Eksploracja danych i wyszukiwanie informacji w mediach społecznościowych

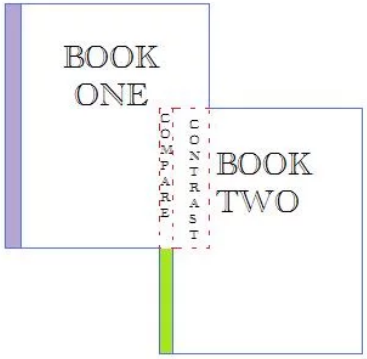
Wykład 2 - reprezentacja tekstu

dr inż. Julian Sienkiewicz

15 października 2018

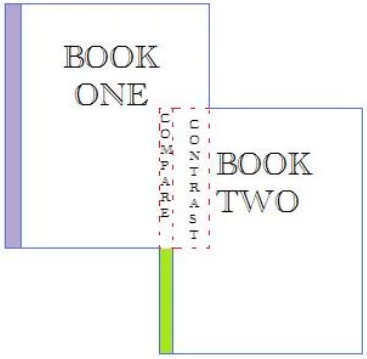
W jaki sposób możemy przedstawić dokument?

- w “codziennym życiu” zwykle nie mamy z tym większych problemów – używamy **opisu**, określając, mniej lub bardziej składnie, zawartość dokumentu,
- w ten sam sposób jesteśmy w stanie dokonać **porównania** dwóch dokumentów



W jaki sposób możemy przedstawić dokument?

- w “codziennym życiu” zwykle nie mamy z tym większych problemów – używamy **opisu**, określając, mniej lub bardziej składnie, zawartość dokumentu,
- w ten sam sposób jesteśmy w stanie dokonać **porównania** dwóch dokumentów



- w przypadku automatycznym, chcielibyśmy otrzymać jakąś określoną **reprezentację dokumentu**,
- dobrze byłoby, aby taka reprezentacja umożliwiała wyznaczanie **statystyk**, czyli była taka sama dla róznych tekstów,
- potrzebne będzie również wykonywanie **porównań**

Miary długości

Najprostszymi do wyznaczenia i często też bardzo wygodnymi statystykami są:

- liczba znaków (liter i znaków interpunkcyjnych lub samych liter),
- liczba słów,

Te zmienne są jednak często dość mocno skorelowane ze sobą i ciężko uważać je za odrębne.

Miary długości

Najprostszymi do wyznaczenia i często też bardzo wygodnymi statystykami są:

- liczba znaków (liter i znaków interpunkcyjnych lub samych liter),
- liczba słów,

Te zmienne są jednak często dość mocno skorelowane ze sobą i ciężko uważać je za odrębne.

Miary złożoności

W wielu przypadkach możliwe jest również określenie poziomu **złożoności** tekstu. Tu pomocne są np. następujące miary:

- miara (indeks) Herdana C,
- zmodyfikowana miara Herdana z
- indeks (wskaźnik czytelności) FOG,

Miara Herdana

- zdefiniowana jako $C = \frac{\log V}{\log M}$,
- V - liczba tokenów (liczba różnych słów), M - długość tekstu,
- dla tekstu bez powtórzeń $C = 1$,
- logarytmy umożliwiają “spłaszczenie” funkcji
- w przypadku wielu dokumentów można użyć zmodyfikowanego indeksu Herdana, który jest po prostu indeksem C poddanym standaryzacji (standaryzacji Z),
- z przedstawia się wzorem

$$z_{N,M} = \frac{N - \mu(M)}{\sigma(M)}$$

- $\mu(M)$ oraz $\sigma(M)$ wyznaczane są po zestawie dokumentów – umożliwia to wzięcie pod uwagę efektu fluktuacji

Wskaźnik FOG

- zdefiniowana jako

$$F = 0.4 \left(\frac{\text{liczba słów}}{\text{liczba zdań}} + 100 \frac{\text{liczba złożonych słów}}{\text{liczba słów}} \right)$$

- złożone słowa to takie (w jęz. angielskim), które mają ponad dwie sylaby,
- problemy: nie zawsze złożone słowa są trudne

