STUDIA INFORMATICA 2009
Volume 30 Number 2A (83)

Jakub CIEŚLEWICZ, Adam PELIKANT
Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych

REPREZENTACJA I WYSZUKIWANIE DOKUMENTÓW TEKSTOWYCH W BAZACH DANYCH

Streszczenie. Artykuł porusza problem wyszukiwania dokumentów na podstawie ich rzeczywistej treści. Opisuje model odwzorowania ciągłych tekstów w postaci wektorowej, mechanizmy nadawania wag poszczególnym cechom dokumentów oraz prezentuje algorytm porównywania oparty na mierze kosinusowej między reprezentacjami wektorowymi dokumentów a reprezentacją zapytania.

Słowa kluczowe: inteligencja obliczeniowa, zgłębianie tekstu, zgłębianie danych, Oracle

TEXT DOCUMENTS' REPRESENTATION AND RETRIEVAL AT DATABASES SYSTEMS

Summary. The article propels the problem of retrieval documents due to their real content. It describes the model of continuous texts representation as vectors and presents the mechanisms of weights assignment to the individual document features as well as the algorithm of comparing leaning on cosines' measure between vector representations of documents and query.

Keywords: business intelligence, text mining, data mining, Oracle

1. Wstęp

W systemach baz danych przechowywane są najczęściej dane strukturalne o ustalonej metodzie reprezentacji numerycznej, jednakże zapotrzebowanie komercyjne wymusza także przechowywanie danych o słabo zdeterminowanej reprezentacji. Takimi danymi są dane multimedialne, ale również dane tekstowe rozumiane jako obiekty typu LOB (*Large Object Binary*). Zawierają one bardzo dużą liczbę informacji w porównaniu z danymi prostymi, jed-

nak systemy BD nie oferują dostatecznie dużej liczby narzędzi do ich przetwarzania, ograniczając się do różnych form wyrażeń regularnych. Proces wnikliwej i ściśle określonej interpretacji tekstu wymaga stosowania zaawansowanych narzędzi, pozwalających na odwzorowanie treści na formę możliwą do przetwarzania numerycznego.

Odpowiedzią na takie zapotrzebowanie było stworzenie dziedziny określonej mianem *text mining*, która nie ma odwzorowania w komercyjnych SZBD. Stosowanie nowej technologii umożliwia wybór metody analizy, dzięki której uzyskane wyniki będą najbliższe oczekiwaniom. *Text mining* pozwala np. automatyzować procesy klasyfikacji dokumentów ze względu na tematykę lub formę, wyszukiwać dokumenty zawierające wiedzę istotną dla użytkownika.

Celem artykułu jest rozpatrzenie zagadnienia zgłębiania tekstu jako metody umożliwiającej sprawne przeszukiwanie zbiorów tekstowych. Przedstawiony zostanie algorytm wyszukiwania tekstu na podstawie wektorowej reprezentacji dokumentów, który został zaimplementowany na platformie Oracle. Ponadto, omówione zostaną zagadnienia związane z ogólnymi zasadami wstępnego przetwarzania dokumentów, w skład których wchodzą: rozpoznanie typu dokumentu, określenie wersji językowej, analiza leksykalna wraz z uwzględnieniem redukcji form fleksyjnych, nadawanie wag wyekstrahowanym cechom. Opisane zostaną kwestie związane z przygotowaniem wektorowej reprezentacji dokumentu w przestrzeni metrycznej oraz odzwierciedleniem zbioru tekstów za pomocą macierzy wektorów dokumentów. W końcu omówiony zostanie zaimplementowany algorytm wyszukiwania dokumentów, stosujący metrykę kosinusową, na podstawie ich wektorowej reprezentacji.

