

Uniwersytet Kardynała Stefana Wyszyńskiego  
w Warszawie  
Wydział Matematyczno-Przyrodniczy  
Szkoła Nauk Ścisłych

Jacek Giedronowicz

Nr albumu: 95175

# Rekomendacja stron www za pomocą silnika Apache Lucene

Praca licencjacka na kierunku *Informatyka*  
w zakresie *Machine Learning*

Praca wykonana pod kierunkiem  
**dr. Roberta Kłopotka**

Lipiec 2020

**Słowa kluczowe**

-

**Dziedzina Socrates-Erasmus**

11.3 Informatyka

**Klasyfikacja tematyczna**

Machine Learning

**English title**

Website recommendation using the Apache Lucene engine

# Spis treści

<b>1. Wprowadzenie</b>	5
<b>2. Stan rynku</b>	7
2.1. Wyszukiwarka	8
2.2. Wyszukiwanie	9
2.3. Podstawowe operatory	11
<b>3. Rekomendacja stron www</b>	13
3.1. Proces indeksowania	13
3.1.1. Document	13
3.1.2. Struktura indeksu	14
3.1.3. Analizer	15
3.2. Proces wyszukiwania	16
3.2.1. TF-IDF Similarity	17
3.2.2. PageRank	17
3.3. Dodatkowe funkcje składni zapytania	17
<b>4. Przedstawienie problemu i sposób jego rozwiązania</b>	19
<b>5. Funkcjonalność zaimplementowanego systemu</b>	21
<b>6. Implementacja oraz użyte technologie</b>	23
6.1. Java	23
6.1.1. Apache Lucene	23
6.2. Spring Boot	23
6.3. Thymeleaf	23
6.4. HTML	23
6.5. Struktura i działanie klas programu	23
6.5.1. Klasy silnika wyszukiwania	23
6.5.2. Klasy kontrolera	23
<b>7. Przypadki użycia</b>	25
<b>8. Podsumowanie</b>	27



# Rozdział 1

## Wprowadzenie

- 
- ok 1-2 str
  - ogólne zagadnienie jak jest stosowane w praktyce (do przewidywania)
  - opis, że klasyfikacja jest trudna, używa się wielu metod, że jednym z podejść jest komitet klasyfikatorów, aby poprawić skuteczność
  - cel pracy, hipotez badawczych
  - krótki opis co jest w danym rozdziale
-



