

Zapytania do bazy danych artykułów medycznych

Słowa kluczowe: przetwarzanie tekstu, zapytania do serwera/pobieranie odpowiedzi, wspomaganie decyzji, Apache Lucene, słownik MeSH

Opis: Napisanie aplikacji (internetowej bądź desktopowej), która będzie nakładką na bazę artykułów medycznych/wyszukiwarkę [PubMed](#). Wyszukiwarka PubMed pozwala wyszukać artykuły medyczne, które najbardziej odpowiadają zapytaniu. Jednak sam proces sortowania artykułów jest nieznany i słabo konfigurowalny.

Przykład wyszukiwarki realizującej funkcjonalność podobną do opisaną w projekcie: [GoPubMed](#).

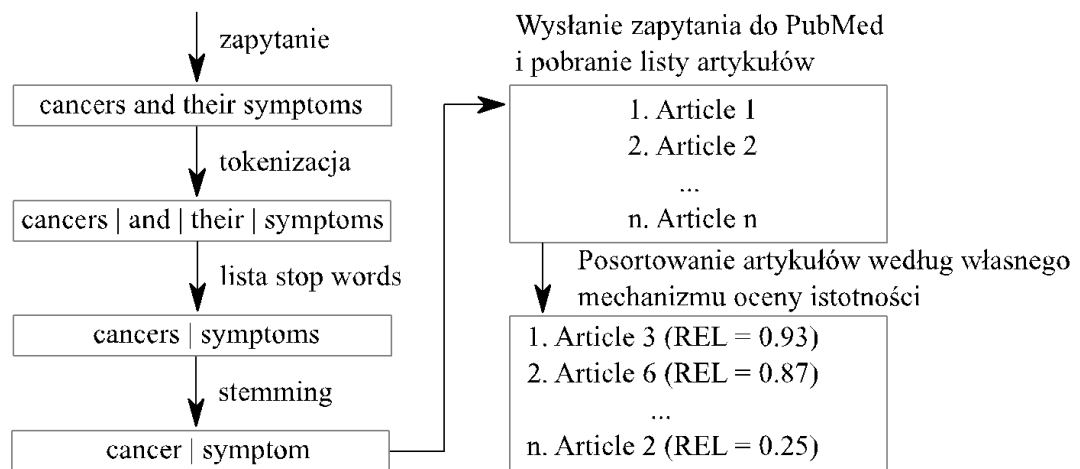
Wymagania (obowiązkowe):

- Aplikacja może być wykonana w Javie lub .NET. Może być desktopowa bądź internetowa.
- Aplikacja powinna przetworzyć oryginalne zapytanie w następujący sposób:
 - Tokenizacja: podzielenie zapytania na tokeny.
 - Usunięcie słów nieznaczących (lista stop words).
 - Zastosowanie stemmingu (obcięcie końcówek, próba uzyskania rdzenia).
 - Wyodrębnienie termów.
- Do przetworzenia zapytania należy wykorzystać Apache Lucene lub Lucene.NET.
- Bazę artykułów PubMed należy pobrać i odtworzyć ze zrzutów w formacie [XML](#) i ich [przyrostowych aktualizacji](#) (proszę załadować do systemu minimum dwa zrzuty i umożliwić późniejsze załadowanie kolejnych).
- Proces sortowania artykułów należy zamodelować samemu (bez wykorzystania zewnętrznych bibliotek). Obowiązkowo należy wykorzystać i samodzielnie zaimplementować miary:
 - Term Frequency (TF): jak często dany term pojawia się w tekście
 - Term Frequency-Inverse Document Frequency (TF-IDF): Podobnie jak TF, jednak obniżona zostaje waga (przydatność) tych termów, które znajdują się w wielu dokumentach.

Do porównania zapytania i dokumentu należy wykorzystać miarę cosinusową.

- Finalnie aplikacja powinna przedstawić listę artykułów posortowanych wg adekwatności dla danego zapytania. Taka lista powinna zawierać najistotniejsze pobrane elementy (tytuł, słowa kluczowe) oraz obliczoną ocenę istotności (preferowana prezentacja na znormalizowanej skali, np.: 0-1) względem zapytania.

- Ogólny schemat procesu:



Wymagania na 4.0:

- Ocenie będzie podlegać estetyka wykonania, przejrzystość/czytelność interfejsu itp.
- Należy zamodelować przynajmniej 2 nowe miary (lub ogólnie – modyfikacje istniejących) opisujące podobieństwo (istotność) zapytania do artykułu. W tym wypadku trzeba przemyśleć co może sprawiać, że dokument faktycznie może być bardziej znaczący (np. wysoka waga słów kluczowych itp.). Interfejs musi mieć możliwość wyboru metody sortowania.

Wymagania na 5.0:

- Zaimplementowane miary mogą okazać się niesatysfakcjonujące w oczach potencjalnego decydenta. Należy zaimplementować przynajmniej dwie (proste) opcje wspomaganie decyzji z trzech niżej wymienionych (lub zaproponować własne):
 - Rozszerzanie zapytania. W tym wypadku należy wykorzystać hierarchiczny słownik pojęć medycznych MeSH i z jego pomocą sugerować hasła rozszerzające zapytanie (w tym pojęcia ogólniejsze i bardziej szczegółowe od zapisanych w deskrypcorze artykułu w PubMed)
 - Modyfikacja mechanizmu porównania zapytania i artykułu. Przykładowo: decydent wiedząc, że dany term ma dla niego szczególne znaczenie, mógłby mieć możliwość podniesienia jego wagi.
 - Decydent mógłby mieć możliwość oceny faktycznej istotności/braku istotności uzyskanych artykułów (np. nadając odpowiednie flagi). Taka dodatkowa wiedza mogłaby zostać użyta do zmodyfikowania oryginalnego zapytania – należy tu wykorzystać metodę Rocchio.
- Należy przedstawić prostą statystyką porównującą oryginalny porządek artykułów (wynikający z budowy rankingu na podstawie TF-IDF i/lub narzucony przez PubMed) z porządkiem uzyskanym przy użyciu zaimplementowanego modelu. Należy wtedy odpowiedzieć na pytanie, czy istnieje zysk? Czy dodatkowy feedback od decydenta znacząco zmienił porządek artykułów? Warto tutaj pokazać przykładowy scenariusz użytkownika aplikacji. Tzn. decydent chciałby znaleźć artykuł na dany temat, jednak po wpisaniu słów kluczowych, najistotniejszy znalazł się na 10 miejscu. Czy udało się podbić jego pozycję,

modyfikując parametry metody?

Linki

- PubMed: <https://www.ncbi.nlm.nih.gov/pubmed>
- MeSH: <https://www.ncbi.nlm.nih.gov/mesh>
- TF: https://pl.wikipedia.org/wiki/Term_frequency
- IDF: <https://pl.wikipedia.org/wiki/TFIDF>
- Cosinus kąta pomiędzy wektorami:
https://pl.wikipedia.org/wiki/Iloczyn_skalarny#Interpretacja_geometryczna
- Apache Lucene: <http://lucene.apache.org/core/>
- Apache Lucene .NET <https://lucenenet.apache.org/>
- Lista stop words: https://en.wikipedia.org/wiki/Stop_words
- Stemmer Portera: <https://tartarus.org/martin/PorterStemmer/>
- Metoda Rocchio: https://en.wikipedia.org/wiki/Rocchio_algorithm