Interpretacja wskaźnika FOG

- liczba lat formalnej edukcji potrzebnej do zorumienie tekstu
- np. teksty dla szerokiej publiczności powinny mieć F co najwyżej 12 – ocna to poziom maturzysty
- istnieje sporo podobnych wskaźników, np. Flesch-Kincaid

Bag-of-words

- **bag-of-words** (BOW) jest chyba najprostszą reprezentacją tekstu,
- jak sama nazwa wskazuje, zakładamy w nim, że zawartośc dokumentu to po prostu poszczególne słowa, bez względu na ich kolejność pojawiania się w tekście,

Bag-of-words

- **bag-of-words** (BOW) jest chyba najprostszą reprezentacją tekstu,
- jak sama nazwa wskazuje, zakładamy w nim, że zawartośc dokumentu to po prostu poszczególne słowa, bez względu na ich kolejność pojawiania się w tekście,

Przykład

- John likes to watch movies,
- Mary likes movies too
- John also likes football,

$$\begin{pmatrix} \text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{Mary} & \text{too} & \text{also} & \text{football} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Bag-of-words

- **bag-of-words** (BOW) jest chyba najprostszą reprezentacją tekstu,
- jak sama nazwa wskazuje, zakładamy w nim, że zawartośc dokumentu to po prostu poszczególne słowa, bez względu na ich kolejność pojawiania się w tekście,

Przykład

- John likes to watch movies,
- Mary likes movies too
- John also likes football,

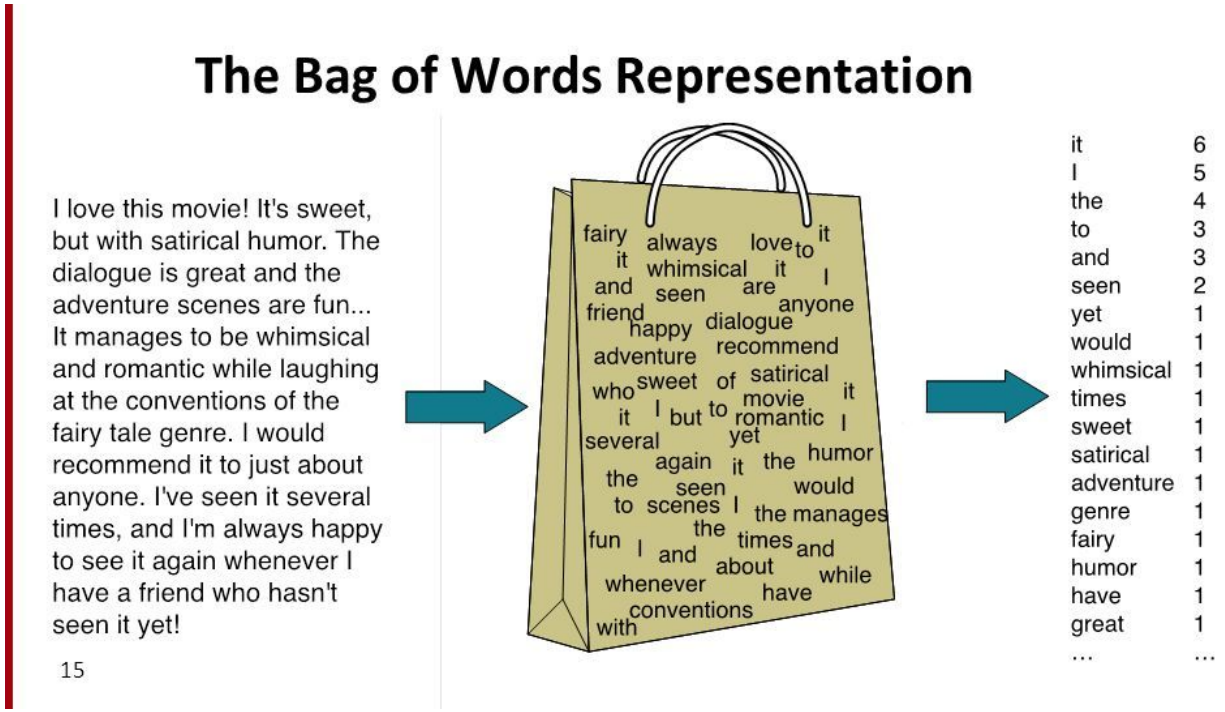
John	likes	to	watch	movies	Mary	too	also	football
1	1	1	1	1	0	0	0	0
0	1	0	0	1	1	1	0	0
1	1	0	0	0	0	0	1	1

Innymi słowy tworzymy po prostu słownik słów, zaznaczając w nim ilość ich występowania.