## Rozdział 2

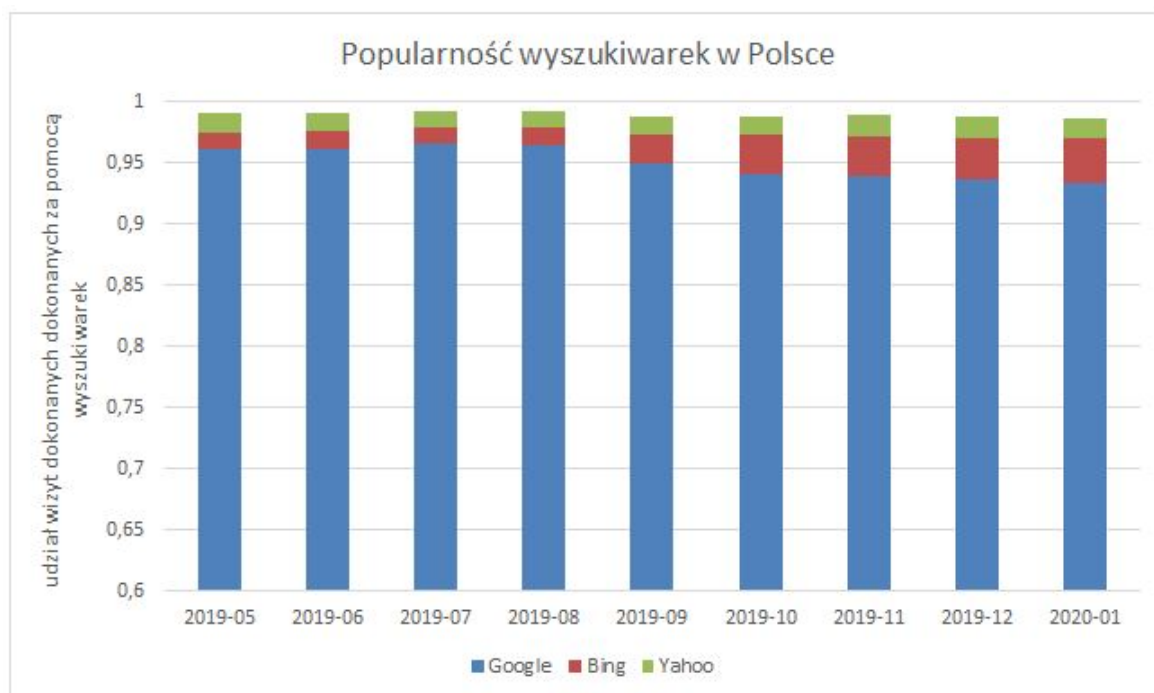
# Stan rynku

- 
- przede wszystkim zdefiniowanie problemu biznesowego i jego uzasadnienie.
  - Jakie są wyszukiwarki?
  - Jakie mają opcje?
  - strony spammerskie (zaburzają ranking)
  - hint: zobacz prace lic o serwisie otomoto

---

Na rynku istnieje wiele wyszukiwarek internetowych, lecz mało kto potrafi wymienić więcej niż pięć. Internet zdominowała firma Google. Nawet takie znane wyszukiwarki jak Bing od Microsoft'u czy Yahoo mają zaledwie kilka procent udziału, który przedstawia się następująco:

- Google - 93,37%
- Bing - 3,61%
- Yahoo - 1,75%
- DuckDuckGo - 0,29%
- Interia Katalog - 0,14%



Rysunek 2.1: Popularność wyszukiwarek w Polsce

Źródło: gs.statcounter.com – styczeń 2020

W Polsce, z powodu trudności w przetwarzaniu naszego języka przez zagraniczne systemy, były próby stworzenia wyszukiwarek dedykowane dla polaków. Systemy z algorytmami rozpoznawania odmian słów w języku polskim. Największe powodzenie miała witryna szukacz.pl, która funkcjonowała przez 10 lat od 2001 do 2011 roku. Jak możemy przeczytać na stronie [http://szukacz.pl/wiecej\\_o\\_szukaczu.html](http://szukacz.pl/wiecej_o_szukaczu.html) [7]:

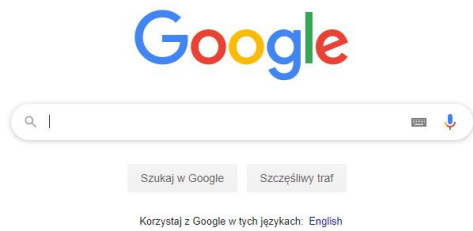
"W szczycie swojego rozwoju, pod koniec 2007 roku, odpowiadał ze 115 milionów dokumentów w języku polskim pochodzących z miliona witryn (kolekcja „Polska”). Do połowy 2007 roku odpowiadał także z 45 milionów wyselekcjonowanych dokumentów w języku angielskim pochodzących z 2 milionów witryn (kolekcja „Świat”; tylko 4 procent pytań było skierowane do tej kolekcji)."

Mimo sporej bazy dokumentów oraz przyzwoitym budżecie, również i ta witryna musiała się poddać międzynarodowemu gigantowi jakim jest Google. Stąd też wybrałem witrynę google.pl jako wzór, do którego będę się odnosił przy implementacji własnego systemu. W dalszej części opiszę najważniejsze funkcjonalności witryny Google i porównam z Bing.

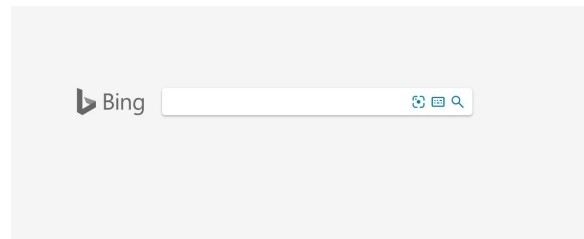
## 2.1. Wyszukiwarka

Na rysunkach 2.2 i 2.3 przedstawiono główne strony wyszukiwarek Google i Bing. Jak widać obie strony są minimalistyczne i intuicyjne. Witryny oferują również możliwość jej spersonalizowania przez bogatą bazę skórek, co zostało przedstawione na rysunkach 2.4 i 2.5

