Eksploracja zasobów internetowych

Wykład 3

Wyszukiwanie dokumentów WWW bazujące na słowach kluczowych

mgr inż. Maciej Kopczyński

Białystok 2014

Wstęp

Wyszukiwanie dokumentów za pomocą słów kluczowych bazujące na regułach boolowskich jest proste i szybkie, jednak posiada dużą wadę.

Nie pozwala na sortowanie zwróconych wyników wyszukiwania pod względem istotności treści.

W jaki sposób poradzić sobie z tym problemem?

Poprzez definiowanie zapytań w sposób precyzyjny lub sortowanie zwróconych treści bazując na ilości wystąpień poszczególnych termów w korpusie dokumentu.

Wstęp

Najbardziej powszechne struktury danych do przechowywania treści pobranych ze stron WWW:

- macierz term-dokument typu boolowskiego,
- macierz term-dokument typu ilościowego,
- macierz term-dokument typu pozycyjnego.

Z wykorzystaniem tych struktur można zwracać wyniki bazujące na słowach kluczowych stosując reguły boolowskie.

Wstęp

Przy wyszukiwaniu danych zgodnych z zapytaniem złożonym ze słów kluczowych ważne jest sortowanie pod względem istotności treści zboru wynikowego.

Osiągnięcie tego celu wymaga zastosowania odpowiednich algorytmów oraz zmodyfikowanych struktur do przechowywania danych.

Powstałe struktury bazują na poznanych do tej pory metodach przechowywania pobranych danych.

Dane przykładowe

Strona internetowa *www.artsci.ccsu.edu*:

Dane przykładowe

Przykładowy zbiór danych ze strony *www.artsci.ccsu.edu*:

Document ID	Document name	Words	Terms
d_1	Anthropology	114	86
d_2	Art	153	105
d_3	Biology	123	91
d_4	Chemistry	87	58
d_5	Communication	124	88
d_6	Computer Science	101	77
d_7	Criminal Justice	85	60
d_8	Economics	107	76
d_9	English	116	80
d_{10}	Geography	95	68
d_{11}	History	108	78
d_{12}	Mathematics	89	66
d_{13}	Modern Languages	110	75
d_{14}	Music	137	91
d_{15}	Philosophy	85	54
d_{16}	Physics	130	100
d_{17}	Political Science	120	86
d_{18}	Psychology	96	60
d_{19}	Sociology	99	66
d_{20}	Theatre	116	80
Total number of words/terms		2195	1545
Number of different words/terms		744	671

Analiza danych

Pod względem możliwości analizy danych pobranych ze stron WWW najbardziej odpowiednia jest struktura typu pozycyjnego:

- względna prostota przechowywania danych,
- łatwość wyszukiwania informacji,
- możliwość zliczenia ilości wystąpień termów w dokumencie,
- możliwość wyszukiwania słów leżących w określonych odległościach względem siebie.

Macierz typu pozycyjnego

Przykładowa macierz term-dokument typu pozycyjnego:

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	[71]
d_2	0	0	0	0	[7]
d_3	0	[65,69]	0	[68]	0
d_4	0	0	0	[26]	[30,43]
d_5	0	0	0	0	0
d_6	0	0	[40,42]	[1,3,7,13,26,34]	[11,18,61]
d_7	0	0	0	0	[9,42]
d_8	0	0	0	0	[57]
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0
d_{11}	0	0	0	0	0
d_{12}	0	0	0	[17]	0
d_{13}	0	0	0	0	0
d_{14}	[42]	0	0	[41]	[71]
d_{15}	0	0	0	0	[37,38]
d_{16}	0	0	0	0	[81]
d_{17}	0	0	0	0	[68]
d_{18}	0	0	0	0	0
d_{19}	0	0	0	0	[51]
d_{20}	0	0	0	0	0

Model wektorowy

Model wektorowy jest bezpośrednio powiązany z macierzowymi strukturami danych i wynikach z ich innej interpretacji logicznej.

Struktury modelu wektorowego pozwalają klasyfikować zbiór wyników pod kątem istotności traktując dokumenty jako wektory wielowymiarowe.

Każdy z takich wektorów posiada ilość współrzędnych równą ilości termów we wszystkich zbiorach dokumentów.

Model wektorowy

Rodzaje powszechnie stosowanych modeli wektorowych w systemach typu *Information Retrieval*:

- boolowski,
- *Term-Frequency* (TF),
- *Inverse Document Frequency* (IDF),
- *Term Frequency – Inverse Document Frequency* (TFIDF).