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

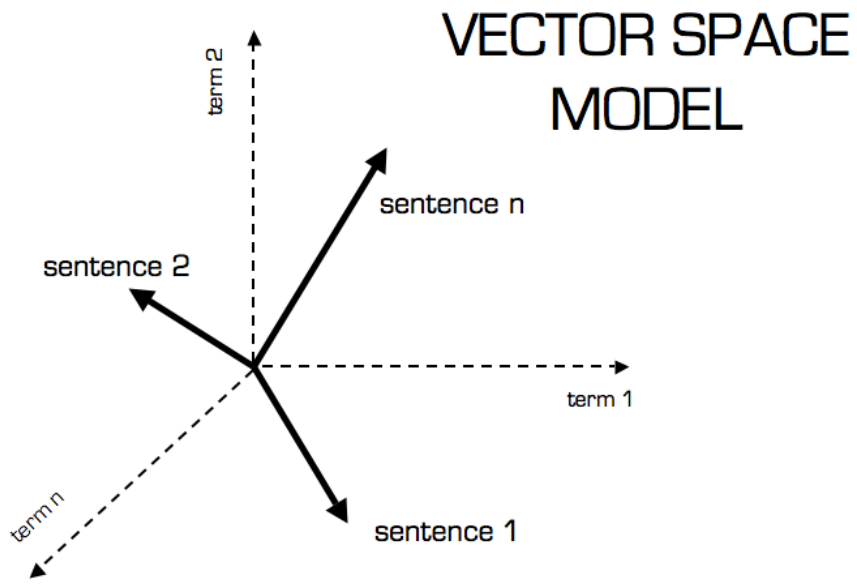
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…



- oczywiście z takiej reprezentacji nie da się stworzyć sensownego tekstu (brak informacji o pozycji słów i gramatyce),
- BOW jest jednak przydatne do tworzenia **klasyfikatorów** opartych na czynnikach (features)
- problem rozsnącego słownika zwykle rozwiązywany przez **hashing trick**

Model Przestrzeni Wektorowej

- Model Przestrzeni Wektorowej (Vector Space Model - VSM) jest pojęciem związanym z BOW, ale nie tożsamym z nim,
- odwołujemy się tu do pojęcia wektora