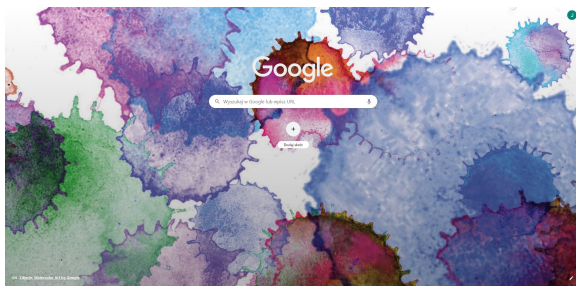




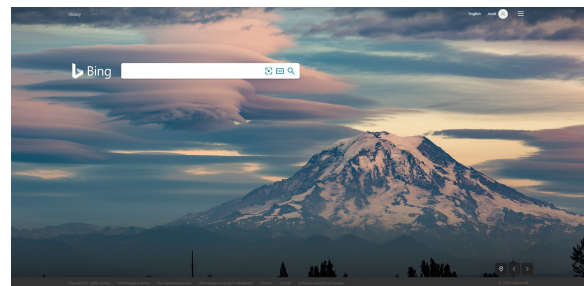
Rysunek 2.2: Główna strona wyszukiwarki google



Rysunek 2.3: Główna strona wyszukiwarki bing



Rysunek 2.4: Główna strona wyszukiwarki Google ze skórka



Rysunek 2.5: Główna strona wyszukiwarki Bing ze skórka

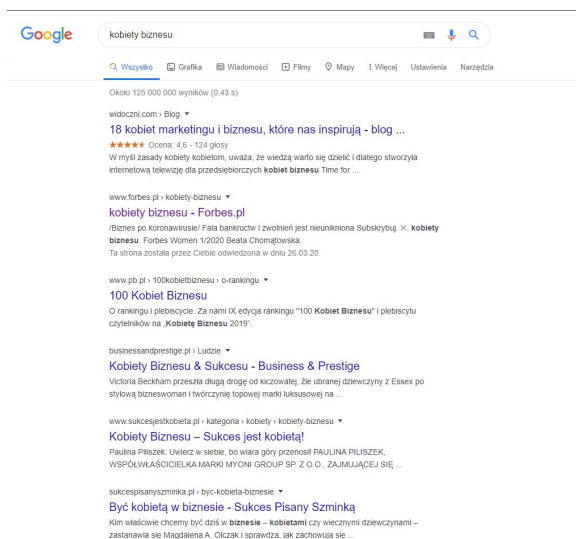
## 2.2. Wyszukiwanie

Po wpisaniu frazy, którą chcemy wyszukać obie wyszukiwarki wypiszą nam listę od kilkudziesięciu tysięcy do nawet kilkuset milionów rekomendowanych stron dla tego hasła.

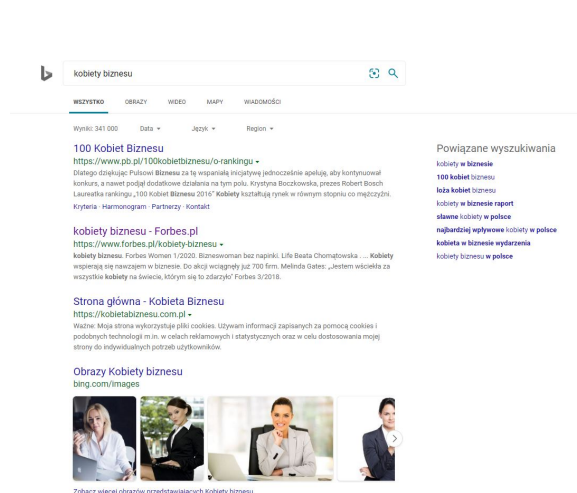
Dla przykładu dla frazy „kobiety biznesu” Google znalazło 125 000 000 wyników kiedy Bing zaledwie 341 000. Dominacja wyszukiwarki Google wśród użytkowników oraz zdecydowanie większa pula zaindeksowanych stron może być powodem dlaczego współcześnie deweloperzy przygotowują swoje strony głównie pod pozycjonowanie wyszukiwarki Google.

