węźle-korzeniu (**doc**), jako zwykłe atrybuty
- krawędzie z grup do słów skł. z informacją o głowach (atrybuty **synh** i **semh**)
- lista wieloznaczna interpretacji jako atrybut **msd** w węzłach segmentów
 - interpretacje – struktury atrybutowe, klasa i kategorie gram. jako zwykłe atrybuty
 - jedna wyróżniona interpretacja na liście wieloznacznej

Model NKJP na obrazku



Plan



- 1 Wstęp
- 2 Przypomnienie modelu danych
- 3 Język zapytań w działaniu**
- 4 Implementacja
- 5 Na zakończenie

Złożone typy danych (1)



[pos~adj && case~gen && {pos,case}!~{adj,gen}]

Być może przymiotnik i być może w dopełniaczu, ale nie naraz

rzeciw? - Na zakończeniu, uwzględniając atmosferę,	jaka [:jaki:f:sg:adj]	w ostatnich dniach wytworzyła się wokół rolnictwa, w tym
wodawczej o rządowym projekcie ustawy o ratyfikacji	umowy [:umowa:f:sg:subst]	między państwami - stronami Traktatu Północnoatlantycki
i proponuje, aby zwiększyć dochody z prywatyzacji w	części [:część:f:sg:subst]	8 - Ministerstwo Finansów o kwotę 4 bln zł i
tak nagłaśnianych, jak masowa repatriacja, na którą	Polski [:Polska:f:sg:subst]	w obecnej sytuacji po prostu nie stać. Stać nas natomiast
na aktywizację absolwentów przewidują również inne	części [:część:f:pl:subst]	Funduszu Pracy, na przykład programy specjalne. Jedna
, to nie wiem, w którym momencie mógłby	on [:on:m1:sg:ppron3]	przyjść. Proszę pana posła Władysława Bulkę o zabranie
towych. - Pan poseł Leszek Moczulski, Konfederacja	Polski [:Polska:f:sg:subst]	Niepodległej... Czy jest pan poseł? Przykro mi, nie ma pa
osób z upośledzeniem umysłowym w stopniu lekkim i	jaki [:jaki:m3:sg:adj]	stanowią one procent poborowych wcielonych do wojska?
poważnym błędem budowanie go tak, jak gdyby ten	jeden [:jeden:m3:sg:adj]	rok decydował o wszystkim. Próba realizacji wszystkich dl

Złożone typy danych (2)



[orth.size >= 4 && msd.all.pos.uniq.size >= 3]

≥ 4 znaki, które można rozumieć jako ≥ 3 różne części mowy

wdrożonył, to minister Kołodko poda się do dymisjił,	była [imperf:być:f:sg:praet]	podana publicznie w zeszłym tygodniul, osobiściel.
ieść rękę i nacisnąć przyciskil. - Przepis obowiązujący	stanowi [imperf:stanować:sg:fin]	, że wnioski w tej sprawie zainteresowani mogli sk
jusza Lewicy Demokratycznejł. Panie pośle - 5 minutł,	tyle [:tyle:f:pl:num]	wynosi czas klubowył.
i zapotrzebowania widownił, chyba że się bardzo nie	lubi [imperf:lubić::sg:fin]	rolnikówł. Dziękujel. (Oklaskił) - Dziękujel, panie po:
oncesje w zakresie świadczenia usług ochrony osób i	mienia [:mienie:n:sg:subst]	. - Chcę wyjaśnićł, że nie zaproszono przewodnic
y w zagospodarowaniu przestrzennym uwzględnione	były [imperf:być:n:pl:praet]	również wymagania osób niepełnosprawnychł. - Pi
ządu Terytorialnegol. - Z powyższych argumentów w	sposób [:sposób:m3:sg:subst]	logiczny wpływa wniosekł, który składam w imieni
, że czas pokażeł, że dokonane przez nas wybory	były [imperf:być:m3:pl:praet]	właściwel. A co do powtarzających się zarzutów kc
datków w części 18ł: Urząd Mieszkalnictwa i Rozwoju	Miast	przeznaczonych na refundacje premii gwarancyjnł.

Porządek segmentowy i krawędziowy



$\$X := [\text{sons} = \{ | \$Y \$Z | \}] \&\& !(\$Y \$Z) / rha$

$\$Y, \Z są kolejne w porządku krawędziowym (jako dzieci $\$X$), ale nie w segmentowym

mił, analogicznie do prośby Unii Wolności, aby	ten [:ten:m3:sg:adj] apel [:apel:m3:sg:subst]	mógł odczytać w czasie pozaklubowym. - Stwierd
wniosek o jego skreślenie. Proszę państwa, wydaje	mi [:ja:m1:sg:ppron12]	się, że od tego momentu ureguluje się pewną dzia
a kwalifikacyjne w stosunku do osób, które będą się	chciały [imperf:chcieć:f:pl:praet] nimi [:on:m3:pl:ppron3]	trudnić. - Uważam, że wniosek, który przekażą za
łęboko przekonany, że czas pokazał, że dokonane	przez [:przez:::prep] nas [:my:m1:pl:ppron12]	wybory były właściwe. A co do powtarzających się
ce pytania: 1. Kiedy zaprzestanie się powoływania	do [:do:::prep] wojska [:wojsko:n:sg:subst]	osób z upośledzeniem umysłowym, kto i w jakiej w
- Panie pośle, przepraszam bardzo, ale pan zdaje	mi [:ja:m1:sg:ppron12]	się, nie mówi o ustawie, która dotyczy pakietu
uchwyciłem) korzysta już z tego gmachu? Czy się	tam [:tam:::adv]	wprowadził? Jaką kwotę łącznie wypłacono? - Pa
musiał wypowiedzieć się przy okazji, kiedy będziemy	ten [:ten:m3:sg:adj] punkt [:punkt:m3:sg:subst]	omawiali, kiedy rząd będzie do tego punktu przyg
yższy niż w zarządzanym przez ministerstwo, co by	je [:on:n:sg:ppron3]	skompromitowało. Myślę, że to jest główny, rzec

Kwantyfikatory (1)



#30, X: \$X.pos=subst

zdanie z co najmniej 30 rzeczownikami

- Propozycje zmian ustawy o organizacji i finansowaniu ubezpieczeń społecznych dotyczą zwiększenia możliwości prawnych ZUS w egzekucji świadczeń na Fundusz Ubezpieczeń Społecznych, m.in. poprzez prawo do występowania z wnioskiem o założenie księgi wieczystej dłużnikom będącym niebudżetowymi państwowymi jednostkami organizacyjnymi, przyznawanie prawa do występowania z wnioskiem o wpisanie należności ZUS do hipoteki, zrównanie uprawnień kontrolerów ZUS z uprawnieniami kontroli skarbowej oraz zabezpieczenie takiej samej ochrony prawnej ZUS, jak jednostkom podległym ministrowi finansów.