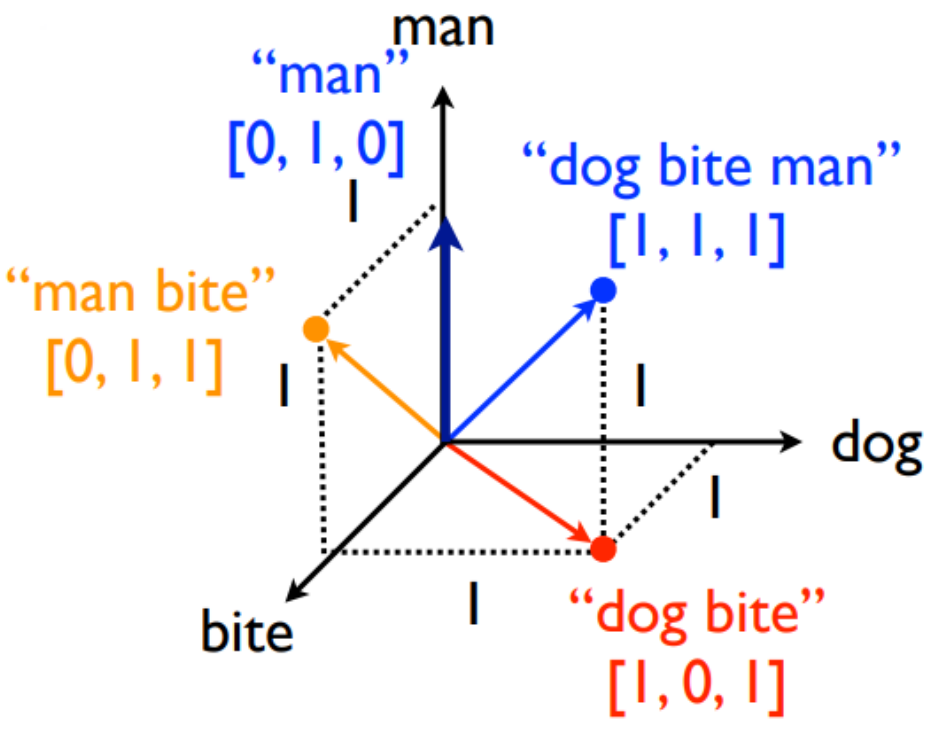


Model Przestrzeni Wektorowej

- zakładamy, że każde słowo to **oś** (czyli jest kierunkiem) w przestrzeni,
- w efekcie każdy obiekt (np. dokument albo słowo) może być reprezentowane przez wektor w tak skonstrukowanej przestrzeni

Prosty przykład

	dog	*man*	*bite*
doc_1	1	1	1
doc_2	1	0	1
doc_3	0	1	1
doc_4	0	1	0



Jak wyznaczyć podobieństwo dwóch dokumentów?

Najprościej za pomocą iloczynu skalarnego:

$$D(A, B) = \sum_{i=1}^{i=N} A_i B_i$$

gdzie N to liczba słów w słowniku, a A_i i B_i to informacja, czy słowo i wystąpiło, odpowiednio, w dokumecie A i B.

Jak wyznaczyć podobieństwo dwóch dokumentów?

Najprościej za pomocą iloczynu skalarnego:

$$D(A, B) = \sum_{i=1}^{i=N} A_i B_i$$

gdzie N to liczba słów w słowniku, a A_i i B_i to informacja, czy słowo i wystąpiło, odpowiednio, w dokumecie A i B.

- pokaźnym problemem takiego podejścia jest to, że nie zakłada ono, iż dokumenty mogą mieć różną długość.
- w przypadku np. wyszukiwania, jest większa sznasa na to, że dłuższy dokuemnt będzie zawierał dane słowo

Odległość

Innym sposobem jest wyznaczenie odległości (Euklidesowej) pomiędzy poszczególnymi dokumentami / zdaniami

$$D(A, B) = \sqrt{\sum_{i=1}^{i=N} (A_i - B_i)^2}$$

Odległość

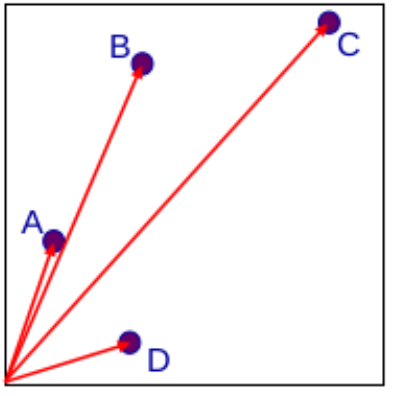
Innym sposobem jest wyznaczenie odległości (Euklidesowej) pomiędzy poszczególnymi dokumentami / zdaniami

$$D(A,B) = \sqrt{\sum_{i=1}^{i=N}(A_i - B_i)^2}$$

Podobieństwo Cosinusowe

Można również wyznaczyć tzw. podobieństwo cosinusowe

$$D(A,B) = \cos(A,B) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}}$$



Odległość

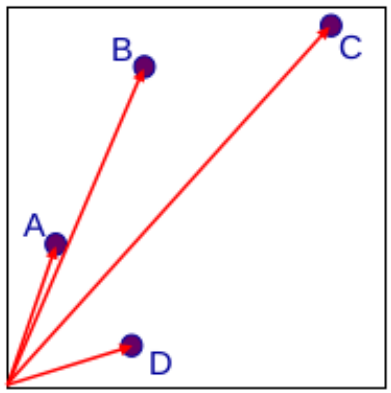
Innym sposobem jest wyznaczenie odległości (Euklidesowej) pomiędzy poszczególnymi dokumentami / zdaniami

$$D(A,B) = \sqrt{\sum_{i=1}^{i=N}(A_i - B_i)^2}$$

Podobieństwo Cosinusowe

Można również wyznaczyć tzw. podobieństwo cosinusowe

$$D(A,B) = \cos(A,B) = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}}$$



- dokumenty “w tym samym kierunku” są do siebie podobne,
- w odróżnieniu od odległości ograniczona miara $D \in \langle 0; 1 \rangle$

