



Wyszukiwarka tekstowa w Django oparta na PostgreSQL

Kim jestem?



solvbot

Założyciel i CTO w Solvbot
Tworzymy oprogramowanie SaaS
dla branży prawnej
(legal-tech i fin-tech)
Oczywiście rekrutujemy!

12 lat na stanowiskach technicznych

Startupy
Zarządzanie produktem
Budowanie zespołu

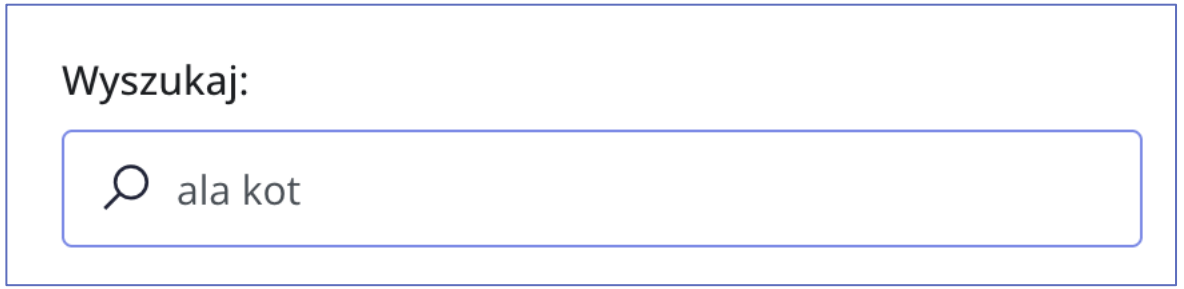
Tematy na dzisiaj

- Rozwiązanie „naiwne”
- Zaawansowane narzędzia
- Faktyczne potrzeby użytkownika
- Benchmark
- Implementacja

solvbot

__icontains

Text:
Ala ma kota. Kot lubi mleko.

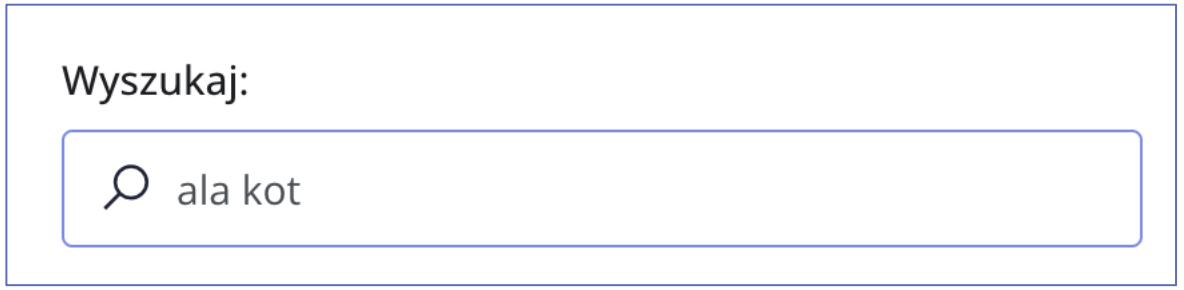


Query:

```
.filter(text__icontains='ala kot')
```

`__icontains`

Text:
Ala ma kota. Kot lubi mleko.



Query:

```
.filter(text__icontains='ala kot')
```

Problem:
Kolejność wyrazów

solvbot

__icontains + split

Text:
Ala ma kota. Kot lubi mleko.

Query:

```
.filter(
   text__icontains='ala',
   text__icontains='kot'
)
```

Split musi być „sprytny” - znaki przestankowe (przecinki, myślniki itp.)
`Np. 'ala, kot'`

solvbot

Polskie znaki (akcenty)

Text:
Łukasz ma psa.

Query:

```
.filter(text__icontains='lukasz’)
```

Problem: unaccent, performance.

__icontains

Text:

Łukasz ma psa malamuta.

Query:

```
.filter(text__icontains='ala')
```

solvbot

__icontains

Text:

Łukasz ma psa m**ala**muta.

Query:

```
.filter(text__icontains='ala')
```

Problem:

__icontains

solvbot

`__icontains`

Text:

Łukasz ma psa m**ala**muta.

Query:

```
.filter(text__icontains='ala')
```

Problem:

`__i`**`contains`**

solvbot

__istartswith, __iendswith

Text:
Łukasz ma psa m**ala**muta.