Opracowana w środowisku Oracle aplikacja realizuje zaproponowany algorytm reprezentacji i przetwarzania tekstu i ma na celu wyszukiwanie dokumentów o największym podobieństwie w sensie miary kosinusowej do frazy wejściowej. Założono przy tym, że: zbiór danych tekstowych składowany jest tylko i wyłącznie wewnątrz bazy danych i nie umożliwia dołączenia treści przechowywanych zewnętrznie w systemie operacyjnym. Dokumenty zebrane w bazie mają postać tekstową. Tekst poddawany jest podstawowej analizie, polegającej na ekstrakcji pojedynczych, relewantnych słów w poprawnej formie gramatycznej. Oznacza to, że treść nie jest przetwarzana pod kątem uproszczenia złożoności fleksji. Zastosowany mechanizm redukcji przestrzeni cech opiera się na eliminacji słów niedozwolonych zebranych na tak zwanej stop liście.

2. Analiza dokumentów

Dokładność wyników uzyskanych w procesie eksploracji tekstu ściśle zależy od uwzględnionych w programie algorytmów przeprowadzających wstępną analizę tekstu. Wyeliminowanie nierelewantnych słów, uproszczenie fleksji i konstrukcji gramatycznych, przy jedno-

cześnie najmniejszym stopniu utraty istotnych danych zapisanych w dokumencie, gwarantuje uzyskanie najlepszych trafień w wyszukiwaniu.

2.1. Identyfikacja typu dokumentu

Powszechnie tworzone dokumenty przybierają formę ciągłego tekstu, w których zebrane informacje zaprezentowane są w poprawnej formie gramatycznej. Coraz większe uznanie zyskują różne formaty, które oprócz treści zawierają metadane, będące dodatkowym opisem treści bądź jawnym oznaczeniem sekcji dokumentu. Takimi formatami są np. pliki typu HTML czy hierarchiczne pliki XML. Same znaczniki nie są istotne dla przetwarzania dokumentu. Jednak zastosowanie algorytmu, wyznaczającego indywidualnie wartość wagi dla treści opisanej danym znacznikiem, pozwola na poprawę trafności wyszukiwania. Algorytmy stosujące zasadę ważności, w zależności od miejsca występowania, w czystym dokumencie tekstowym są bardziej skomplikowane niż te dedykowane dla plików strukturalnych.

2.2. Wstępne przetwarzanie

Proces wstępnego przetwarzania najczęściej stosowany jest już w czasie ładowania danych z zewnętrznych nośników do wewnętrznego magazynu – bazy danych. Etap wczytywania to głównie ustalenie granic między wyrazami, *n*-gramami słów bądź między zdaniami. Granice wyznaczane są na podstawie zbioru separatorów, który powinien zostać osobno określony dla każdej grupy analizowanych dokumentów. W przypadku większości języków naturalnych określenie takiego zbioru nie jest skomplikowane. Usunięcie z analizy kropek czy przecinków nie ma większego wpływu, ponieważ niosą niewiele informacji. W przypadku języków sztucznych problem się komplikuje, gdyż tracone są informacje na temat wielokrotnego wystąpienia ciągu znaków białych, przestankowych, końca linii, wiersza, akapitu etc.

2.3. Ujednolicenie wielkości liter

Uproszczenie przetwarzania i konieczność zwolnienia użytkownika z obowiązku stosowania określonego schematu konstrukcji zapytania tekstowego wymaga ujednolicenia wielkości liter. Przed dokonaniem transformacji należy zastanowić się, czy informacje wynikające np. z obecności wielkiej litery na początku wyrazu nie są zbyt cenne i mogą zostać utracone. W przypadku języka polskiego ciąg wyrazów zaczynających się wielką literą określa nazwy własne. Rozbicie takiego *n*-gramu słownego i przetwarzanie pojedynczych wyrazów niekorzystnie wpływa na trafność wyszukiwania.

