



Przetwarzanie języka naturalnego/05

2024.04.11

Krzysztof Misztal

misztal.edu.pl

Spis treści

1 Wektoryzacja tekstu

2 Miary podobieństwa

Spis treści

1 Wektoryzacja tekstu

2 Miary podobieństwa

Założmy, że rozważamy problem stworzenia wyszukiwarki, tzn. mamy pewną bazę dokumentów (niekoniecznie stron), oraz użytkownika, który wpisuje jakąś frazę. Chcemy przedstawić mu najlepszy dokument (albo posortowaną listę najlepszych dokumentów).

Problem wyszukiwania

Mając dane zapytanie q wyszukaj najlepszy dokument d z puli D .

Problem wyszukiwania

- jest to istotnie różne od sprawdzania, czy q jako zdanie, należy do jakiegoś modelu języka zadanego przez określony dokument d
- przede wszystkim D jest duże, więc musimy mieć bardzo wydajną metodę (zarówno pamięciowo jak i obliczeniowo)
- zapytania bardzo rzadko są zdaniami
- to, co nas tak na prawdę interesuje to **sens**, **potrzeba**, która doprowadziła do wpisania q , nie zaś samo q

Reprezentacja binarna (set of words)

$$\phi(\text{"Ala ma kota. Ala lubi te\dot{z} psy"}) = \{\text{Ala, ma, kota, lubi, te\dot{z}, psy}\}$$

Jaccard coefficient

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Właściwości:

- $J(A, A) = 1$
- $J(A, B) = 0 \Leftrightarrow A \cap B = \emptyset$
- A i B mogą być dowolnej (różnej) długości
- $J(A, B) \in [0, 1]$

Jaccard coefficient

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$J(\phi(\text{"Ala ma kota"}), \phi(\text{"Ala ma psa"})) =$$

$$= \frac{|\{Ala, ma, kota\} \cap \{Ala, ma, kota\}|}{|\{Ala, ma, kota\} \cup \{Ala, ma, psa\}|}$$

$$= \frac{|\{Ala, ma\}|}{|\{Ala, ma, kota, psa\}|}$$

$$= \frac{2}{4}$$

Jaccard coefficient

- Nie bierze pod uwagę częstotliwości wystąpień słów
- Normalizacja przez sumę mnogościową nie jest najlepsza

Set of words

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Bag of words

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Bag of words – reprezentacja

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

$$\phi(\text{Antony and Cleopatra}) = [157, 4, 232, 0, 57, 2, 2] \in \mathbb{N}^7$$

Term frequency

Term frequency

Częstotliwością termu (term frequency, tf) t w dokumencie d nazywamy liczbę wystąpień t w d i oznaczamy przez $tf_{t,d}$

$$tf_{t,d} = count(t, d) = \#\{i : d[i] = t\}$$


Term frequency – zastosowania i problemy


- Można wykorzystywać do mierzenia na ile dany dokument odzwierciedla zapytanie w wyszukiwarce, tzn.:

$$score(q, d) = \sum_{t \in q} tf_{t,d}$$

- Jeśli szukamy hasła "dog", to dokument zawierający 100 słów "dog" będzie 100 razy lepszy niż ten zawierający jedno słowo "dog"
- Wraz ze wzrostem częstotliwości występowania termu powinna wzrastać "ocena", ale na pewno nie liniowo!

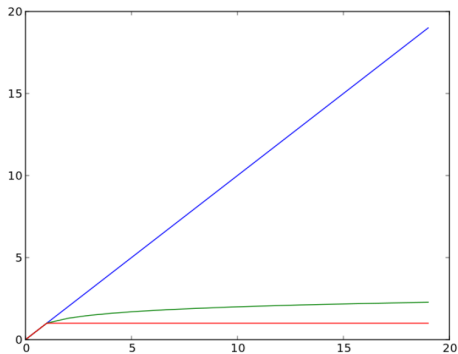
Log term frequency


$$w_{t,d} = \begin{cases} 1 + \log(tf_{t,d}), & \text{jeśli } tf_{t,d} > 0 \\ 0, & \text{wpp.} \end{cases}$$


$$score(q, d) = \sum_{t \in q} w_{t,d} = \sum_{t \in q \cap d} 1 + \log(tf_{t,d})$$