Query:

```
.filter(
        text__istartswith='ala ',
        text__icontains=' ala ',
        text__iendswith=' ala',

        text__istartswith='kot ',
        text__icontains=' kot ',
        text__iendswith=' kot',
)
```

solvbot

__istartswith, __iendswith

Text:

Łukasz ma psa m**ala**muta.

Query:

```
.filter(
        text__istartswith='ala ',
        text__icontains=' ala ',
        text__iendswith=' ala',

        text__istartswith='kot ',
        text__icontains=' kot ',
        text__iendswith=' kot’,
)
```

Problem:

Text:

Ala, ala.

Wcześniej `split` był w aplikacji, teraz potrzebny jest preprocessing tekstów w baze.

Performance? 😢

solvbot

Podsumowując

„Łukasz ma kota.”

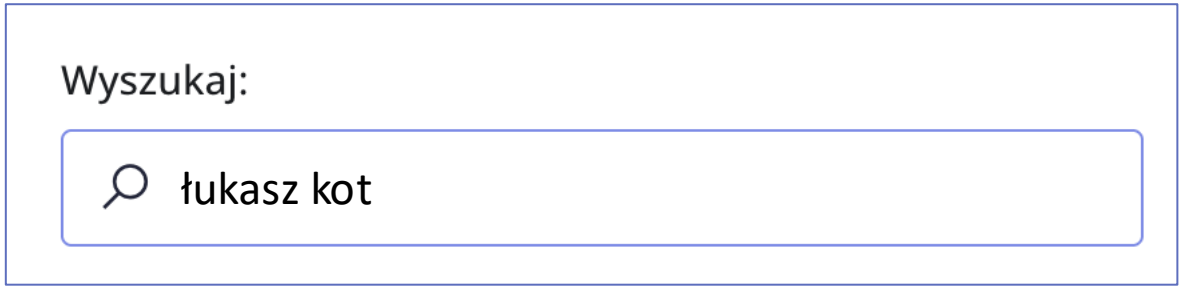
- Unaccent
- Lowercase
- Split (+ znaki przestankowe)

„lukasz ma kota”
+ indeks (performance)

solvbot

Mamy to?

Text:
Łukasz ma kota.



solvbot

Gramatyka

Text:
Łukasz ma kota.

Wyszukaj:

łukasz kot



__icontains nie jest idealny do wyszukiwania

solvbot

Tematy na dzisiaj

- Rozwiązanie „naiwne”
- **Zaawansowane narzędzia**
- Faktyczne potrzeby użytkownika
- Benchmark
- Implementacja

solvbot

Stemming

Usunięcie końcówki fleksyjnej z wyrazu.

„Łukasza” -> „Łukasz”

Stripping – usuwanie przedrostków i końcówek
„najlepiej” -> „lepiej” -> „lep…” szy?

Lematyzacja – używa gramatyki, identyfikuje części mowy – bardziej poprawny, bardziej skomplikowany

solvbot

Stemming

Przykład:

„Łukasz ma psa” -> [łukasz, mieć, pies]

Ranking wyników

Tekst 1:
„Łukasz ma psa malamuta.”

Tekst 2:
„Ala ma kota, ma rybki i ma misia.”



solvbot

Ranking wyników

Tekst 1:
„Ala ma psa malamuta.”

Tekst 2:
„Ala ma kota, ma rybki i ma katar.”



solvbot

Wagi wyrazów

TF-IDF – term frequency – inverse document frequency

TF – jak często szukane słowo występuje w danym dokumencie

IDF – jak dużo informacji zawiera dane słowo, tzn. jakie jest popularne we wszystkich dokumentach (mniej pospolite = lepiej)

Korekcja błędów

Można korygować gramatycznie.

Trudny problem: dokumenty z OCR

- Zawierają nazwy własne (np. Sieradzki, Tauron).
- Najwięcej błędów rozpoznawania w słowach spoza słownika.
- Nazwy własne – największa zawartość informacji.

Odległość edycyjna (Levenshteina)?