2.4. Rozpoznanie języka dokumentu

Poprawne rozpoznanie języka dokumentu jest istotne przy wyborze algorytmu odpowiedzialnego za zmniejszenie różnorodności wyrazów przez sprowadzenie ich, np. do formy podstawowej. Takie informacje mogą być pobrane z metadanych, a w przypadku dokumentów nieposiadających metadanych, język określić można przez wyznaczanie grupy wyrazów niedozwolonych (zebranych na tzw. stop-listach) najliczniej reprezentowanej w dokumencie. Inne podejście to zastosowanie metody statystycznej polegającej na wyznaczeniu, charakterystycznego dla każdego języka, rozkładu występowania *n*-gramów literowych, czyli *n*-literowych ciągów w wyrazach.

2.5. Stop-lista

Stop-lista jest zbiorem słów, których obecność podyktowana jest zasadami gramatycznymi i koniecznością zachowania logicznej formy przekazu. W czasie komputerowej analizy tekstów kształt wypowiedzi zostaje całkowicie zniszczony i z tego powodu zasadne jest usuwanie tego typu słów. Dodatkowo, wyrazy te stanowią duży procent wszystkich słów w tekście i przez to przodują w statystykach opartych na zliczaniu ilości wystąpień. Po drugie, obecność prawie w każdym dokumencie powoduje, że ich siła dyskryminacyjna jest zerowa i uwzględnienie w analizie może wprowadzić przekłamania. Ponadto, zmniejszenie zbioru analizy wpływa na redukcję wymiarów przestrzeni, co pozwala zaoszczędzić pamięć oraz skrócić czas działania algorytmów.

2.6. Redukcja form fleksyjnych

Odmiana czasowników przez osoby, rzeczowników przez przypadki oraz zachowanie odpowiedniej formy czasu to, z punktu widzenia analizy komputerowej, nadmiarowe informacje. Aby ułatwić porównywanie dokumentów tekstowych, w których te same słowa występują w różnych formach fleksyjnych, należy sprowadzić je do formy podstawowej, tzw. lemmy (najprostsza z możliwych form, kanoniczna postać leksemu), bądź leksemu (znaczenie leksykalne oraz zespół wszystkich funkcji gramatycznych, jakie dany wyraz może spełniać). Jako najprostszą postać wyrazów przyjmuje się bezokolicznik dla czasowników, mianownik liczby pojedynczej dla rzeczowników. Redukcja form fleksyjnych pozwala wyeliminować niespójności w listach frekwencyjnych, czyli ilościowego podsumowania pojawienia się wyrazu w tekście, wynikających z rozdziału wystąpień danego słowa w różnych formach gramatycznych.

Uproszczenie form wyrazów przeprowadza się za pomocą algorytmów lematyzacji bądź steaming'u. Lematyzacja jest procesem uwzględniającym kontekst słowa i budowę grama-

tyczną zdania, w którym ono występuje. Efektem działania algorytmu lematyzacji jest podstawowa forma wyrazu. Steaming polega na wyeliminowaniu przyrostków i przedrostków. W wyniku jego zastosowania otrzymuje się rdzenie wyrazów.

Zarówno lematyzację, jak i steaming można przeprowadzić opierając się na dwóch głównych podejściach. Pierwsze, to postać słownikowa zawierająca zbiór różnych form gramatycznych poszczególnych słów wraz z odpowiadającą im formą podstawową – lematem lub rdzeniem. Drugi algorytm opiera się na zasadach i regułach gramatyk językowych. Stopień trudności oraz poprawność działania takiego algorytmu zależą od poziomu skomplikowania zasad językowych.

Eliminacja przedrostków i przyrostków oraz lematyzacja komplikują się z powodu homografii wynikającej z obecności słów posiadających taką samą formę graficzną, ale mających zupełnie inne znaczenie i funkcjonalność. Problem wikła się jeszcze bardziej, gdy pojawia się homonimia, czyli występowanie wyrazów stanowiących tę samą część mowy, ale mających wiele znaczeń w zależności od kontekstu. Skala problemu jeszcze bardziej rośnie, jeśli uwzględniony zostanie język potoczny, żargon, gwara.