Log term frequency

$tf_{t,d}$	$w_{t,d}$	SOW
0	0	0
1	1	1
2	1.3	1
10	2	1
1000	4	1



Document frequency weighting

- **Rzadkie słowa są bardziej informatywne, niż częste**
- np. jeśli szukamy zapytaniem "William Shakespeare", to zdecydowanie większą wagę należy poświęcić stronom, które zawierają term "Shakespeare" (35,000,000 wyników w Google) niż stronom zawierającym "William" (281,000,000 wyników w Google).
- bardziej skrajnie - szukając danych o muszce owocówce (używając zapytania "melanogaster fly") ważniejsze są strony o konkretnym gatunku ("drosophila melanogaster" - 1,410,000 wyników) niż te o muchach ogólnie (188,000,000)

Document frequency

Document frequency

Częstotliwością termu (Document frequency, df) t w zbiorze dokumentów nazywamy liczbę dokumentów w których występuje t i oznaczamy przez df_t

$$df_t = \sum_{d \in D} \min\{1, tf_{t,d}\} = \#\{d \in D : tf_{t,d} > 0\}$$

Document frequency

- Document frequency jest miara nieinformatywności termu
- Chcąc mieć informatywności musimy ten obiekt "odwrócić"

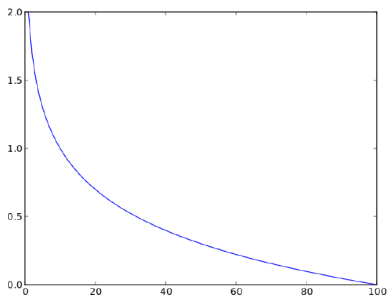
Inverse Document frequency

$$idf_t = \log\left(\frac{N}{df_t}\right)$$

gdzie $N = \#D$ to liczba dokumentów

Inverse Document frequency

df_t	$idf_t (N = 100)$
100	0
99	0.004
98	0.009
...	...
50	0.3
...	...
5	1.3
4	1.4
3	1.5
2	1.7
1	2



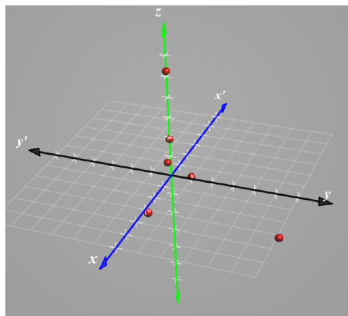
tf-idf weighting

$$tf.idf_{t,d} = \underbrace{(1 + \log(tf_{t,d}))}_{\text{tf- trafność}} \underbrace{\log(\frac{N}{df_t})}_{\text{idf-normalizacja}}$$

$$score(q, d) = \sum_{t \in q \cap d} tf.idf_{t,d}$$

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95



document	x	y	z
Antony and Cleopatra	5.25	1.21	1.51
Julius Caesar	3.18	6.1	0
The Tempest	0	0	1.9
Hamlet	0	1	0.12
Othello	0	0	5.25
Macbeth	0.35	0	0.88

Rysunek: VSM na bazie utworów Szekspira

- **V**ector **S**pace **M**odel
- mamy $\|V\|$ wymiarowa przestrzeń rzeczywista
- każdy term to wymiar przestrzeni (os)
- dokumenty to punkty (wektory) w tej przestrzeni
- bardzo wysoko wymiarowa przestrzeń
- bardzo rzadkie wektory

"Mając dane zapytanie q wyszukaj najlepszy dokument d z puli D "

- zapytania również można potraktować jak dokumenty i wyrazić je w naszym VSM
- dokumenty można posortować wg. trafności
- trafność = bliskość wektorów = odwrotność odległości



Co to jest "bliskość wektorów"?