Kwantyfikatory (2)



```
##X: ($X.pos~subst => $X.orth="p.*"/i)
```

zdanie, w którym wszystkie potencjalne rzeczowniki zaczynają się od litery P :)

- Proszę|, panie pośle|.

- Dziękuję| panu posłowi|.

- Dziękuję|, panie pośle|.

Poprawkę nr 11 już rozpatrzyliśmy|.

Panowie Posłowie|!

Przepraszam|, zgubiłem się|.

- Był|, był|.

Nie tworzymy po prostu przepisów|, które są fikcyjn|.

- Proszę| bardzo|, naturalnie|.

Dziękuję|.

- Panie Prezesie|!

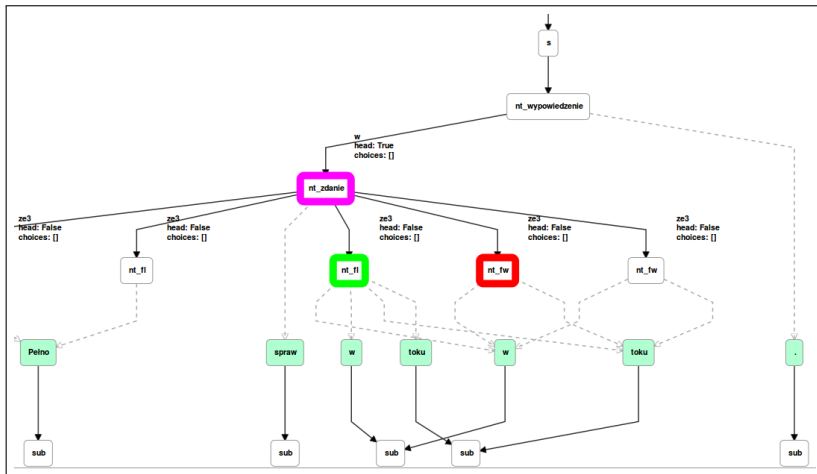
Dziękuję|.

- Nie jest istotn|, czy pan poseł jest za|, czy przeciwi|!

Wyrażenia pionowe: Składnica (1)

$\$A > \{\$B, \$C\} \ \&\& \ \$B > \$D \ \&\& \ \$C > \$D \ \&\& \ \$B.type \neq \$C.type$

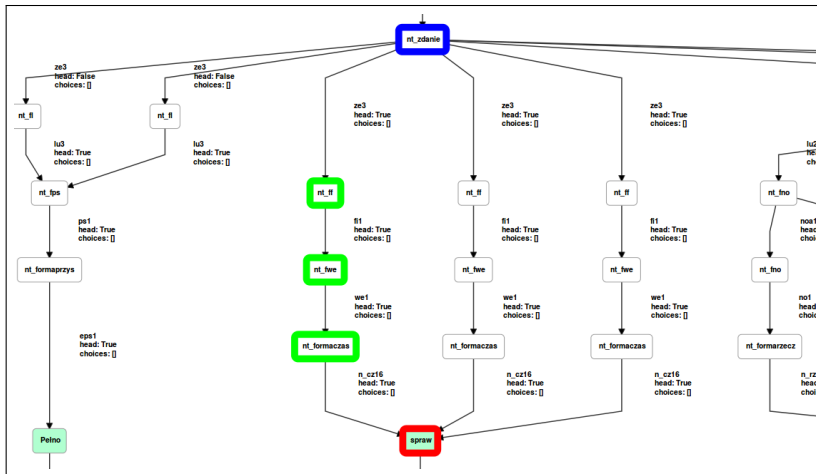
“diament”: $\$A$ dominuje $\$D$ poprzez $\$B$ i $\$C$, które są różnego typu



Wyrażenia pionowe: Składnica (2)

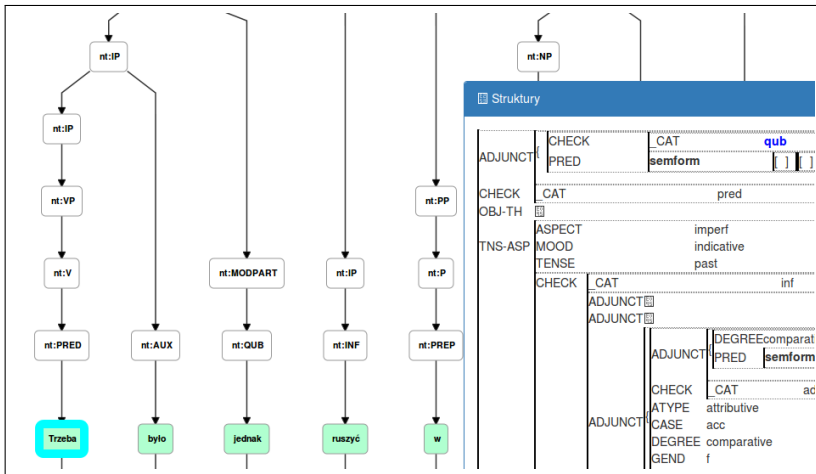
`[nt_zdanie] > [head]{4,}{:,$P+=[]:} $S:=[seg]`

od zdania do segmentu — min. 4 kroki — tylko głównymi dziećmi — spamiętuj w \$P





c-struktura $\$X$ mapuje się na f-strukturę $\$Y$, której a... try... bu... tem jest "qub"