3. Wektorowa reprezentacja dokumentu

Przetwarzanie danych o słabo zdeterminowanej strukturze wymaga użycia reprezentacji pozwalającej na znaczną poprawę wydajności operacji numerycznych, przy jednoczesnym uwzględnieniu wszystkich istotnych informacji zawartych w treści. Najczęściej spotykaną formą odwzorowania jest postać wektora liczb rzeczywistych. Podejście to, oprócz sprawnego wyszukiwania dokumentów, umożliwia również przeprowadzanie analiz opartych na metodach *data mining'u*, takich jak klasyfikacja czy grupowanie. Wektor jest ściśle powiązany ze słowami znajdującymi się w dokumencie, których częstość (ilość) określa jego współrzędne.

3.1. Metody nadawania wag

Statystyczne analizy tekstów dowodzą, że pewne wyrazy, które nie kwalifikują się na stop-listę, występują w znaczącej ilości w tekstach o odmiennej tematyce. Stosując wektorową reprezentację tekstu, wpływ takich współrzędnych można zminimalizować przez dobór odpowiedniego mechanizmu nadawania wag. Źle dobrany algorytm może wprowadzić pewne szumy w wyznaczaniu odległości między wektorami w przestrzeni, w wyniku czego odległość między podobnymi dokumentami może zostać zwiększona, a między różnymi zmniejszona.

3.1.1. Reprezentacja boolowska

Podejście oparte na boolowskim odwzorowaniu treści w wektorze, dla cechy skojarzonej ze współrzędną wektora oraz występującej w dokumencie, ustala wagę równą 1. Gdy dana cecha czy wyraz nie pojawia się w treści, współrzędna przyjmuje wartość 0. Zaletą omówionego podejścia jest prostota realizacji oraz fakt, że wyznaczenie wartości niektórych metryk dla wektorów zero-jedynkowych może być przeprowadzone w sposób wymagający mniejszej mocy obliczeniowej.

Przyjęcie tak banalnego modelu nadawania wag ma swoje konsekwencje. Na podstawie oznaczenia, czy dane słowo występuje, czy też nie występuje w dokumencie, trudno jest wnioskować o jego tematyce. Wystąpienie pewnego słowa w tekście może mylnie determinować jego tematykę, ponieważ przypisuje się identyczne wagi pojedynczemu i wielokrotnemu wystąpieniu słowa, niezależnie od długości dokumentu. Model taki nie uwzględnia również informacji, w ilu dokumentach dane słowo występuje.

3.1.2. Reprezentacja ilościowa

Reprezentacja ilościowa ustala wagi na podstawie częstości słowa (*Term Frequency* – TF). Częstość bezwzględna określa ilość wystąpień słowa w dokumencie, natomiast częstość względna to stosunek ilości wystąpień danego wyrazu w tekście do liczby wszystkich wyrazów uwzględnianych w analizie. Nadawanie wag metodą częstości bezwzględnej powoduje, iż dokumenty krótkie i długie dotyczące tego samego tematu reprezentowane są przez zupełnie inne wektory. W takim wypadku konieczne jest przeprowadzenie normalizacji wektorów, tak aby ich długość była wartością jednostkową. Innym rozwiązaniem może być zastosowanie metryki, która nie jest czuła na bezwzględne wartości współrzędnych. Problem różnic w reprezentacji dokumentów w zależności od ich długości nie występuje, gdy stosowana jest częstość względna.

Największą korzyścią wynikającą ze stosowania wag TF jest łatwość przetwarzania. Wyznaczenie częstości cech wymaga przeanalizowania tylko jednego dokumentu. Nie są wykorzystywane żadne statystyczne informacje dotyczące innych dokumentów. Prostota metody sprawia, że jest ona szybka i wydajna, co wpływa na powszechne jej stosowanie.