Ciekawe narzędzia

Whoosh – czysty Python

Apache Lucene

> *Lucene Core is a Java library providing powerful indexing and search features, as well as spellchecking, hit highlighting and advanced analysis/tokenization capabilities.*

Narzędzia zbudowane na jej podstawie:

- PyLucene (Python bindings)
- Solr
- Elasticsearch

Tematy na dzisiaj

- Rozwiązanie „naiwne”
- Zaawansowane narzędzia
- **Faktyczne potrzeby użytkownika**
- Benchmark
- Implementacja

solvbot

Możemy ograniczyć funkcjonalność?

Solvbot – setki tysięcy dokumentów, każdy po kilka stron A4.

Poza tym:

- imiona i nazwiska
- nazwy instytucji
- tytuły (frazy wpisane przez użytkownika)

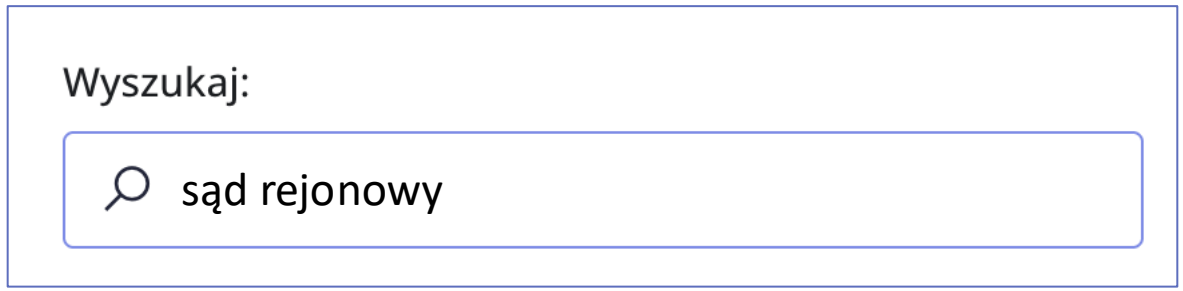
  dokumentów, zadań, zgłoszeń, wiadomości, spotkań itp.

Tu wyszukiwanie nie wymaga gramatyki!

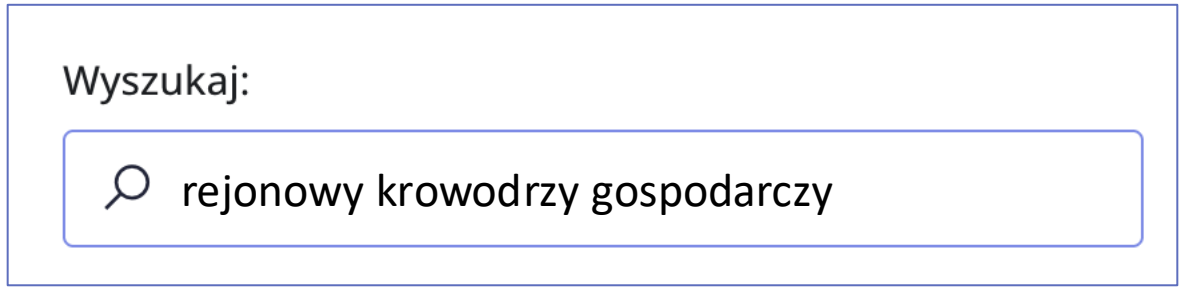
solvbot

Problem użytkownika

Sąd Rejonowy dla Krakowa-Krowodrzy w Krakowie, XV Wydział Gospodarczy



~300 sądów po 10 wydziałów



Dużo pisania

solvbot

Problem użytkownika

Sąd Rejonowy dla Łodzi-Śródmieścia w Łodzi, XV Wydział Gospodarczy

Wyszukaj:

rej lodz gosp



solvbot

To już wiemy

```
.filter(
      text__istartswith='lodz',
      text__icontains=' lodz’,
)
```

Preprocessing w bazie:

- Unaccent
- Lowercase
- Split (+ znaki przestankowe)