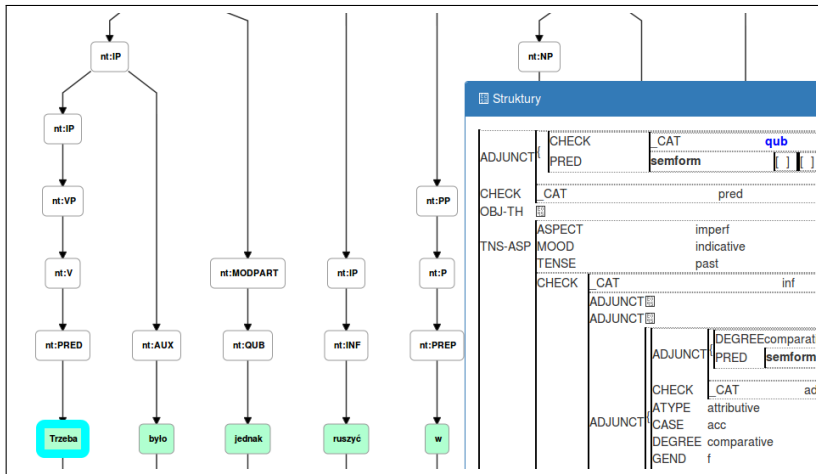


Wyrażenia pionowe: LFG



$\$X \gg \$Y \ \&\& \ \$Y >((\text{ADJUNCT} | \text{_HAS} | \text{CHECK} | \text{_CAT})^+) \text{qub}$

c-struktura $\$X$ mapuje się na f-strukturę $\$Y$, której a... try... bu... tem jest "qub"



Post-processing (1)



```
[seg && orth=$0 && base=$B]/pa ;;
      group $0, $B ;;
zip($0, ", ") group $B sort count() desc
```

najróżniej odmieniane formy bazowe w korpusie

B	#4	#3
być	Być, Był, Było, Jest, Jestem, być, był, była, było, byli, będzie, będziemy, będą, będącym, będę, em, jest, jestem, m, są, śmy,	21
on	go, ich, je, jego, jej, ją, nich, nimi, nią, on, one,	11
mieć	Mam, Mamy, ma, mają, mający, mam, mamy, miał, miała, mieli,	10
poprawka	Poprawka, Poprawkę, poprawce, poprawek, poprawka, poprawkach, poprawkami, poprawki, poprawką, poprawkę,	10
który	która, które, którego, której, który, których, którym, którzy, którą,	9
poseł	Posel, Posłowie, poseł, posła, posłom, posłowi, posłowie, posłów, posle,	9
ten	Ta, ta, te, tego, tej, ten, tych, tym, tę,	9
ustawa	Ustawę, ustaw, ustawa, ustawami, ustawie, ustawy, ustawą, ustawę,	8
móc	Mogło, mogli, mogą, mogą, mogła, może, mógł,	7
izba	Izba, Izbie, Izbo, Izby, Izba, Izbę,	6
jaki	Jaką, jaka, jaki, jakie, jakiej, jakim,	6
komisja	Komisje, Komisji, Komisją, Komisję, komisje, komisji,	6
minister	Ministrów, minister, ministra, ministrowi, ministrowie, ministrze,	6
przyjąć	przyjąć, przyjął, przyjęcie, przyjęciem, przyjętymi, przyjętą,	6

Post-processing (2)



```
[seg && orth=$0 && base=$B]/pa ;;
group $0, $B sort $C:=count() desc ;;
zip($0, ":", $C, ", ") group $B sort count() desc
```

... i jeszcze trochę szczegółów

B	#5
być	jest:46, m:17, em:15, będzie:10, są:9, Jest:6, były:5, będziemy:5, b
on	ich:8, je:4, one:4, go:3, jego:3, jej:2, nimi:2, on:2, ją:1, nich:1, nią:1,
mieć	ma:7, Mam:2, mają:2, Mamy:1, mający:1, mam:1, mamy:1, miał:1,
poprawka	poprawki:14, poprawkę:6, poprawce:3, Poprawka:1, Poprawkę:1, p
który	które:7, która:4, który:3, którego:2, której:2, których:2, którym:1, kt
poseł	poseł:39, posłów:26, posła:19, pośle:13, posłowi:3, Posłowie:2, Pos
ten	tej:19, ten:9, tego:7, tym:7, tych:6, ta:3, te:3, tę:3, Ta:1,

Post-processing (3)



```
([base=$X] [base=$Y])/pa ;;
group $X, $Y where total($X) >= 3 sort scp() desc
```

ranking par form bazowych względem symetrycznego p-stwa warunkowego (SCP)

#4	#5	#6	#7	X	Y	#8
1	3	3	3	31	grudzień	5503
1	9	9	9	nacisnąć	przycisk	5503
1	3	3	3	październik	1992	5503
1	13	13	13	porządek	dzienny	5503
0.8181818127632141	9	11	9	podnieść	ręka	5503
0.75	3	4	3	bezrobocie	wśród	5503
0.75	3	4	3	bezwzględny	większość	5503
0.6666666865348816	2	3	2	Kazimierz	Wilk	5503
0.6666666865348816	2	3	2	nowy	demokracja	5503
0.6666666865348816	2	3	2	przemysł	cukrowniczy	5503
0.6666666865348816	2	3	2	rzecz	brać	5503
0.6399999856948853	16	16	25	zabrać	głos	5503
0.6000000238418579	3	3	5	następny	mówca	5503
1	5	5	5	sojusz	lewica	5503
1	3	3	3	wśród	młodzież	5503
0.8333333134651184	5	5	6	lewica	demokratyczny	5503

```
[children~[edges.size=$L1 && rule=$R1] &&
  children~[edges.size=$L2 && rule=$R2]] && $L1!=$L2
```

Składnica: węzeł z różnicznymi zestawami dzieci; przechwytujemy nazwy reguł

```
[base~$B:=".*ć"]
```

przechwycenie wartości spełniającej daną specyfikację

```
[date] within [title=Sejm/x]
```