3.1.3. Reprezentacja TFIDF

Algorytm TFIDF przy określaniu ważności cechy uwzględnia odwrotną częstość dokumentu. Człon TF, podobnie jak wyżej oznacza częstość, a IDF odwrotną częstość dokumentu (*inverse document frequency*). Wartości te definiowane są następująco:

$$TF(i) = \frac{N_i}{N} \tag{1}$$

dla wartości względnej bądź

$$TF(i) = N_i \tag{2}$$

dla wartości bezwzględnej, gdzie N określa łączną liczbę cech danego rodzaju (słów, n-gramów), N_i informuje o liczbie wystąpień konkretnej cechy w danym dokumencie.

$$IDF(i) = \ln \frac{D}{D_i} \tag{3}$$

gdzie D to liczba wszystkich dokumentów w zbiorze, a D_i określa liczbę dokumentów zawierających cechę i.

Po uwzględnieniu powyższychwzorów otrzymuje się

$$TFIDF(i) = TF(i) \cdot IDF(i) = \frac{N_i}{N} \cdot \ln \frac{D}{D_i} \tag{4}$$

Stosowanie wag TFIDF eliminuje efekt przeszacowania wartości słowa. Oprócz częstości cechy w dokumencie, pod uwagę brana jest również obecność tej cechy w innych dokumentach. Z własności logarytmu naturalnego wynika, że siła dyskryminacyjna dla wyrazu występującego w niewielkiej liczbie dokumentów jest większa, niż dla słowa występującego w znacznej części zbioru tekstów. Współczynnik TFIDF spełnia rolę podobną do entropii w zagadnieniach zgłębiania danych.

Wyznaczanie wag TFIDF wymaga nie tylko analizy jednego dokumentu, ale również korzystania z charakterystyki całego zbioru. Do obliczenia poprawnej wartości konieczne jest przechowywanie informacji o ilości elementów w zbiorze oraz o liczbie dokumentów, w których dana cecha występuje. Należy pamiętać o konieczności wyznaczania nowych wartości wag TFIDF dla wszystkich cech każdego dokumentu zarówno w chwili zmiany zawartości dokumentu, jak i zmiany zbioru przez dodanie bądź usunięcie dokumentu.

3.1.4. Macierzowe odwzorowanie zbioru dokumentów

Wektory będące odwzorowaniem dokumentu są tak skonstruowane, aby opisywały tylko cechy, które w nim występują. W tym momencie należy spojrzeć na zgromadzone i przetworzone zasoby jako całość. Skoro pojedynczy dokument reprezentowany jest w postaci wektora, to naturalne wydaje się być przedstawienie zbioru N dokumentów w postaci macierzy nazywanej TFM (*term frequency matrix*), której elementy $TFM[d_i, n_i]$ opisują wagę cechy n_i w dokumencie d_i, reprezentowanym w postaci wyżej omówionych wektorów. Każdy dokument odwzorowany jest przez wektory o różnej liczbie współrzędnych, których wartość jest równa wadze określonej dla konkretnej cechy dokumentu. Jeśli pewna cecha została wprowadzona do macierzy z powodu występowania w dokumencie d_i, a nie wstępuje w dokumencie d_{i+1}, to dla tej cechy w dokumencie d_{i+1} ustala się wagę równą 0. W związku z tym macierz TMF jest macierzą rzadką.

Dla dużych zbiorów danych wymiary macierzy wektorów dokumentów mogą być bardzo duże. Pierwszy wynikał będzie z wielkości zbioru dokumentów, drugi będzie iloczynem liczby dokumentów i ilości wyodrębnionych cech zbioru, wziętą bez powtórzeń.