Model wektorowy – oznaczenia

Oznaczenia wykorzystywane na kolejnych slajdach odnoszące się do struktur logicznych:

- $d_1, d_2, ..., d_n$ – dokumenty
- $t_1, t_2, ..., t_m$ – termy
- n_{ij} – ilość termów t_i w dokumencie d_j
- m – ilość wszystkich termów
- n – ilość wszystkich dokumentów

Model wektorowy – boolowski

Pojedynczy wiersz w boolowskim modelu wektorowym opisany jest jako:

$$\vec{d}_j = (d_j^1 d_j^2 \ldots d_j^n) \qquad d_i^j = \begin{cases} 0 & dla\ n_{ij} = 0 \\ 1 & dla\ n_{ij} > 0 \end{cases}$$

Dla zbioru termów *lab, laboratory, programming, computer* oraz *program* dla dokumentu d_6 wektor będzie zadany jako:

$$\vec{d}_6 = (0 \quad 0 \quad 1 \quad 1 \quad 1)$$

Macierz typu boolowskiego

Przykładowa macierz term-dokument typu boolowskiego:

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	1
d_2	0	0	0	0	1
d_3	0	1	0	1	0
d_4	0	0	0	1	1
d_5	0	0	0	0	0
d_6	0	0	1	1	1
d_7	0	0	0	0	1
d_8	0	0	0	0	1
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0
d_{11}	0	0	0	0	0
d_{12}	0	0	0	1	0
d_{13}	0	0	0	0	0
d_{14}	1	0	0	1	1
d_{15}	0	0	0	0	1
d_{16}	0	0	0	0	1
d_{17}	0	0	0	0	1
d_{18}	0	0	0	0	0
d_{19}	0	0	0	0	1
d_{20}	0	0	0	0	0

Model wektorowy – TF

Pojedynczy wiersz w modelu wektorowym typu *Term-Frequency* opisany jest jako:

$$\vec{d}_j = (d^1_j d^2_j \ldots d^n_j)$$

Każdy współczynnik wektora opisany jest zależnością:

$$d^i_j = TF(t_i, d_j)$$

Model wektorowy – TF

Sposoby obliczania współczynników *TF*:

- suma termów: $$TF(t_i, d_j) = \begin{cases} 0 & dla\ n_{ij} = 0 \\ \dfrac{n_{ij}}{\sum_{k=1}^{m} n_{kj}} & dla\ n_{ij} > 0 \end{cases}$$
- maksimum: $$TF(t_i, d_j) = \begin{cases} 0 & dla\ n_{ij} = 0 \\ \dfrac{n_{ij}}{max_k\, n_{kj}} & dla\ n_{ij} > 0 \end{cases}$$
- logarytm: $$TF(t_i, d_j) = \begin{cases} 0 & dla\ n_{ij} = 0 \\ 1 + \log(1 + \log n_{ij}) & dla\ n_{ij} > 0 \end{cases}$$

Model wektorowy – TF

Współczynniki *TF* służą do normalizacji wartości termów opisujących dokumenty WWW. Dzięki nim można przeskalować duże wartości termów związanych z dokumentami do wartości mniejszych, mieszczących się w zdefiniowanym zakresie.

Współczynniki *TF* powiązane są z każdym termem w każdym dokumencie. Do zapisania współczynników *TF* wymagana jest macierz dwuwymiarowa (w sensie logicznym!).

Model wektorowy – IDF

Załóżmy, że zbiór D jest zbiorem wszystkich dokumentów, zaś zbiór $D_{t_i} = \{d_j | n_{ij} > 0\}$ zbiorem dokumentów zawierających term t_i.

Sposoby obliczania współczynników *IDF*:

- ułamek: $IDF(t_i) = \frac{|D|}{|D_{t_i}|}$
- logarytm: $IDF(t_i) = \log \frac{1 + |D|}{|D_{t_i}|}$

Model wektorowy – IDF

Współczynniki *Inverse Document Frequency* służą do skalowania współczynników wektorów dokumentów.

Dla termów występujących często w różnych dokumentach, istotność tego termu nie może być tak duża, jak termu występującego w niewielu dokumentach.

Czy wpółczynniki *IDF* powiązane są tylko z termami czy również z dokumentami?

Współczynniki *IDF* powiązane są tylko z termami (bez uwzględniania dokumentów).