Każda z wyświetlanych pozycji przekazuje nam podstawowe informacje takie jak:

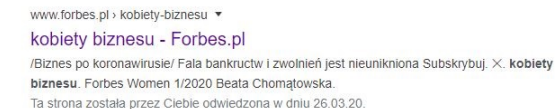
- tytuł strony (zaznaczony kolorem zielonym)
- link do strony (zaznaczony kolorem fioletowym)
- skrót zawartości strony (zaznaczony kolorem brązowym)



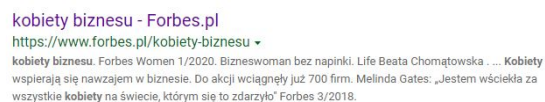
Rysunek 2.6: Lista rekomendowanych przez Google stron dla hasła: kobiety biznesu



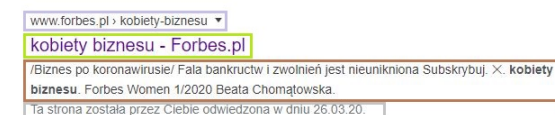
Rysunek 2.7: Lista rekomendowanych przez Bing stron dla hasła: kobiety biznesu



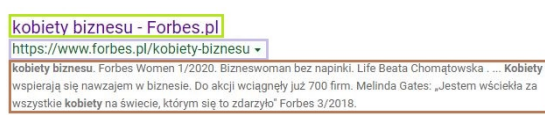
Rysunek 2.8: Wyświetlana pozycja w liście wyników google



Rysunek 2.9: Wyświetlana pozycja w liście wyników bing



Rysunek 2.10: Wyświetlana pozycja w liście wyników Google z zaznaczonymi obszarami

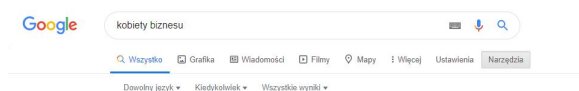


Rysunek 2.11: Wyświetlana pozycja w liście wyników Bing z zaznaczonymi obszarami

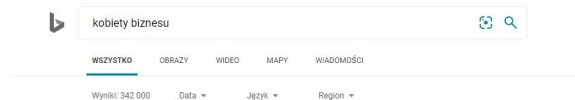
Oprócz podstawowego wyszukiwania linków do stron, mamy możliwość wybrania kategorii przedstawione na rysunkach 2.12 i 2.13 takich jak:

- grafika
- wiadomości
- filmy

Ponadto wyszukiwarki dają nam możliwość intuicyjnego filtrowania wyników. W tym punkcie wyszukiwarki nieznacznie się różnią, gdyż Bing oferuje takie parametry jak: data, język, region. Natomiast Google: data, język i możliwość pokazania dokładnych wyników zważając o wyniki fraz bliskoznacznych.



Rysunek 2.12: Kategorie i filtry wyszukiwania Google



Rysunek 2.13: Kategorie i filtry wyszukiwania Bing

## 2.3. Podstawowe operatory

Aby podnieść jakość i zawęzić pole poszukiwań pożądaných informacji przez użytkownika, Google zaimplementowało tak zwane operatory. Operatory definiują zakres poszukiwań poprzez słowa i znaki kluczowe:

- " " - wpisując wyrażenie w cudzysłów wymusza ścisłe dopasowanie. Przykład: fraza "Steave Jobs" wpisana z cudzysłowem wymusza aby w wyniku były dokładnie te dwa słowa obok siebie.
- OR - operator logiczny pozwalający zapytanie *Steave OR Jobs* wyszukać wyniki zawierające frazę *Steave* lub *Jobs* lub oba. Operator można zastąpić znakiem |.
- AND - domyślny operator logiczny, wyświetlający tylko te wyniki, które zawierają frazę *Steave* i *Jobs*. Operator jest domyślny więc przydatny jest tylko w połączeniu z innymi operatorami.
- – - operator wykluczający. W przykładzie *jaguar speed -car* wynikiem będą strony powiązane z frazą *jaguar speed* ale niezawierające *car*. Zatem chodziło nam o znalezienie prędkości jaguara będącego zwierzęciem a nie samochodem.
- .. - operator zakresu liczb. Przykład: *podatki 2010..2015* da nam wynik wyszukiwania związanymi z podatkami między 2010 a 2015 rokiem.
- ( ) - operator grupowania fraz z innymi operatorami. Przykład: *(iPad OR iPhone)apple*. Operator AND jest domyślny więc nie jest obowiązkowy.
- *define:* - operator wyszukuje definicje szukanej frazy.
- *filetype:* lub *ext:* - operator określający rodzaj pliku będący wynikiem wyszukiwania takie, jak pdf, docx, txt, ppt. Przykład: *Steve Jobs ext:pdf* - wyświetli tylko pliki pdf związane ze Steve'em Jobs'em.
- *site:* - ogranicza wyniki wyszukiwania tylko do jednej strony lub domeny. Przykład: *wniosek site:.gov.pl* wyszuka informacje o wnioskach tylko na stronach rządowych kończących się na *.gov.pl*.
- *inurl:* - operator przeszukuje podaną frazę tylko w adresie url strony.
- *intitle:* - operator przeszukuje podaną frazę tylko w tytule strony.
- *intext:* - operator przeszukuje podaną frazę tylko w treści strony. Pominie strony zawierające frazę w tytule czy adresie url.