3.1.5. Redukcja wymiaru przestrzeni cech

Problem wymiarowości macierzy wektorów dokumentów silnie oddziałuje na wydajność i efektywność przetwarzania. Łatwo zauważyć, że na redukcję wymiarowości wpływ mają opisane powyżej etapy analizy dokumentu tekstowego. Eliminacja wyrazów ze stop-listy, lematyzacja, steaming itp. w wielkim stopniu przyczyniają się do ograniczenia rozmiaru wektorów, co jest jednoznaczne z ograniczeniem wielkości macierzy.

Wymiarowość dodatkowo możne zostać ograniczona przez minimalne wartości dla wagi bądź liczności cechy w dokumencie, które będą uwzględniane w analizie. Stosując tę formę redukcji wymiaru, należy mieć na uwadze swoiste wady. Po pierwsze, ustalenie minimalnej wartości wagi lub liczności wpływa na utratę części informacji w odwzorowaniu wektorowym, po drugie, należy rozważyć przypadek pojawienia się dokumentu, w którym wartość liczbowa opisująca cechę, do tej pory eliminowaną z całego zbioru, jest większa od nałożonego ograniczenia. Możliwe jest również zastosowanie innych metod redukcji wymiarów, takich jak ukryte indeksowanie semantyczne (*latent semantic indexing* – LSI) bądź PCA (*principal component analysis*).

3.2. Analizowanie zasobów – wyszukiwanie dokumentów

Odwzorowanie pełnego zbioru zasobów w postaci macierzy wektorów dokumentów umożliwia przeprowadzanie analiz, których działanie pozwoli np. na poznanie asocjacji między elementami lub wyznaczyć zbiory treści podobnych tematycznie.

3.2.1. Przestrzeń metryczna

Reprezentacja wektorowa dokumentów umożliwia zdefiniowanie miary odległości pomiędzy dokumentami a zapytaniem użytkownika. Dokumenty o podobnej tematyce powinny charakteryzować się podobną częstością występowania tych samych słów kluczowych. Każdy dokument może być reprezentowany jako punkt w n-wymiarowej przestrzeni. Do wyznaczenia odległości pomiędzy dowolnie wybranymi wektorami dokumentów oraz pomiędzy wektorem dokumentu a wektorem zapytania użytkownika można stosować miary służące do wyznaczania odległości w przestrzeni euklidesowej. Miara odległości powinna spełniać aksjomaty metryki.

Definicja przestrzeni metrycznej

Przestrzenią metryczną (X,d) nazywamy dowolny zbiór X wraz z funkcją odległości (metryką) $d : X \times X \to R^+$ spełniającą następujące warunki:

- odległość między różnymi punktami jest niezerowa

$$d(x,y) = 0 \Leftrightarrow x = y \tag{5}$$

- symetryczność

$$d(x,y) = d(y,x) \tag{6}$$

- nierówność trójkąta

$$d(x,y) \le d(x,z) + d(z,y) \tag{7}$$

Wektor będący reprezentacją dokumentu, w odniesieniu do przestrzeni euklidesowej, można zdefiniować w następujący sposób: jeśli dokument można przedstawić w postaci *k*-wymiarowego wektora liczbowego, wówczas można traktować go jako punkt w *k*-wymiarowej przestrzeni euklidesowej. Za miarę odległości między takimi punktami można przyjąć dowolną z miar stosowanych dla tej przestrzeni.

3.2.2. Miara kosinusowa

Miara kosinusowa definiuje podobieństwo lub inaczej odległość $d(x, y)$ dokumentów x i y, jako znormalizowany iloczyn skalarny wektorów reprezentujących x i y, tj. jako iloczyn skalarny obu wektorów podzielony przez iloczyn ich długości:

$$d(x,y) = \frac{x \circ y}{|x| \cdot |y|} \tag{8}$$

gdzie iloczyn skalarny dwóch wektorów, rozważanych w przestrzeni euklidesowej, zdefiniowany jest jako suma iloczynu odpowiednich współrzędnych

$$x \circ y = \sum_{i=1}^{n} x_i y_i \tag{9}$$

natomiast iloczyn długości opisuje wzór

$$|x| \cdot |y| = \sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2} \tag{10}$$