Model wektorowy – TFIDF

Pojedynczy wiersz w modelu wektorowym typu *Term-Frequency Inverse Document Frequency* opisany jest jako:

$$\vec{d}_j = (d_j^1 d_j^2 \ldots d_j^n)$$

Każdy współczynnik wektora opisany jest zależnością:

$$d_j^i = TF(t_i, d_j)\, IDF(t_i)$$

Model wektorowy *TFIDF* łączy w sobie zalety współczynników *TF* oraz współczynników *IDF*.

TFIDF – przykład

Wektor *TF* dokumentu d_6 (strona wydziału *Computer Science*):

$$\vec{d}_6 = (0 \quad 0 \quad 0{,}026 \quad 0{,}076 \quad 0{,}039)$$

Współczynniki modelu *IDF* dla poszczególnych termów (logarytm):

lab	laboratory	Programming	computer	program
3.04452	3.04452	3.04452	1.43508	0.559616

Wektor *TFIDF* dokumentu d_6:

$$\vec{d}_6 = (0 \quad 0 \quad 0{,}079 \quad 0{,}112 \quad 0{,}022)$$

Macierz typu pozycyjnego

Przykładowa macierz term-dokument typu pozycyjnego:

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	[71]
d_2	0	0	0	0	[7]
d_3	0	[65,69]	0	[68]	0
d_4	0	0	0	[26]	[30,43]
d_5	0	0	0	0	0
d_6	0	0	[40,42]	[1,3,7,13,26,34]	[11,18,61]
d_7	0	0	0	0	[9,42]
d_8	0	0	0	0	[57]
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0
d_{11}	0	0	0	0	0
d_{12}	0	0	0	[17]	0
d_{13}	0	0	0	0	0
d_{14}	[42]	0	0	[41]	[71]
d_{15}	0	0	0	0	[37,38]
d_{16}	0	0	0	0	[81]
d_{17}	0	0	0	0	[68]
d_{18}	0	0	0	0	0
d_{19}	0	0	0	0	[51]
d_{20}	0	0	0	0	0

Słowa kluczowe – wyszukiwanie

Korzystając z modelu wektorowego można przeprowadzić wyszukiwanie dokumentów bazując na obliczaniu odległości pomiędzy wektorami.

W jaki sposób przedstawić zapytanie w formie wektora?

Słowa kluczowe zapytania można przekształcić do termów i bazując na zbiorze wszystkich termów stworzyć wektor reprezentujący to zapytanie.

Słowa kluczowe – wyszukiwanie

Zwracane wyniki będzie można posortować pod względem istotności bazującej na termach poprzez obliczenie odległości pomiędzy wektorem reprezentującym zapytanie, a pozostałymi wektorami reprezentującymi dokumenty.

Odległości pomiędzy wektorami muszą być obliczane z wykorzystaniem wybranej normy metrycznej.

Wyszukiwanie – normy

Najczęściej w technice *IR* wykorzystywane są następujące normy metryczne:

- norma Euklidesowa:

$$\|\vec{q}-\vec{d}_j\|=\sqrt{\sum_{i=1}^{m}(q^i-d_j{}^i)^2}$$

- podobieństwo cosinusowe:

$$\vec{q}\cdot\vec{d}_j=\sum_{i=1}^{m} q^i d_j{}^i$$

Wyszukiwanie – normy

Które ze zwróconych dokumentów dla normy Euklidesowej są bardziej istotne?

Dla podobieństwa euklidesowego dokument jest tym bardziej istotny, im mniejsza jest wartość wynikowa normy.

Które ze zwróconych dokumentów dla normy cosinusowej są bardziej istotne?

Dla podobieństwa cosinusowego dokument jest tym bardziej istotny, im większa jest wartość wynikowa normy.

Wyszukiwanie – przykład

Załóżmy, że korzystając z poznanych wcześniej metod chcemy wyszukać te dokumenty, które zawierają termy *computer* oraz *program*.