Google zawiera znacznie więcej operatorów. Niektóre z nich wymienione są na stronie [sites.google.com](https://www.google.com/sites) [5]



## Rozdział 3

# Rekomendacja stron www

Cały algorytm możemy podzielić na dwa etapy: indeksowanie i wyszukiwanie. Użytkownik oczekuje od aplikacji szybkiego i trafnego wyszukiwania jednak, aby tak się stało trzeba najpierw zebrać wszystkie dane i nadać im specjalną strukturę dzięki której będziemy mogli szybko znaleźć interesującą nas informację. To właśnie nazywamy procesem indeksowania.

### 3.1. Proces indeksowania

#### 3.1.1. Document

Dokument reprezentuje stronę internetową. Jest główną jednostką indeksowania i wyszukiwania. Technicznie jest zbiorem (kontener typu **Set**) zawierający pola (**field**) złożone z klucza i wartości. Dzięki temu możemy zapisać w ustrukturyzowany sposób najważniejsze dla nas informacje, po których później będziemy wyszukiwać. Do wyszukiwania stron www przydatne będą takie pola jak :

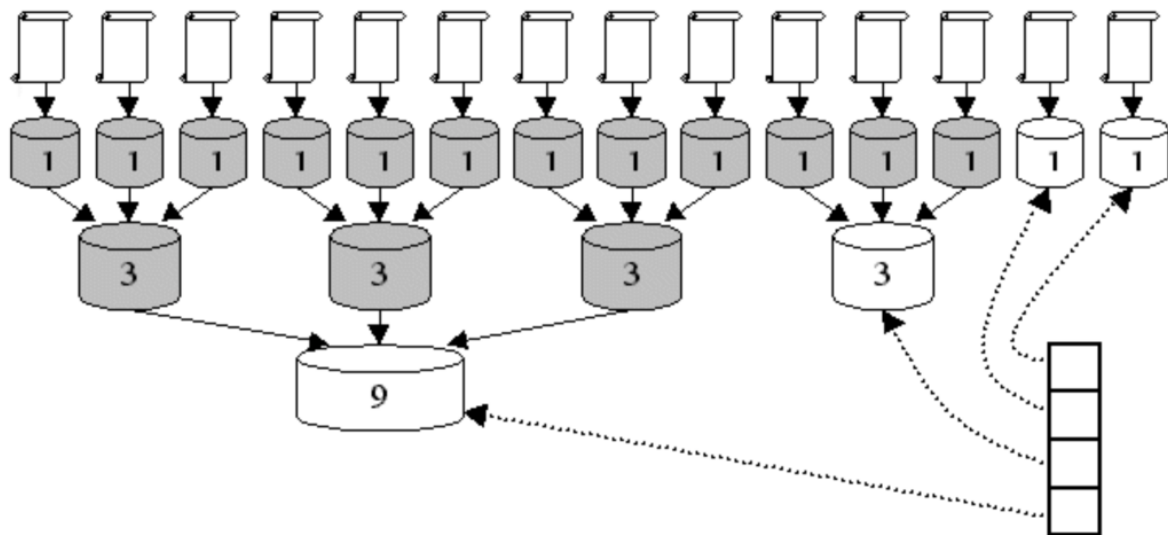
- **id** - unikalny identyfikator dokumentu
- **url** - adres url strony
- **path** - ścieżka do kodu html na dysku twardym
- **title** - tytuł strony
- **keywords** - słowa kluczowe z meta znacznika `<meta name="keywords">`
- **description** - opis strony z meta znacznika `<meta name="description">`
- **headers** - nagłówki ze znaczników `<h1>` do `<h6>`
- **emphasize** - wyróżnione słowa za pomocą znaczników takich jak `<em>`, `<b>`, `<strong>`, `<i>`, `<mark>`
- **images** - informacje o obrazach dostępnych na stronie