Zauważyć można, że tak zdefiniowana miara nie spełnia pierwszego aksjomatu metryki. Należy wprowadzić modyfikację, tak aby warunek o zerowej odległości miedzy takimi samymi wektorami był spełniony:

$$d(x,y) = 1 - \frac{x \circ y}{|x| \cdot |y|} \tag{11}$$

Miarę kosinusową powiązać można również z kosinusem kąta między wektorami x i y

$$\cos(\angle(x,y)) = \frac{x \circ y}{|x| \cdot |y|} \tag{12}$$

3.2.3. Reprezentacja zapytania użytkownika

W procesie wyszukiwania udział biorą, z jednej strony, zbiór zasobów odwzorowany w macierzy reprezentacji wektorowej dokumentów, z drugiej, zapytanie użytkownika w postaci ciągu wyrazów, będącym kryterium wyszukiwania. Wydobycie ze zbioru dokumentów istotnych treści wiąże się z określeniem odległości każdego z wektorów zbioru zasobów od wektora zapytania. Przed wyznaczeniem miar, konieczne jest odwzorowanie wymagań użytkownika w postaci wektora. Słowom występującym w zapytaniu i wektorze cech opisującym zbiór nadaje się wagi na podstawie metody boolowskiej. Takie podejście pozwoli na określenie kierunku wektora zapytania i jest wystarczające do wyznaczenia odległości między wektorem kryterium wyszukiwania a wektorowym odwzorowaniem dokumentu.

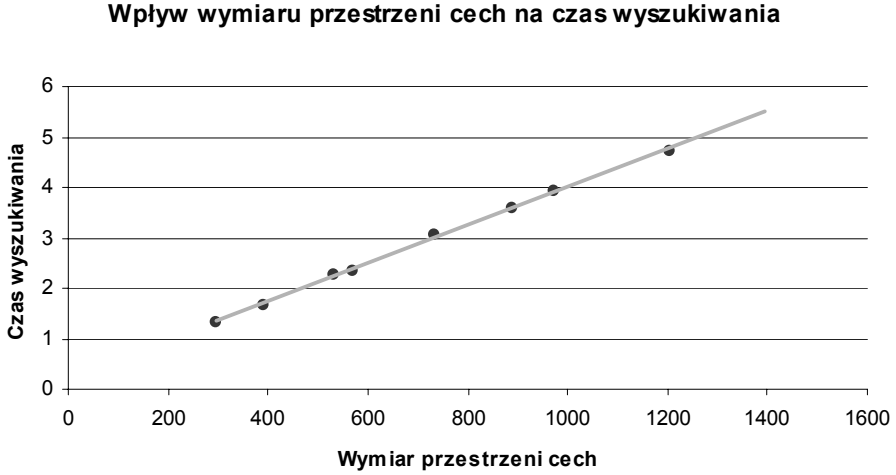
3.2.4. Porównywanie wektorów

Analiza wartości kosinusa kąta pomiędzy reprezentacjami wektorowymi zapytania i dokumentu pozwala na wyznaczenie ich podobieństwa. Im większa wartość kosinusa, tym większe prawdopodobieństwo, że dokument dotyczy tematyki określonej w zapytaniu. Prezentowany zbiór uzyskanych wyników można ograniczyć przez określenie minimalnej wartości kosinusa. Wszystkie dokumenty, będące w przestrzeni wektorowej dalej niż wyznaczone minimum, traktowane są jak dokumenty o odmiennej tematyce.

4. Ocena wydajności

Ocena przydatności algorytmu określona została na podstawie dwóch zależności. Pierwsza określa wpływ liczności zbioru na czas wyszukiwania, przy zachowaniu stałego wymiaru przestrzeni cech; druga prezentuje wpływ wymiaru przestrzeni cech na czas wyszukiwania przy zachowaniu stałej liczby elementów w zbiorze. Charakterystyki zostały wyznaczone na podstawie przeprowadzonych pomiarów czasu działania algorytmu dla określonego zbioru danych.