Wektor *q* odpowiadający zapytaniu jest określony współrzędnymi:

$$\vec{q} = (0 \quad 0 \quad 0 \quad 0{,}5 \quad 0{,}5)$$

Po nałożeniu współczynników *IDF* na wektor *q* otrzymamy:

$$\vec{q} = (0 \quad 0 \quad 0 \quad 0{,}718 \quad 0{,}28)$$

Wyszukiwanie – przykład

Doc	TFIDF Coordinates (normalized)					$\vec{q}\cdot\vec{d}_j$ (rank)	$\left\vert\vec{q}-\vec{d}_j\right\vert$ (rank)
d_1	0	0	0	0	1	0.363	1.129
d_2	0	0	0	0	1	0.363	1.129
d_3	0	0.972	0	0.234	0	0.218	1.250
d_4	**0**	**0**	**0**	**0.783**	**0.622**	**0.956 (1)**	**0.298 (1)**
d_5	0	0	0	0	1	0.363	1.129
d_6	**0**	**0**	**0.559**	**0.811**	**0.172**	**0.819 (2)**	**0.603 (2)**
d_7	0	0	0	0	1	0.363	1.129
d_8	0	0	0	0	1	0.363	1.129
d_9	0	0	0	0	0	0	1
d_{10}	0	0	0	0	0	0	1
d_{11}	0	0	0	0	0	0	1
d_{12}	0	0	0	1	0	0.932	0.369
d_{13}	0	0	0	0	0	0	1
d_{14}	**0.890**	**0**	**0**	**0.424**	**0.167**	**0.456 (3)**	**1.043 (3)**
d_{15}	0	0	0	0	1	0.363	1.129
d_{16}	0	0	0	0	1	0.363	1.129
d_{17}	0	0	0	0	1	0.363	1.129
d_{18}	0	0	0	0	0	0	1
d_{19}	0	0	0	0	1	0.363	1.129
D_{20}	0	0	0	0	0	0	1

Wyniki wyszukiwania

Zarówno norma euklidesowa, jak i norma cosinusowa pozwalają zwracać zbiory dokumentów posortowane pod względem ich istotności. Obliczana istotność zależy od częstości występujących w ich korpusach termów. Jednak normy te nie uwzględniają faktu występowania wyszukiwanego termu w treści dokumentu.

Należy zatem pamiętać, aby brać pod uwagę tylko te dokumenty, które zawierają wszystkie termy z zapytania opartego o słowa kluczowe.

Wyszukiwanie z operatorami

Silniki wyszukiwania mają możliwość przetwarzania zapytań złożonych ze słów kluczowych wraz z operatorami boolowskimi, np. *AND*, *OR* lub *NOT*.

Wprowadzając domyślne zapytanie złożone ze słów kluczowych, wykorzystywany jest operator *AND*.

W jaki sposób zrealizować operatory OR oraz NOT?

Przed obliczeniem odległości (lub w trakcie obliczeń) można wybierać tylko, te dokumenty, które spełniają podane zależności.

Wyszukiwanie słów z błędami

Problemem w definiowaniu zapytań złożonych ze słów kluczowych są możliwe błędy w zapisie poszczególnych słów. Jedną z metod radzenia sobie z tym problemem jest dekompozycja termów na *n-gramy*.

W przypadku popełnienia błędu w zapisie termu, porównanie fragmentów termów pozwoli na znalezienie podobieństwa i zwrócenie odpowiednich wyników.

Średnia długość stosowanych *n-gramów* waha się w zakresie od 2 do 4.

Wyszukiwanie słów z błędami

Przykładowo, term *program* może być rozłożony na 2-gramy:

{*pr, ro, og, gr, ra, am*}

Term *program* zapisany z błędem, np. *prorgam* zostanie rozłożony na następujące 2-gramy:

{*pr, ro, or, rg, ga, am*}

Porównanie dwóch sekwencji pokazuje, że 2-gramy pokrywają się w 3 na 6 przypadków, w związku z tym można podejrzewać, że termy te są takie same.

Sprzężenie zwrotne oceny wyników

W procesie sortowania wyszukanych dokumentów pod względem istotności, można wykorzystać ocenę zwróconych wyników przez użytkownika na zasadzie sprzężenia zwrotnego.

Użytkownik przydziela wyniki do dwóch zbiorów:

- D_+ – dokumenty istotne,
- D_- – dokumenty nieważne.

Problemem w tym rozwiązaniu jest jednak czas, który użytkownik musi poświęcić na klasyfikację dokumentów.

Pseudo-ocena wyników: np. 5 wyników do D_+, reszta do D_-.

Sprzężenie zwrotne oceny wyników

Po dokonaniu oceny przez użytkownika wektor zapytania jest przeliczany z wykorzystaniem metody *Rocchio.* Dokumenty istotne zwiększają współczynniki wektora zapytania, zaś dokumenty nieistotne współczynniki te osłabiają.