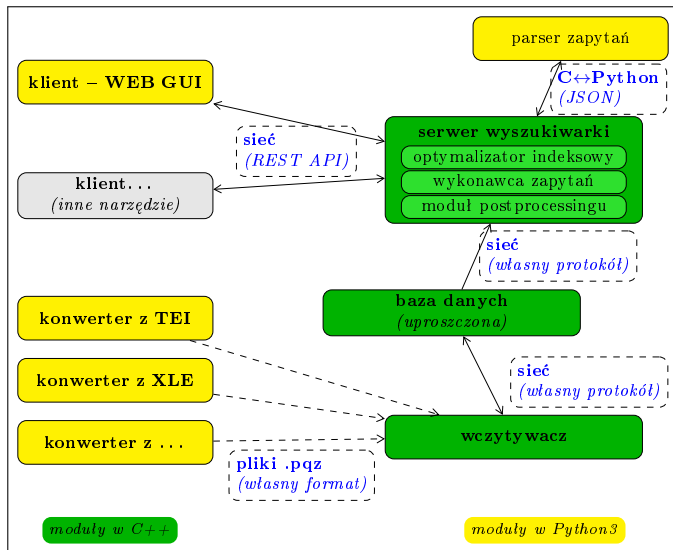
jednostki nazewnicze (named:date); lukier składniowy dla type; metadane

```
[nt_fzd] >{,5} (regularnie [nt_fpm])/rha
```

“nierówne” wyrażenie regularne poziome, w ograniczonym kontekście pionowym

- 1 Wstęp
- 2 Przypomnienie modelu danych
- 3 Język zapytań w działaniu
- 4 Implementacja**
 - Architektura wyszukiwarki
 - O indeksach
 - Wykonawca zapytań
 - NKJP w praktyce
- 5 Na zakończenie

Architektura wyszukiwarki



"Baza danych"



Istniejące uniwersalne bazy danych okazały się niewystarczające dla naszych potrzeb:

- bazy SQL-owe (np. PostgreSQL): za dużo danych, nieefektywne
- bazy tekstowe (np. Solr): brak struktury, problem z danymi binarnymi
- bazy no-SQL (np. MongoDB): sztywne ograniczenia, problemy z efektywnością, semantyka regex-ów

Napisaliśmy własną, uproszczoną bazę:

- brak SQL (bo po co?), w zamian prosty binarny protokół
- zbiory danych jak tabelki SQL
- 3 wzajemnie wykluczające się tryby pracy tabeli:
 - tylko dopisywanie (współbieżnie),
 - tylko odczyt (współbieżny),
 - sortowanie danych (wewnętrzne, niezbędne przed odczytem)
- w pełni kontrolujemy semantykę regex-ów (boost::regex)
- pojedynczy indeks (albo raczej porządek rekordów)
- szybkie wyszukiwanie regex-owe (skan po unikalnych napisach)

- integruje moduły odpowiedzialne za wykonywanie zapytań
- wielowątkowy, wielodostępowy (równoczesne sesje klientów i zapytania)
- komunikacja ze światem przez API REST (JSON, HTTP)
- autoryzacja *HTTP Basic Auth* w ramach API (użytkownicy/role ↔ różne uprawnienia, limity)

Format wyników

Na poziomie API wyniki są przekazywane w postaci zserializowanej do JSON. Dane korpusowe są w formie podobnej do formatu wejściowego, zawierają dodatkowe dane pomocnicze dla GUI.

- integruje moduły odpowiedzialne za wykonywanie zapytań
- wielowątkowy, wielodostępowy (równoczesne sesje klientów i zapytania)
- komunikacja ze światem przez API REST (JSON, HTTP)
- autoryzacja *HTTP Basic Auth* w ramach API
(użytkownicy/role ↔ różne uprawnienia, limity)

Format wyników

Na poziomie API wyniki są przekazywane w postaci zserializowanej do JSON. Dane korpusowe są w formie podobnej do formatu wejściowego, zawierają dodatkowe dane pomocnicze dla GUI.

- Podstawowy interfejs użytkownika wyszukiwarki,
- przenośny, łatwy do rozszerzania i modyfikacji (Python/Django),
- przedstawia wyniki na 3 sposoby:
 - KWIC (podobnie jak w starym Poliarpie),
 - graficzne, dynamiczne struktury (drzewa i f-struktury),
 - zwykły tekst.

Poliqarp2
Zgłaszanie błędów
Zaloguj
Polski / Polish

korpusik

[base=pan] ".*[sc]ki."

Wyszukaj

Znalazłem 3 wyników

1

Kolory przypisane zmiennym: \$ _Q

KWIC

Grafy

Tekst

Pobierz jako plik CSV

dłoń prawą, a raczej miękką okładkę "Poezji"	pana [:pan:m1:sg:subst] Mickiewicza [:Mickiewicz:m1:sg:subst]	. Ów to miękki ucisk przypomniął pannie Anieli pewną altankę z
na biuro komisarskie, chętnie użyczona na ten cel przez	pana [:pan:m1:sg:subst] Czartkowskiego [:Czartkowski:m1:sg:subst]	. Taki był naturalny porządek rzeczy, któremu nikt we wsi się
niepokoju i frasunkul. Gdy już drzwi zamknęły się za ostatnim,	pan [:pan:m1:sg:subst] Słowiński [:Słowiński:m1:sg:subst]	westchnął rozgłośnie, a głęboko, podług miary komisarского trudu

Poliqarp2
Zgłaszanie błędów

Zaloguj
Polaki / Polish

korpusik

[base=pan] *.*[sc]ki.**

Wyszukaj

Znalazłem 3 wyników

1

Kolory przypisane zmiennym: \$_Q

KWIC

Grafy

Tekst

▼ Filtrowanie elementów grafu

Niekiedy bezwiednym a przelotnym gestem przyciskała do piersi dłoń prawą, a raczej miękką okładkę "Poezyj" pana Mickiewicza!