Spis treści

1 Wektoryzacja tekstu

2 Miary podobieństwa

Odległość między dwoma wektorami

- mamy dokumenty d_i wyrażone jako wektory w $\mathbb{R}^{|V|}$, nazwijmy je x_i
- interesuje nas znalezienie funkcji f , takiej, że:
 - ☐ $f(x_i, x_j) = 0$ gdy $x_i = x_j$
 - ☐ $f(x_i, x_j) = f(x_j, x_i)$
 - ☐ $f(x_i, x_j) > f(x_i, x_k)$ gdy x_k jest bardziej "podobne" do x_i niż x_j (bardzo nieformalnie)

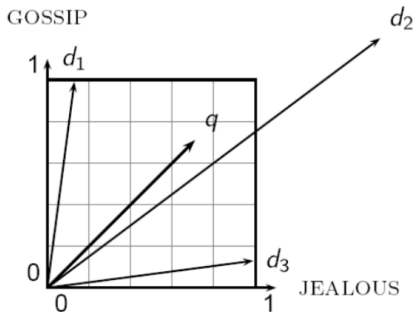
Odległość między dwoma wektorami

Norma euklidesowa różnicy wektorów

$$f(x_i, x_j) = \|x_i - x_j\|$$

Odległość między dwoma wektorami

$$f(x_i, x_j) = \|x_i - x_j\|$$



$$f(q, d_2) = \|q - d_2\| > \|q - d_1\| = f(q, d_1)$$

$$f(q, d_2) = \|q - d_2\| > \|q - d_3\| = f(q, d_3)$$

Odległość między dwoma wektorami

- Wyobraźmy sobie sytuację, gdzie mamy dokument d' będący konkatenacją dokumentu d z samym sobą
- "Semantycznie" te dwa dokumenty mają tę samą informację
- Odległość euklidesowa może być dowolnie duża

Odległość między dwoma wektorami

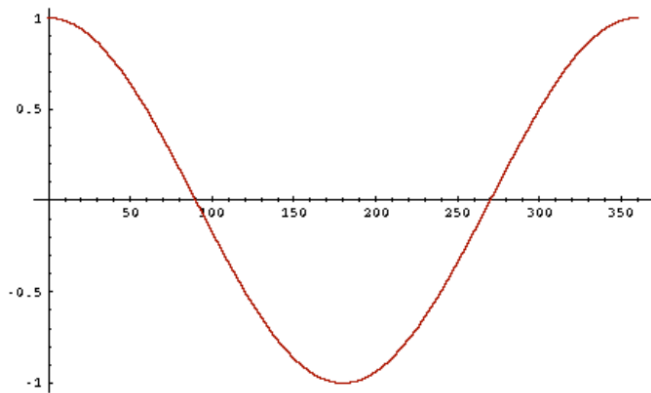
- Wyobraźmy sobie sytuację, gdzie mamy dokument d' będący konkatenacją dokumentu d z samym sobą
- "Semantycznie" te dwa dokumenty mają tę samą informację
- Odległość euklidesowa może być dowolnie duża
- Idea: używajmy np. kąta zamiast odległości

Liczenie kąta między wektorami w wysoko wymiarowej przestrzeni

Prosta obserwacja, następujące działania są równoważne

- Sortowanie dokumentów po malejącym kącie między nimi
- Sortowanie dokumentów po rosnącym kosinusie kąta między nimi

$$\cos(\alpha)$$



Kosinus kąta przy tfidf

$$\forall t, d : tf.idf_{t,d} \geq 0$$

$$\angle(x_i, x_j) \in [0, \pi/2]$$

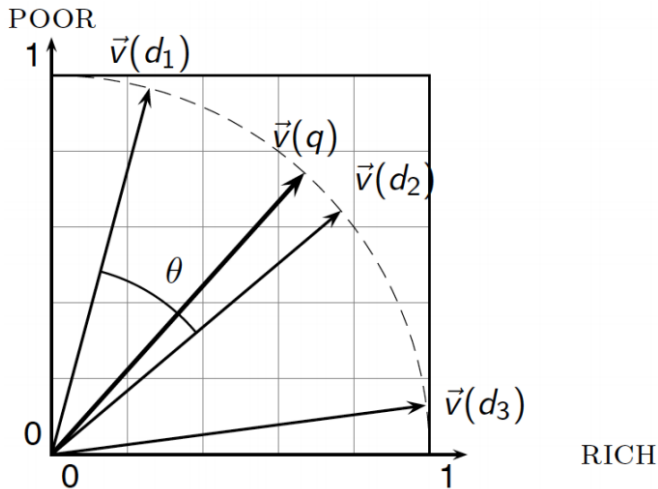
$$\cos(\angle(x_i, x_j)) \in [0, 1]$$

Kosinus kąta

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^{|V|} x_i y_i}{\sqrt{\sum_{i=1}^{|V|} x_i^2} \sqrt{\sum_{i=1}^{|V|} y_i^2}}$$