Zależność modyfikująca wektor zapytania *q*:

$$\vec{q}\,' = \alpha \vec{q} + \beta \sum_{d_j \in D_+} \vec{d}_j - \gamma \sum_{d_j \in D_-} \vec{d}_j$$

Ocena wyników – przykład

Załóżmy, że $\alpha = 1$, $\beta = 0.5$, zaś $\gamma = 0$. Zapytanie q zostanie zmodyfikowane przez trzy istotne dokumenty zwrócone przez oryginalne zapytanie. Dodatkowo, wybrane zostaną trzy termy o najwyższych współczynnikach *IDF*: *lab*, *laboratory*, *programming*. Modyfikacja wektora zapytania q będzie wyglądała następująco:

$$\vec{q}\,' = \vec{q} + 0.5(\vec{d}_4 + \vec{d}_6 + \vec{d}_{14})$$

$$\vec{q}\,' = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0.932 \\ 0.363 \end{pmatrix} + \left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.559 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.89 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} 0.445 \\ 0 \\ 0.28 \\ 0.932 \\ 0.363 \end{pmatrix}$$

Ocena wyników – przykład

Po modyfikacji wektora zapytania *q* i ponownym zwróceniu wyników zgodnie z podobieństwem liczonym za pomocą miary cosinusów, otrzymane zostały następujące wartości dla dokumentów:

- d_6 (*Computer Science*): 0.863,
- d_4 (*Chemistry*): 0.846,
- d_{14} (*Music*): 0.754.

Wynik jest lepszy, gdyż zapytanie złożone z termów *computer* oraz *program*, powinno na pierwszym miejscu zwrócić stronę WWW dotyczącą wydziału *Computer Science*.

Miary oceny jakości wyników

W systemach typu *IR* istotne są mechanizmy pomiaru jakości działania algorytmów zwracających zbiory wynikowe. W tym celu opracowany został system *precision-recall*.

Model ten operuje na dwóch zbiorach:

- R_q – zbiór dokumentów zwróconych przez algorytm zgodnie z zapytaniem q,
- D_q – zbiór istotnych dokumentów zgodnych z zapytaniem q, stworzony przez eksperta.

Miary oceny jakości wyników

Ilość zwróconych i istotnych wyników w stosunku do wszystkich istotnych dokumentów jest określany jako *recall*:

$$recall = \frac{|D_q \cap R_q|}{|D_q|}$$

Wartości parametru *recall* mogą się zmieniać w zakresie od 0 do 1.

Jaki jest najgorszy i najlepszy przypadek?

Najgorszy przypadek to wartość 0, kiedy algorytm nie zwrócił żadnego istotnego dokumentu. Najlepszy przypadek to 1, jednak wcale nie oznacza to, że algorytm zwrócił poprawny zbiór wynikowy.

Miary oceny jakości wyników

Ilość zwróconych i istotnych wyników w stosunku do wszystkich zwróconych dokumentów jest nazywany precyzją (ang. *precision*):

$$precision = \frac{|D_q \cap R_q|}{|R_q|}$$

Wartości precyzji mogą się zmieniać w zakresie od 0 do 1.

Jaki jest najgorszy i najlepszy przypadek?

Najgorszy przypadek to wartość 0, kiedy algorytm nie zwrócił żadnego istotnego dokumentu. Najlepszy przypadek to 1, kiedy zbiór wynikowy zwrócony przez algorytm zawiera wszystkie istotne dokumenty.

Miary oceny jakości wyników

Każdy system *IR* dąży do tego, aby jednoczesna wartość obydwu współczynników była równa 1, jednak jest to praktycznie niemożliwe.

Modyfikacja zapytań pod kątem uzyskania maksymalnej wartości (ale nie jednocześnie!) *precision* **lub** *recall* jest jest banalna.

Aby *recall* wyniósł 1, należy tworzyć ogólne zapytania, np. złożone z 1 słowa kluczowego.

Aby *precision* wyniósł 1, należy tworzyć tak szczegółowe zapytania, że będą dotyczyły tylko jednego dokumentu.

Miary oceny jakości wyników

W rzeczywistych systemach *IR* występuje zawsze jedna z zależności:

- $D_q \cap R_q \subset D_q$ – w tym przypadku zbiór wynikowy trzeba powiększyć,
- $D_q \subset R_q$ – w tym przypadku, zbiór wynikowy trzeba zmniejszyć.

Działanie skutecznego systemu *IR* musi być zawsze kompromisem pomiędzy wartością współczynnika *precision* i wartością współczynnika *recall*.

Dziękuję za uwagę!