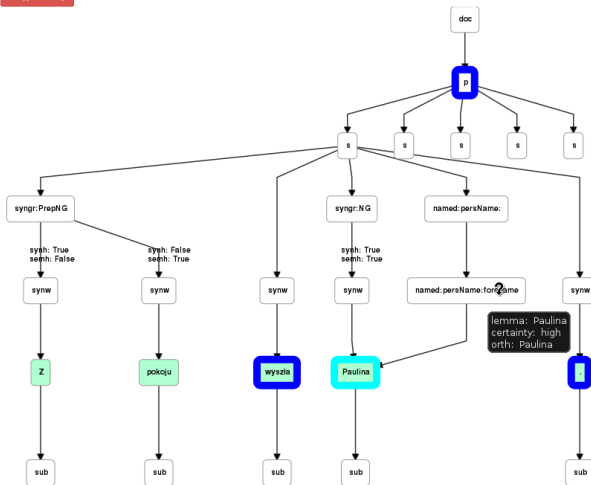
Ukryj struktury

Kancelaria dworska zmieniała się na biuro komisaarskie! chętnie użyczona na ten cel przez pana Czarstkowskiego!

Pokaz struktury

Z pokoju wyszła Paulina. Poprawiając sobie włosy rozejrzała się po korytarzu i majestatycznym krokiem oddaliła się do swojego mieszkania. Romek tkwił jeszcze ukryty za drzwiami - po prostu nie mógł się ruszyć. Przecież wczoraj po południu Paulina była u niego w mieszkaniu. Stała w progu, ręce oparła na biodrach.

Ukryj struktury



Odpowiada za indeksowanie korpusu, tzn.:

- ładowanie plików .pqz do bazy,
- weryfikację poprawności danych, mieszczące się w limitach itp. (→ specjalny tryb testowy),
- tworzenie struktur pomocniczych na poziomie dokumentów,
- buduje indeksy blokowe i meta-indeks.

Problemy z danymi wejściowymi



- Korpusy są dostępne w różnych formatach,
- nawet TEI TEI nie równe,
- obsługa tych formatów z poziomu C++ kusząca ale bardzo kosztowna,
- co jak się pojawi nowy korpus w nowym formacie?

Konwersja na format .pqz



- Z poziomu skryptu pythonowego wczytujemy pliki,
- budujemy z dostępnych klocków strukturę dokumentu, który będziemy przeszukiwać,
- klocki – struktury danych używane w Poliqarpie dostępne w postaci niewielkiej biblioteki,
- zapis plików i wstępne sprawdzenie poprawności też w bibliotece.

Format .pqz

Format *.pqz* to poliqarpowe klocki zserializowane do JSON i spakowane gzip-em.

Konwersja na format .pqz



- Z poziomu skryptu pythonowego wczytujemy pliki,
- budujemy z dostępnych klocków strukturę dokumentu, który będziemy przeszukiwać,
- klocki – struktury danych używane w Poliqarpie dostępne w postaci niewielkiej biblioteki,
- zapis plików i wstępne sprawdzenie poprawności też w bibliotece.

Format .pqz

Format *.pqz* to poliqarpowe klocki zserializowane do JSON i spakowane gzip-em.

Wybór indeksów



- Czym drobniejszy indeks tym **skuteczniejszy**...
- i **cięższy**.
- Indeksy bigramów we wszystkich relacjach pozwalają wyliczyć wiele zapytań bez wykonawcy, ale są za duże.
- Indeksy bloków segmentów (stary Poliqarp):
 - mają sens tylko przy rzadkich wartościach (orth, cały tag),
 - wymagają szybkiego wykonawcy,→ dla nas za słabe.
- Dokładność pojedynczego węzła spisuje się najlepiej.
- Rozszerzenie: oprócz zwykłych pozycji także pozycje sąsiadów w relacjach.

Wybór indeksów



- Czym drobniejszy indeks tym **skuteczniejszy**...
- i **cięższy**.
- Indeksy bigramów we wszystkich relacjach pozwalają wyliczyć wiele zapytań bez wykonawcy, ale są za duże.
- Indeksy bloków segmentów (stary Poliqarp):
 - mają sens tylko przy rzadkich wartościach (orth, cały tag),
 - wymagają szybkiego wykonawcy,→ dla nas za słabe.
- Dokładność pojedynczego węzła spisuje się najlepiej.
- Rozszerzenie: oprócz zwykłych pozycji także pozycje sąsiadów w relacjach.

Wybór indeksów



- Czym drobniejszy indeks tym **skuteczniejszy**...
- i **cięższy**.
- Indeksy bigramów we wszystkich relacjach pozwalają wyliczyć wiele zapytań bez wykonawcy, ale są za duże.
- Indeksy bloków segmentów (stary Poliqarp):
 - mają sens tylko przy rzadkich wartościach (orth, cały tag),
 - wymagają szybkiego wykonawcy,→ dla nas za słabe.
- Dokładność pojedynczego węzła spisuje się najlepiej.
- Rozszerzenie: oprócz zwykłych pozycji także pozycje sąsiadów w relacjach.

- Indeksy dla każdego atrybutu na poziomie węzłów (uwaga: listy wieloznaczne),
- dla responsywności podzielone na fragmenty (np. co 1mln segmentów),
- opcjonalny meta-indeks (indeks na poziomie powyższych fragmentów).
- Dodatkowo niezależny indeks dla zapytań prostych: (pod)ciągi segmentów bez spacji. Poziom zdań.

Rozmiary w praktyce (NKJP):

- same dokumenty: ok 400b/seg
- indeks zapytań prostych: 15b/seg
- indeks węzłowy: ok 150b/seg
- rozszerzenie o sąsiadów: ok 800b/seg

- Indeksy dla każdego atrybutu na poziomie węzłów (uwaga: listy wieloznaczne),
- dla responsywności podzielone na fragmenty (np. co 1mln segmentów),
- opcjonalny meta-indeks (indeks na poziomie powyższych fragmentów).
- Dodatkowo niezależny indeks dla zapytań prostych: (pod)ciągi segmentów bez spacji. Poziom zdań.