- x i y to wektory tf.idf
- $\cos(x, y)$ to kosinus kąta między nimi lub czasem "podobienstwo kosinusowe" (cosine similarity) tych wektorów
- gdyby x i y były jednostkowe, to wystarczyłoby liczyć iloczyn skalarny

Znormalizowana reprezentacja VSM



Proces porównywania dokumentów - klasyczna wersja VSM

- 1 Policz $tfidf_{t,d}$ dla każdego dokumentu i każdego termu
- 2 Zapisz reprezentacje VSM każdego dokumentu korzystając z tfidf
- 3 Znormalizuj każdy z wektorów (podziel go przez jego normę)
- 4 W przypadku potrzeby porównania dwóch dokumentów - policz iloczyn skalarny pomiędzy ich reprezentacjami

Generalizacije tfidf

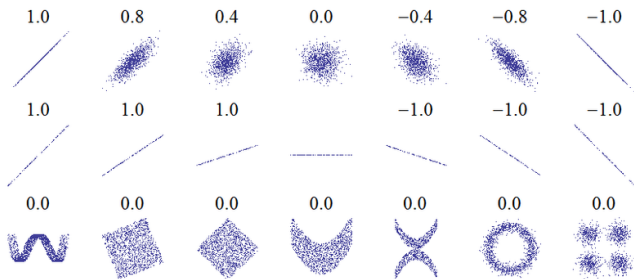
Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^\alpha$, $\alpha < 1$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$				

Czy to jedyna możliwość?

Jest wiele innych, używanych do analizy tekstu metryk, m.in.

- Korelacja Pearsona
- Uśredniona dywergencja Kullbacka-Leiblera

Korelacja Pearsona



$$r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}$$

$$r_{xy} = \cos(\angle(x - \bar{x}, y - \bar{y}))$$

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(x) \log \left(\frac{P(x)}{Q(x)} \right) dx$$

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(x) \log \left(\frac{P(x)}{Q(x)} \right) dx$$

$$D_{KL}(x||y) = \sum_t w_{t,x} \log \left(\frac{w_{t,x}}{w_{t,y}} \right)$$

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} P(x) \log \left(\frac{P(x)}{Q(x)} \right) dx$$

$$D_{KL}(x||y) = \sum_t w_{t,x} \log \left(\frac{w_{t,x}}{w_{t,y}} \right)$$

$$D_{JS}(P||Q) = \frac{D_{KL}(P||\frac{P+Q}{2}) + D_{KL}(Q||\frac{P+Q}{2})}{2}$$

Klastrowanie dokumentów - Purity

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.1	0.5	0.5	0.5	0.38
classic	0.56	0.85	0.98	0.85	0.84
hitech	0.29	0.54	0.51	0.56	0.53
re0	0.53	0.78	0.75	0.78	0.77
tr41	0.71	0.71	0.72	0.78	0.64
wap	0.32	0.62	0.63	0.61	0.61
webkb	0.42	0.68	0.57	0.67	0.75

Na podstawie "Similarity Measures for Text Document Clustering"
- Anna Huang (University of Waikato)

Klastrowanie dokumentów - Entropia

Data	Euclidean	Cosine	Jaccard	Pearson	KLD
20news	0.95	0.49	0.51	0.49	0.54
classic	0.78	0.29	0.06	0.27	0.3
hitech	0.92	0.64	0.68	0.65	0.63
re0	0.6	0.27	0.33	0.26	0.25
tr41	0.62	0.33	0.34	0.3	0.38
wap	0.75	0.39	0.4	0.39	0.4
webkb	0.93	0.6	0.74	0.61	0.51

Na podstawie "Similarity Measures for Text Document Clustering"
- Anna Huang (University of Waikato)

Dziękuję za uwagę.