Rysunek 3.1: Struktura dokumentu

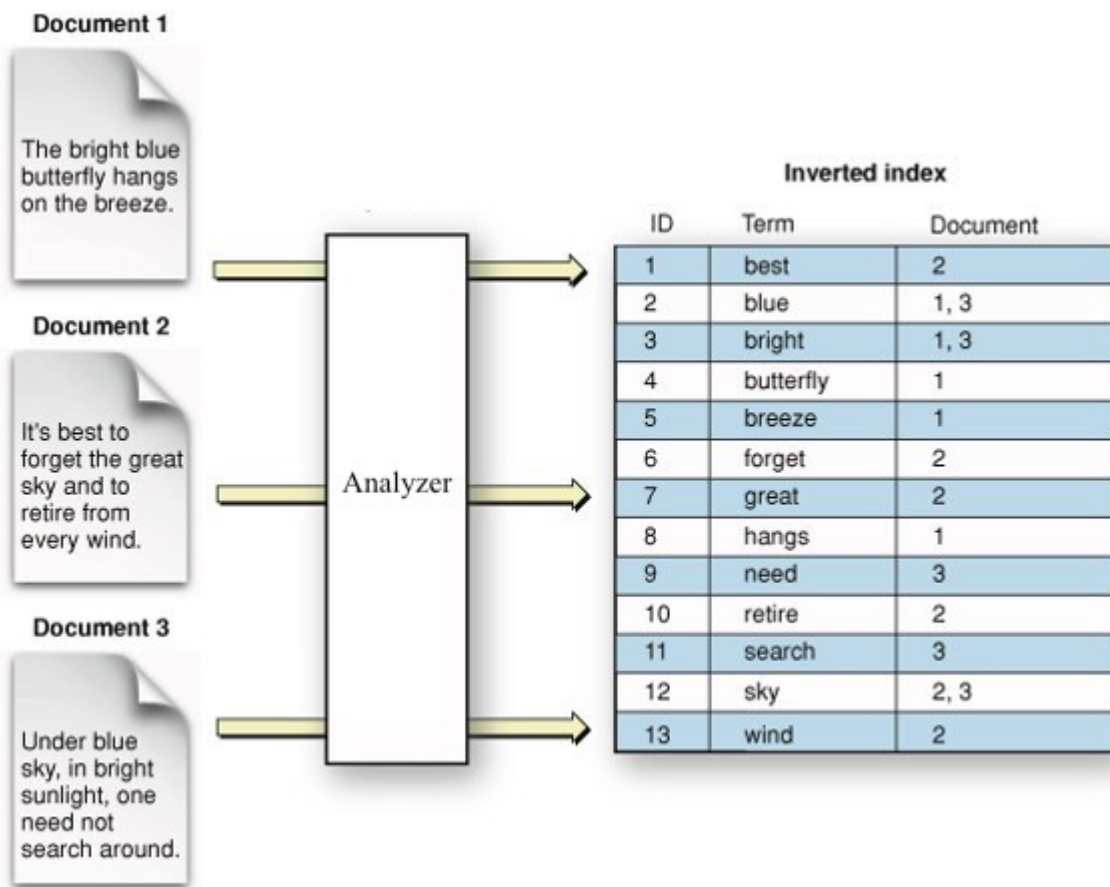
### 3.1.2. Struktura indeksu

Index jest pojedynczym katalogiem na dysku struktury drzewiastej [6]. Składa się z segmentów (**segment**), dokumentów (**document**), pól (**field**), wyrażeń (**term**). Segmenty można traktować jak pod-indeks jednakże nie stanowią one niezależnych indeksów. Liczba tworzących je plików zależy od liczby pól zawartych w indeksie. Wszystkie pliki należące do tego samego segmentu mają taki sam prefiks ale różnią się sufiksem.



Rysunek 3.2: Struktura indeksu [2]

Wyrażenia zapisane w dokumencie (**term**) są analizowane i zapisywane w specjalnej strukturze zwanej inverted index. Polega ona na przedstawieniu relacji **document** - **term** w postaci tabeli przedstawionej na rysunku 3.3. Możemy nie niej szybko znaleźć wyrażenie i id dokumentu, który zawiera dane wyrażenie.