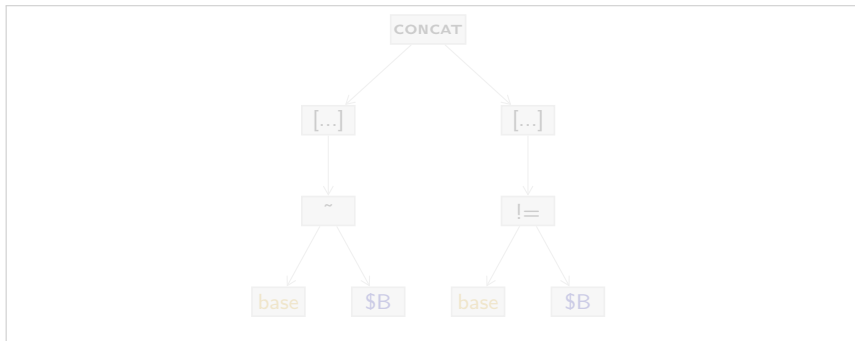
Rozmiary w praktyce (NKJP):

- same dokumenty: ok 400b/seg
- indeks zapytań prostych: 15b/seg
- indeks węzłowy: ok 150b/seg
- rozszerzenie o sąsiadów: ok 800b/seg

Wykonawca a cechy języka



- zmienne $\rightsquigarrow$ efekty uboczne $\rightsquigarrow$ wykonanie sekwencyjne
- specyfikacje, wieloznaczność $\rightsquigarrow$ rozgałęzienie wykonania
- **stan** programu: węzeł (w zapytaniu) + wartościowanie

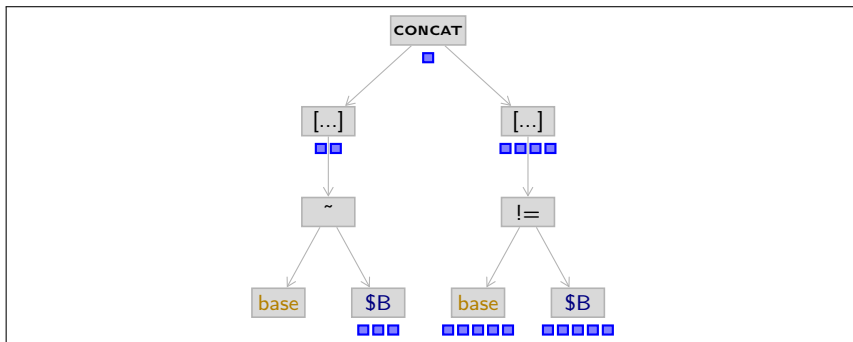


- “zwykłe” silniki RE: BFS $\leftrightarrow$ agresywna priorytyzacja, ale u nas:
 - wybór hojnej semantyki
 - większy stopień rozdrzewienia (i późnego filtrowania)
- nieliniowy odczyt wejścia; więcej zmiennych (np. pod flagę /NI!)

Wykonawca a cechy języka



- zmienne $\rightsquigarrow$ efekty uboczne $\rightsquigarrow$ wykonanie sekwencyjne
- specyfikacje, wieloznaczność $\rightsquigarrow$ rozgałęzienie wykonania
- **stan** programu: **węzeł** (w zapytaniu) + **wartościowanie**

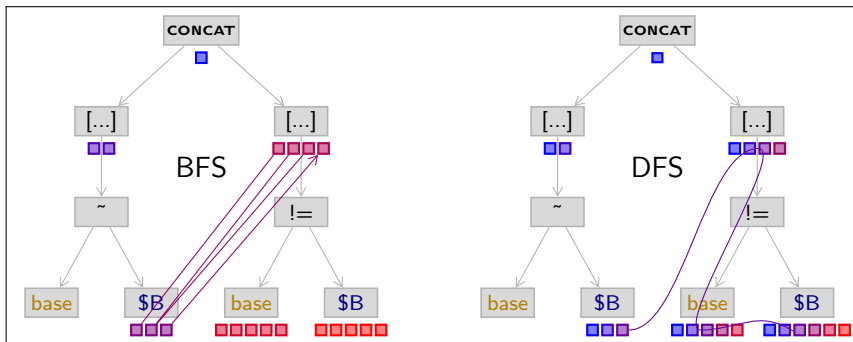


- “zwykłe” silniki RE: BFS $\leftrightarrow$ agresywna priorytyzacja, ale u nas:
 - wybór hojnej semantyki
 - większy stopień rozdrzewienia (i późnego filtrowania)
- nieliniowy odczyt wejścia; więcej zmiennych (np. pod flagę /NI!)



Wykonawca a cechy języka

- zmienne $\rightsquigarrow$ efekty uboczne $\rightsquigarrow$ wykonanie sekwencyjne
- specyfikacje, wieloznaczność $\rightsquigarrow$ rozgałęzienie wykonania
- **stan** programu: **węzeł** (w zapytaniu) + **wartościowanie**

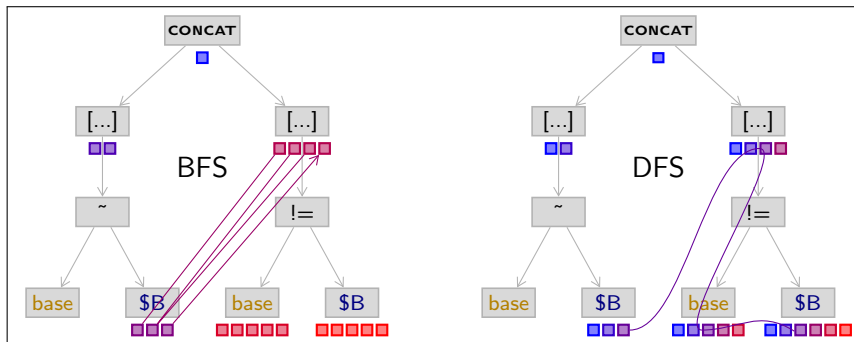


- “zwykłe” silniki RE: BFS $\leftrightarrow$ agresywna priorytyzacja, ale u nas:
 - wybór hojnej semantyki
 - większy stopień rozdrzewienia (i późnego filtrowania)
- nieliniowy odczyt wejścia; więcej zmiennych (np. pod flagę `/NI!`)



Wykonawca a cechy języka

- zmienne $\rightsquigarrow$ efekty uboczne $\rightsquigarrow$ wykonanie sekwencyjne
- specyfikacje, wieloznaczność $\rightsquigarrow$ rozgałęzienie wykonania
- **stan** programu: **węzeł** (w zapytaniu) + **wartościowanie**