Rys. 1. Wykres zależności czasu wyszukiwania od liczności zbioru dokumentów
Fig. 1. Dependence the time of an information retrieval on the number of documents in collection



Rys. 2. Wykres zależności czasu wyszukiwania od wymiaru przestrzeni cech
Fig. 2. Dependence the time of an information retrieval on feature space dimension

Rząd algorytmu wynikający ze stopnia wielomianu aproksymującego dla zmiennej liczności zbioru wynosi w przybliżeniu 1,5. W przypadku zwiększania wymiaru przestrzeni cech algorytm wykazuje charakter liniowy. Na tej podstawie można przyjąć, że algorytm nadaje się do powszechnego zastosowania w praktyce.

5. Podsumowanie

Połączenie reprezentacji wektorowej dokumentów oraz odległości kosinusowej, jako miary podobieństwa dokumentów, powoduje, że zaprezentowany algorytm jest niezależny od metod wstępnej analizy tekstu czy schematu nadawania wag. Analiza tekstu opierać się może nie tylko na słownikach, ale również na programach stosujących zasady języka naturalnego, takie jak lematyzacja czy *steaming*. Ta sama implementacja algorytmu wyszukiwania poz-

wala na porównywanie zarówno reprezentacji wektorowej zbudowanej z pojedynczych słów jak i dowolnie skomplikowanych *n*-gramów. Zaletą algorytmu jest również to, że nie wymaga stosowania specjalnych indeksów, przez co nie jest konieczne ich ciągłe przebudowywanie w systemach, w których występuje częsta aktualizacja danych. Zadowalająca wydajność prezentowanego algorytmu pozwala na wykorzystanie go w systemach zgłębiania tekstu.

BIBLIOGRAFIA

1. Makhoul J., Kubala F., Schwartz R., Weischedel R.: Performance measures for information extraction. In Proceedings of DARPA Broadcast News Workshop, 1999, s. 249÷252.
2. Jing L., Huang H., Shi H.: Improved feature selection approach TFIDF in text mining. School of Computer & Information Technology, University Beijing, 2002.
3. Wang P., Hu J., Zeng H., Chen L., Chen Z.: Improving Text Classification by Using Encyclopedia Knowledge. Seventh IEEE International Conference on Data Mining.
4. Mazur P.: Text Segmentation in Polish. ISDA'05.
5. Morzy T., Morzy M., Leśniewska A.: Eksploracja tekstu I, Eksploracja tekstu II, wykłady poświęcone eksploracji danych. http://wazniak.mimuw.edu.pl/.

Recenzent: Dr inż. Paweł Kasprowski

Wpłynęło do Redakcji 20 stycznia 2009 r.

Abstract

This paper describes a text representation as a vector and searching technique using cosinus measure. It presents text preprocessing like detecting type and language of document, eliminating irrelevant words, steaming, lemmatization, feature selecting,. There are also information about building feature-vector document representation. The values of the vector elements for a document are calculated as a combination of statistic term frequency TF (1) and inverse document frequency IDF (3). The value of feature for document is calculated as the product (4). It is refer to as weight of word in document. The algorithm of comparing leaning on cosines' measure between vector representations of documents and query. The article raises problem of reduce future space dimensions. The algorithm performance was examine in special develop application. Figure 1. shows dependence the time of an

information retrieval on the number of documents in collection. Dependence the time of an information retrieval on feature space dimension is presented on figure 2.

Adresy

Jakub CIEŚLEWICZ: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego18/22, 90-924 Łódź, Polska, jakub.cieslewicz@p.lodz.pl.
Adam PELIKANT: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego18/22, 90-924 Łódź, Polska, apelikan@p.lodz.pl.