Weźmy realny przykład

- Movie: Rocky (1976)
- Plot:

Rocky Balboa is a struggling boxer trying to make the big time. Working in a meat factory in Philadelphia for a pittance, he also earns extra cash as a debt collector. When heavyweight champion Apollo Creed visits Philadelphia, his managers want to set up an exhibition match between Creed and a struggling boxer, touting the fight as a chance for a "nobody" to become a "somebody". The match is supposed to be easily won by Creed, but someone forgot to tell Rocky, who sees this as his only shot at the big time. Rocky Balboa is a small-time boxer who lives in an apartment in Philadelphia, Pennsylvania, and his career has so far not gotten off the canvas. Rocky earns a living by collecting debts for a loan shark named Gazzo, but Gazzo doesn't think Rocky has the viciousness it takes to beat up deadbeats. Rocky still boxes every once in a while to keep his boxing skills sharp, and his ex-trainer, Mickey, believes he could've made it to the top if he was willing to work for it. Rocky, goes to a pet store that sells pet supplies, and this is where he meets a young woman named Adrian, who is extremely shy, with no ability to talk to men. Rocky befriends her. Adrain later surprised Rocky with a dog from the pet shop that Rocky had befriended. Adrian's brother Paulie, who works for a meat packing company, is thrilled that someone has become interested in Adrian, and Adrian spends Thanksgiving with Rocky. Later, they go to Rocky's apartment, where Adrian explains that she has never been in a man's apartment before. Rocky sets her mind at ease, and they become lovers. Current world heavyweight boxing champion Apollo Creed comes up with the idea of giving an unknown a shot at the title. Apollo checks out the Philadelphia boxing scene, and chooses Rocky. Fight promoter Jergens gets things in gear, and Rocky starts training with Mickey. After a lot of training, Rocky is ready for the match, and he wants to prove that he can go the distance with Apollo. The 'Italian Stallion', Rocky Balboa, is an aspiring boxer in downtown Philadelphia. His one chance to make a better life for himself is through his boxing and Adrian, a girl who works in the local pet store. Through a publicity stunt, Rocky is set up to fight Apollo Creed, the current heavyweight champion who is already set to win. But Rocky really needs to triumph, against all the odds...

Co nam da zliczanie słów:

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3

Ale wszystkie one są tak samo dla nas ważne?

rank	term	freq.	rank	term	freq.
1	a	22	16	creed	5
2	rocky	19	17	philadelphia	5
3	to	18	18	has	4
4	the	17	19	pet	4
5	is	11	20	boxing	4
6	and	10	21	up	4
7	in	10	22	an	4
8	for	7	23	boxer	4
9	his	7	24	s	3
10	he	6	25	balboa	3
11	adrian	6	26	it	3
12	with	6	27	heavyweigh	3
13	who	6	28	champion	3
14	that	5	29	fight	3
15	apollo	5	30	become	3

Potrzebujemy więc sposobu, aby jakoś “wyróżnić” te słowa, które faktycznie odnoszą się do istotnej treści.

Potrzebujemy więc sposobu, aby jakoś “wyróżnić” te słowa, które faktycznie odnoszą się do istotnej treści.

Można oczywiście dokonać *stopword reduction*, czyli pozbyć się takic słów jak a, the, is... traktując je jako funkcyjne, ale to nie do końca jest rozwiązanie “systemowe”.

Potrzebujemy więc sposobu, aby jakoś “wyróżnić” te słowa, które faktycznie odnoszą się do istotnej treści.

Można oczywiście dokonać *stopword reduction*, czyli pozbyć się takic słów jak a, the, is... traktując je jako funkcyjne, ale to nie do końca jest rozwiązanie “systemowe”.