- “zwykłe” silniki RE: BFS $\leftrightarrow$ agresywna priorytyzacja, ale u nas:
 - wybór hojnej semantyki
 - większy stopień rozdrzewienia (i późnego filtrowania)
- nieliniowy odczyt wejścia; więcej zmiennych (np. pod flagę `/NI!`)

DFS w Poliqarpie



- Zalety:
 - Szybciej pierwsze (na ogół jedyne) dopasowanie w zdaniu
 - Mniejszy narzut na wizytę w węźle
Pozwala na rozdrobnienie i specjalizację węzłów
- `return [...]` $\rightsquigarrow$ `for ... yield;` programowanie zdarzeniami
- Skutki uboczne — komplikacja kodu
 - stos wywołań: historia przodków $\rightsquigarrow$ historia in-order
 - wspomagany profiling
 - przerywanie wykonania $\rightsquigarrow$ własna obsługa wyjątków
 - ?, *, + jednak BFS-em...

DFS w Poliqarpie



- Zalety:
 - Szybciej pierwsze (na ogół jedyne) dopasowanie w zdaniu
 - Mniejszy narzut na wizytę w węźle
Pozwala na rozdrobnienie i specjalizację węzłów
- `return [...]` $\rightsquigarrow$ `for ... yield;` programowanie zdarzeniami
- Skutki uboczne — komplikacja kodu
 - stos wywołań: historia przodków $\rightsquigarrow$ historia in-order
 - wspomagany profiling
 - przerywanie wykonania $\rightsquigarrow$ własna obsługa wyjątków
 - ?, *, + jednak BFS-em...

Optymalizacje



- Do poskromienia: **masywne kopiowanie** wartościowań (walka o czas **oraz** pamięć!)
- Techniki:
 - Usuwanie niepotrzebnych zmiennych
 - Przenoszenie zmiennych z wartościowań do węzłów
 - Copy-on-write
 - Czyszczenie zużytych wartościowań
- Problemy:
 - semantyka: “rollback on every failure”
 - pętle
 - zmienne specjalne ($\$_Q$ — wynik, $\$_V$ — warianty)

- Do poskromienia: **masywne kopiowanie** wartościowań (walka o czas **oraz** pamięć!)
- Techniki:
 - Usuwanie niepotrzebnych zmiennych
 - Przenoszenie zmiennych z wartościowań do węzłów
 - Copy-on-write
 - Czyszczenie zużytych wartościowań
- Problemy:
 - semantyka: “rollback on every failure”
 - pętle
 - zmienne specjalne (**$\$_Q$** — wynik, **$\$_V$** — warianty)

Pre-processing dla wykonawcy



- Typizer:
 - statyczny (po co? wydajność, bezpieczeństwo)
 - polimorfizm
 - nieograniczone niejawne konwersje
 - niejawne zmienne — brak prostej dedukcji w pionie
 - mapowanie atrybutów (**msd.case**)
 - hierarchia typów
- Lukry: [**case**~nom] $\rightsquigarrow$ [**msd.all** contains [**case**=nom]] itp.
- Zmienne związane: [**base**~\$B], \$X >* (\$Y \$Z) itp.
- Kontekst wyrażenia regularnego: poziome / pionowe / listowe
- Zarządca obiektów (zasadniczo część wczytywacza):
 - pre-processing danych dla poziomych wyrażeń regularnych
 - grupowanie obiektów wg typu, priorytetyzacja
 - uruchamianie wykonawcy dla małych kontekstów pionowych

Pre-processing dla wykonawcy



- Typizer:
 - statyczny (po co? wydajność, bezpieczeństwo)
 - polimorfizm
 - nieograniczone niejawne konwersje
 - niejawne zmienne — brak prostej dedukcji w pionie
 - mapowanie atrybutów (**msd.case**)
 - hierarchia typów
- Lukry: [**case**~nom] $\rightsquigarrow$ [**msd.all** contains [**case**=nom]] itp.
- Zmienne związane: [**base**~\$B], \$X >* (\$Y \$Z) itp.
- Kontekst wyrażenia regularnego: poziome / pionowe / listowe
- Zarządca obiektów (zasadniczo część wczytywacza):
 - pre-processing danych dla poziomych wyrażeń regularnych
 - grupowanie obiektów wg typu, priorytetyzacja
 - uruchamianie wykonawcy dla małych kontekstów pionowych

Pre-processing dla wykonawcy



- Typizer:
 - statyczny (po co? wydajność, bezpieczeństwo)
 - polimorfizm
 - nieograniczone niejawne konwersje
 - niejawne zmienne — brak prostej dedukcji w pionie
 - mapowanie atrybutów (**msd.case**)
 - hierarchia typów
- Lukry: [**case**~nom] $\rightsquigarrow$ [**msd.all** **contains** [**case**=nom]] itp.
- Zmienne związane: [**base**~\$B], \$X >* (\$Y \$Z) itp.
- Kontekst wyrażenia regularnego: poziome / pionowe / listowe
- Zarządca obiektów (zasadniczo część wczytywacza):
 - pre-processing danych dla poziomych wyrażeń regularnych
 - grupowanie obiektów wg typu, priorytetyzacja
 - uruchamianie wykonawcy dla małych kontekstów pionowych

Pre-processing dla wykonawcy



- Typizer:
 - statyczny (po co? wydajność, bezpieczeństwo)
 - polimorfizm
 - nieograniczone niejawne konwersje
 - niejawne zmienne — brak prostej dedukcji w pionie
 - mapowanie atrybutów (**msd.case**)
 - hierarchia typów
- Lukry: [**case**~nom] $\rightsquigarrow$ [**msd.all contains** [**case**=nom]] itp.
- Zmienne związane: [**base**~\$B], \$X >* (\$Y \$Z) itp.
- Kontekst wyrażenia regularnego: poziome / pionowe / listowe
- Zarządca obiektów (zasadniczo część wczytywacza):
 - pre-processing danych dla poziomych wyrażeń regularnych
 - grupowanie obiektów wg typu, priorytetyzacja
 - uruchamianie wykonawcy dla małych kontekstów pionowych