Rysunek 3.3: Struktura indeksu

Powyższy sposób pozwala nam na szybkie znalezienie interesującego nas wyrażenia - dzięki analyzerowi wyrażenia będące uosobieniem tego samego słowa, niezależnie od formy gramatycznej, zajmują jeden rekord w bazie w postaci najprostszego wyrażenia.

### 3.1.3. Analyzer

Analyzer jest mechanizmem analizowania tekstu i generowania tak zwanych **token**ów - będącymi jednostkami leksykalnymi. Dzięki temu dla programu słowa "wyrazić, wyrażenia, wyrażenie" to jedno i to samo.

Dla różnych potrzeb i co ważniejsze dla różnych języków zostały stworzone różne analizery. Przykładowo:

- **WhitespaceAnalyzer**

Rozdziela tekst bazując na białych znakach (spacja). Przykład.

```
["This is an example term analysis test."]
```

↓

```
["This", "is", "an", "example", "term", "analysis", "test."]
```

- **SimpleAnalyzer**

Rozdziela tekst bazując na nieliterowych znakach takich jak liczby, znaki białe czy interpunkcja i zmienia je na małe znaki. Przykład.

```
["This is an 1example term analysis test."]
```

↓

```
["this", "is", "an", "example", "term", "analysis", "test"]
```

- **StopAnalyzer**

Rozdziela tekst bazując na białych znakach. Usuwa interpunkcje oraz takie słowa jak 'a', 'an', 'the' itp. Przykład.

```
["This is an example term analysis test."]
```

↓

```
["example", "term", "analysis", "test"]
```

- **StandardAnalyzer**

Standardowy analizator obsługujący nazwy, adresy email itp. Zmienia wszystkie znaki na małe oraz usuwa interpunkcje i zaimki/przedimki. Przykład.

```
["This is an example term analysis test."]
```

↓

```
["example", "term", "analysis", "test"]
```

- **PolishAnalyzer**

Odpowiedni analizator dla języka polskiego. Zmienia litery na małe, usuwa interpunkcje i co najważniejsze zmienia podane wyrażenia na ich podstawową formę (o ile to możliwe bezokolicznik lub mianownika). Przykład.

```
["To jest przykładowy test analizy wyrażenia."]
```

↓

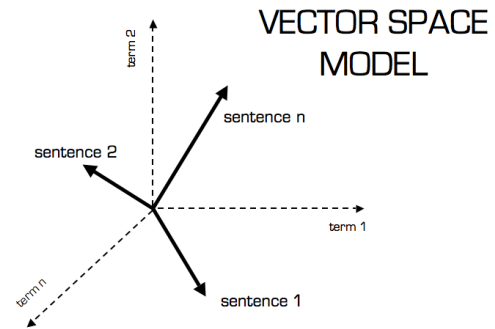
```
["przykładowy", "test", "analiza", "wyrazić"]
```

W procesie wyszukiwania będziemy musieli użyć dokładnie tego samego analizera do przygotowania zapytania.

## 3.2. Proces wyszukiwania



Po zaindeksowaniu wszystkich dokumentów możemy je teraz przeszukiwać. Proces wyszukiwania przechodzi po indeksie szukając najlepszego dopasowania do naszego zapytania (Query), przeanalizowanego przez nasz analizy. Do punktowania (Scoring) wykorzystywany jest model podobieństwa TF-IDF (TF-IDF Similarity) implementujący model przestrzeni wektorowej (Vector Space Model - VSM). W VSM im więcej razy dane słowo pojawia się w dokumencie w stosunku do frekwencji we wszystkich dokumentach tym wyżej punktowany jest.



Rysunek 3.4: VSM

### 3.2.1. TF-IDF Similarity

Podobieństwo TF-IDF (term frequency – inverse document frequency) [3] jest metodą obliczania wagi słów w oparciu o liczbę ich wystąpień. Mierzone jest w modelu VSM, w którym zarówno dokumenty jak i zapytania reprezentowane są jako wektory ważone w przestrzeni wielowymiarowej.

Założmy, że  $t$  oznacza term oraz  $d$  dokument. Wartość podobieństwa TF-IDF termu  $t$  dla dokumentem  $d$  oblicza się ze wzoru:

$$tfidf(t) = tf(t, d) \times idf(t)$$

gdzie  $tf(t, d)$  nazywamy frekwencją termu (term frequency) i wyrażamy wzorem

$$tf(t, d) = \frac{n_{t,d}}{\sum_k n_{k,d}}$$

gdzie  $n_{t,d}$  jest liczbą wystąpień termu  $t$  w dokumencie  $d$ .