+ indeks (performance)

solvbot

Tematy na dzisiaj

- Rozwiązanie „naiwne”
- Zaawansowane narzędzia
- Faktyczne potrzeby użytkownika
- **Benchmark**
- Implementacja

solvbot

Benchmark

```
CREATE TABLE t_random
AS
SELECT i, random_string(5, 8) AS text
FROM generate_series(1, 5000000) AS i;
```



bit.ly/33RXfMk

500K wierszy tekstu

Każdy 5 słów po 8 znaków

1072MB na dysku

solvbot

Bez indeksów

```
select * from t_random where ...

text like '%łódź%’;
-- 372ms

text ilike '%łódź%’;
-- 643ms

upper(text) like upper('%łódź%’);
-- 624ms

unaccent(text) ilike unaccent('%łódź%’);
-- 1551ms
```

solvbot

Bez indeksów

Nasz cel:

```
select * from t_random where
(
      unaccent(text) ilike unaccent('łódź%') or
      unaccent(text) ilike unaccent('% łódź%’)
) and (
      unaccent(text) ilike unaccent('gosp%') or
      unaccent(text) ilike unaccent('% gosp%’)
);

-- 2932ms
```

solvbot

Trigram

```
# select show_trgm('sad rejonowy’);

                                  show_trgm
-----------------------------------------------------------------------
 {"  r","  s"," re"," sa","ad ",ejo,jon,now,ono,owy,rej,sad,"wy "}
```

solvbot

Indeksy w PostgreSQL

GiST

Każdy document jest przechowywany jako signatura –
indeks jest stratny.
Sygnatura powstaje z sumy binarnej hashów każdego z leksemów
w tekście.

GIN

Indeks nie jest stratny, działa coraz wolniej dla dłuższych
dokumentów.
Nie polecany powyżej 100K leksemów.

solvbot

Indeksy GIN i GiST

	GIN
Wyszukiwanie	3x szybciej
Budowa indeksu	3x wolniej
Aktualizacja indeksu	Trochę wolniej (*)
Rozmiar indeksu	3x większy
Duże dokumenty	Gorzej przy 10K leksemów

* Wymaga fast-update

Benchmark GIN

```
CREATE INDEX idx_gin ON t_random
USING gin (text gin_trgm_ops);

-- 45 sec, size 632MB
```

Benchmark GIN

```
select * from t_random where ...

text ilike '%łódź%’;
-- 394ms (było 643ms)

upper(text) like upper('%łódź%’);
-- 609ms (było 624ms), uses Seq Scan

unaccent(text) ilike unaccent('%łódź%’);
-- 1565ms (było 1551ms), uses Seq Scan
```

solvbot

GIN + unaccent

```
CREATE OR REPLACE FUNCTION imm_unaccent(text) RETURNS text AS
$$
SELECT public.unaccent('public.unaccent', $1)
$$ LANGUAGE sql IMMUTABLE;


CREATE INDEX idx_gin_unaccent ON t_random
USING gin (imm_unaccent(text) gin_trgm_ops);


-- 58sec, 724MB (bez unaccent: 45 sec, size 632MB)
```

solvbot

GIN + unaccent

Nasz cel:

```
select * from t_random where
(
      imm_unaccent(text) ilike imm_unaccent('łódź%') or
      imm_unaccent(text) ilike imm_unaccent('% łódź%’)
) and (
      imm_unaccent(text) ilike imm_unaccent('gosp%') or
      imm_unaccent(text) ilike imm_unaccent('% gosp%’)
);

-- 81ms, używa Bitmap Heap Scan (było 2932ms)
```

solvbot



VERY NICE!

Uwaga!

```
select * from t_random where
text ilike '%łódź%’;

-- 641ms (było 81ms)
-- wraca do Seq Scan
```

solvbot

Tematy na dzisiaj

- Rozwiązanie „naiwne”
- Zaawansowane narzędzia
- Faktyczne potrzeby użytkownika
- Benchmark
- **Implementacja**

solvbot

Paczka

```
# TODO...
```

DEMO!

rej krow gosp

solvbot

Korespondencja przychodząca w post. karne Artur Bors, Dębica

Nadawca: Utwórz nowy kontakt

---

rej krow gosp

Sąd Rejonowy dla Krakowa - Krowodrzy w Krakowie, XV Wydział Gospodarczy

Sędzia Komisarz Sędzia Wyznaczony - Sąd Rejonowy dla Krakowa - Krowodrzy w Krakowie, XV Wydział Gospodarczy

Treść:

solvbot

Co dalej?

Solvbot - setki tysięcy dokumentów, każdy po kilka stron A4

- Dokumenty PDF - OCR
- Full text search
- eDiscovery
- proces discovery

solvbot

Szymon Sobczak

szymon@solvbot.pl

https://www.linkedin.com/in/szymonsobczak/

solvbot

www.solvbot.pl/praca