Pre-processing dla wykonawcy



- Typizer:
 - statyczny (po co? wydajność, bezpieczeństwo)
 - polimorfizm
 - nieograniczone niejawne konwersje
 - niejawne zmienne — brak prostej dedukcji w pionie
 - mapowanie atrybutów (**msd.case**)
 - hierarchia typów
- Lukry: [**case**~nom] $\rightsquigarrow$ [**msd.all contains** [**case**=nom]] itp.
- Zmienne związane: [**base**~\$B], \$X >* (\$Y \$Z) itp.
- Kontekst wyrażenia regularnego: poziome / pionowe / listowe
- Zarządca obiektów (zasadniczo część wczytywacza):
 - pre-processing danych dla poziomych wyrażeń regularnych
 - grupowanie obiektów wg typu, priorytetyzacja
 - uruchamianie wykonawcy dla małych kontekstów pionowych

NKJP: konfrontacja założeń z rzeczywistością

Problem 1: Co jest dokumentem?

- Pojedynczy artykuł w gazecie?
 - Pojedyncze wydanie gazety?
 - Rocznik gazety?
-
- W NKJP nie ma spójnej konwencji.
 - Pomysły: **plik XML**, *< bibitem >*.
 - Ze względu na wydajność chcemy mieć małe dokumenty.
 - Praktyczne podejście:
 - konwerter (opcjonalnie) dzieli wejście na akapity,
 - wczytywacz zakłada, że użytkownik wie co robi.
- i tak jest kiepsko przy słabej segmentacji (PELCRA...).

NKJP: konfrontacja założeń z rzeczywistością

Problem 1: Co jest dokumentem?

- Pojedynczy artykuł w gazecie?
 - Pojedyncze wydanie gazety?
 - Rocznik gazety?
-
- W NKJP nie ma spójnej konwencji.
 - Pomysły: **plik XML**, *< bibitem >*.
 - Ze względu na wydajność chcemy mieć małe dokumenty.
 - Praktyczne podejście:
 - konwerter (opcjonalnie) dzieli wejście na akapity,
 - wczytywacz zakłada, że użytkownik wie co robi.
- i tak jest kiepsko przy słabej segmentacji (PELCRA. . .).

NKJP: konfrontacja założeń z rzeczywistością

Problem 1: Co jest dokumentem?

- Pojedynczy artykuł w gazecie?
 - Pojedyncze wydanie gazety?
 - Rocznik gazety?
-
- W NKJP nie ma spójnej konwencji.
 - Pomysły: **plik XML**, *< bibitem >*.
 - Ze względu na wydajność chcemy mieć małe dokumenty.
 - Praktyczne podejście:
 - konwerter (opcjonalnie) dzieli wejście na akapity,
 - wczytywacz zakłada, że użytkownik wie co robi.
- i tak jest kiepsko przy słabej segmentacji (PELCRA. . .).

NKJP: konfrontacja założeń z rzeczywistością

Problem 2: krzaki w danych korpusowych

- Fora internetowe – napisy bez sensu, bez spacji. . .
 - fragmenty kodu źródłowego oprogramowania,
 - ASCII-Art (wyniki słabego OCR?)
-
- Długie ciągi segmentów bez spacji napychają indeks dla zapytań prostych, ale chcemy obsługiwać np.
chciał|że|by|m|.|. .
 - Rozwiązanie praktyczne: limit długości rozbijanego ciągu, dłuższych niż 20 segmentów nie rozbijamy.

NKJP: konfrontacja założeń z rzeczywistością

Problem 2: krzaki w danych korpusowych

- Fora internetowe – napisy bez sensu, bez spacji...
 - fragmenty kodu źródłowego oprogramowania,
 - ASCII-Art (wyniki słabego OCR?)
-
- Długie ciągi segmentów bez spacji napychają indeks dla zapytań prostych, ale chcemy obsługiwać np.
`chciał|że|by|m|.|. .`
 - Rozwiązanie praktyczne: limit długości rozbijanego ciągu, dłuższych niż 20 segmentów nie rozbijamy.

Plan



- 1 Wstęp
- 2 Przypomnienie modelu danych
- 3 Język zapytań w działaniu
- 4 Implementacja
- 5 Na zakończenie**
 - Wydajność
 - Przyszłość

Wstępne pomiary wydajności



zapytanie	wyniki:		Poliqarp 1.x		Poliqarp 2.0		#wyników
			1-10	max	1-10	max	
[base=po] [orth=polsku]			0.1s	0.1s	0.6s	2.8s	96
[]*[base=po] [orth=polsku] within p			18.2s	187s	0.9s	10.3s	88
[base=po]			0.1s	0.6s	0.3s	272s	25730
[{base,number}={marszałek,pl}]			–	–	0.5s	8.2s	339
[type="syngrr:.*NG"]>[pos=Adj]			–	–	1.3s	7.9s	75

Dane testowe

Fragmencik korpusu NKJP, 10mln segmentów.

Z przyczyn m.in. opisanych wcześniej pełen NKJP nie jest gotowy:(

Plany na najbliższą przyszłość:



- doprowadzenie do porządku indeksowania pełnego NKJP...
- testy pół-publiczne,
- publikacja kodu i dokumentacji, wdrożenie na nkjp.pl,

Kontynuacja prac

Dostosowanie na potrzeby korpusu KorBa, w szczególności:
klikalny kreator zapytań w GUI.

Plany na najbliższą przyszłość:



- doprowadzenie do porządku indeksowania pełnego NKJP...
- testy pół-publiczne,
- publikacja kodu i dokumentacji, wdrożenie na nkjp.pl,

Kontynuacja prac

Dostosowanie na potrzeby korpusu KorBa, w szczególności:
klikalny kreator zapytań w GUI.

Plany na najbliższą przyszłość:



- doprowadzenie do porządku indeksowania pełnego NKJP...
- testy pół-publiczne,
- publikacja kodu i dokumentacji, wdrożenie na nkjp.pl,

Kontynuacja prac

Dostosowanie na potrzeby korpusu KorBa, w szczególności:
klikalny kreator zapytań w GUI.