$idf(t)$  nazywamy odwróconą frekwencją dokumentów (inverse document frequency) wyrażona wzorem

$$idf(t) = \log \frac{|D|}{1 + |\{d: t \in d\}|}$$

gdzie  $|D|$  jest liczbą dokumentów a  $|\{d: t \in d\}|$  jest liczbą dokumentów zawierających dany term. Dodając jedynkę tj  $1 + |\{d: t \in d\}|$  zabezpieczamy się przed dzieleniem przez 0.

Dla lepszej jakości wyszukiwań i bezpieczeństwa, dobra wyszukiwarka nie polega tylko na podobieństwie. Bierze również pod uwagę czy strona znajduje się w cenionej witrynie, czy witryna jest niskiej jakości albo nawet spamuje. Takie witryny powinny być odnotowane na specjalnej liście spammerskiej a wyszukiwarka nie powinna rekomendować żadnych stron z danej witryny.

Za to punktację wysokiej jakości stron może podnieść tak zwany PageRank.

### 3.2.2. PageRank

Algorytm opracowany przez założyciela Google - Larry'ego Page'a i Sergey'a Brin'a [4]

## 3.3. Dodatkowe funkcje składni zapytania



## Rozdział 4

# Przedstawienie problemu i sposób jego rozwiązania

- 
- Przede wszystkim przedstawienie tego problemu, który będzie opisany w pracy
  - problem przedstawiony w pracy może być jednym z podproblemów ogólnego problemu biznesowego
  - opis metod lub algorytmów np w postaci pseudokodu lub schematów
  - uzasadnienie użytych metod
  - testy kilku parametrów, przygotowanie danych itd.
-



## Rozdział 5

# Funkcjonalność zaimplementowanego systemu

- 
- opis jakie funkcjonalności będzie miał system
  - w jaki sposób dana funkcjonalność przyczyni się do rozwiązania problemu
-



## Rozdział 6

# Implementacja oraz użyte technologie

- 
- dokładnie jak zostało to wszystko implementowane
  - opis architektury
  - jaka platforma, diagram klas, jaka funkcjonalność jest w jakiej klasie/module/package
  - opis głównych metod (nie opisujemy wszystkich metod)
  - parametry jakie ma aplikacja, opis instalacji
- 

### 6.1. Java

#### 6.1.1. Apache Lucene

### 6.2. Spring Boot

### 6.3. Thymeleaf

### 6.4. HTML

### 6.5. Struktura i działanie klas programu

#### 6.5.1. Klasy silnika wyszukiwania

#### 6.5.2. Klasy kontrolera





## Rozdział 7

# Przypadki użycia

- 
- co dokładnie trzeba zrobić, aby zrealizować daną funkcjonalność
  - zrzuty ekranu
-



## Rozdział 8

# Podsumowanie

- 
- ok 1-2 str
  - cel pracy udało się osiągnąć
  - w jaki sposób udało się osiągnąć cel z rozdziału 1 (w których rozdziałach jest to opisane)
  - napotkane problemy
  - w jaki sposób można aplikację/system udoskonalić

---

[1]



# Bibliografia

- [1] Google Power Searching. *British Columbia Institute of Technology*, 2007.
- [2] Shraddha T. Shela Deepali D. Rane. Review of Lucene Indexing Algorithm on Public Cloud. *International Journal for Research in Applied Science And Engineering Technology*, 2017.
- [3] Apache Software Foundation. Class TFIDFSimilarity. [https://lucene.apache.org/core/4\\_0\\_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html](https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html), 2000-2012.
- [4] Google. How Search Works. <https://www.google.com/search/howsearchworks/crawling-indexing/>, 2010.
- [5] Google. Google Search Education Evangelism. <https://sites.google.com/site/gwebsearcheducation/advanced-operators>, 2011.
- [6] Addagada Sridevi. Indexing and Searching Document Collections using Lucene. *University of New Orleans Theses and Dissertations. 1070*, 2007.
- [7] 24 Godziny Sp. z o.o. szukacz.pl. [http://szukacz.pl/wiecej\\_o\\_szukaczu.html](http://szukacz.pl/wiecej_o_szukaczu.html), 2000-2018.