Można ocenić jak relatywnie ważne jest słowo, w odniesieniu do innych dokumentów.

$$idf_i = \log\left(\frac{M}{df_i}\right)$$

gdzie M to liczba dokumentów, które rozpatrujemy, a df_i to liczba dokumentów, w których występuje słowo i.

Potrzebujemy więc sposobu, aby jakoś “wyróżnić” te słowa, które faktycznie odnoszą się do istotnej treści.

Można oczywiście dokonać *stopword reduction*, czyli pozbyć się takic słów jak a, the, is... traktując je jako funkcyjne, ale to nie do końca jest rozwiązanie “systemowe”.

Można ocenić jak relatywnie ważne jest słowo, w odniesieniu do innych dokumentów.

$$idf_i = \log\left(\frac{M}{df_i}\right)$$

gdzie M to liczba dokumentów, które rozpatrujemy, a df_i to liczba dokumentów, w których występuje słowo i.

Daje to tzw. **odwrotną częstość w dokumentach** (IDF – inverse term frequency).

Daje to następujący efekt:

rank	term	idf	rank	term	idf
1	doesn	11.66	16	creed	6.84
2	adrain	10.96	17	paulie	6.82
3	viciousness	9.95	18	packing	6.81
4	deadbeats	9.86	19	boxes	6.75
5	touting	9.64	20	forgot	6.72
6	jergens	9.35	21	ease	6.53
7	gazzo	9.21	22	thanksgivin	6.52
8	pittance	9.05	23	earns	6.51
9	balboa	8.61	24	pennsylvani	6.50
10	heavyweigh	7.18	25	promoter	6.43
11	stallion	7.17	26	befriended	6.38
12	canvas	7.10	27	exhibition	6.31
13	ve	6.96	28	collecting	6.23
14	managers	6.88	29	philadelphia	6.19
15	apollo	6.84	30	gear	6.18

- czynnik IDF jest tym większy im dany wyraz **rzadziej** występuje w całym zbiorze dokumentów
- nie musi to jednak oznaczać, że jest bardzo istotny dla tego konkretnego dokumentu – może to być np. błąd lub jakieś mało popularne, wyszukane słowo
- potrzebny jest jeszcze czynnik, który odnosi się do częstości występowania słowa w danym tekście, czyli **częstość termów**

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

gdzie $n_{i.j}$ to liczba wystąpień wyrazu i w dokumencie j

- razem daje to tzw. **TF-IDF**

$$(\mathrm{tf} - \mathrm{idf})_{i,j} = tf_{i,j} \times idf_i$$

I końcowo mamy:

rank	term	idf	rank	term	idf
1	rocky	96.72	16	meat	11.76
2	apollo	34.20	17	doesn	11.66
3	creed	34.18	18	adrain	10.96
4	philadelphia	30.95	19	fight	10.02
5	adrian	26.44	20	viciousness	9.95
6	balboa	25.83	21	deadbeats	9.86
7	boxing	22.37	22	touting	9.64
8	boxer	22.19	23	current	9.57
9	heavyweigh	21.54	24	jergens	9.35
10	pet	21.17	25	s	9.29
11	gazzo	18.43	26	struggling	9.21
12	champion	15.08	27	training	9.17
13	match	13.96	28	pittance	9.05
14	earns	13.01	29	become	8.96
15	apartment	11.82	30	mickey	8.96

- tak otrzymane wartości mogą stanowić współrzędne wektorów
- problem związany z założeniem **niezależności bazy** – Generalized Vector Space Model

Generalized Vector-Space Model

- Basic VSM: $sim(Q,D) = \sum_w Q_w D_w$
- Generalized vector-space: $sim(Q,D) = \sum_v \sum_w Q_v \cdot D_w \cdot S_{v,w}$

  "similarity" of A and B

  $Q_v \cdot D_w$: tf-idf weights of v in Q, w in D
- Example:
  - Q = "pet lion"
  - D = "cats and dogs"
  - basic: sim(Q,D) = 0
  - generalized: $sim(Q,D) = Q_{pet}D_{cat} + Q_{pet}D_{dog} + Q_{lion}D_{cat}$

	dog	lion	cat	pet
dog	1	0	0	1
lion	0	1	1	0
cat	0	1	1	1
pet	1	0	1	1