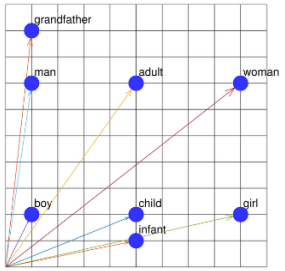


Wyszukiwanie i Przetwarzanie Informacji

Information Retrieval & Search

Irmina Masłowska

irmina.maslowska@cs.put.poznan.pl

Information Retrieval models

Klasyczne modele IR

- model Boole'owski
- model wektorowy (VSM – Vector Space Model)
- model probabilistyczny (BIR – Binary Independence Model)

Nieklasyczne modele IR

- model oparty na zbiorach rozmytych (fuzzy sets)
- rozszerzony model Boole'owski
- model LSI (Latent Semantic Indexing)
- model oparty na sieciach neuronowych (NN)
- uogólniony model wektorowy (Generalized VSM)
- nieklasyczne modele probabilistyczne (sieci Bayesowskie, belief networks, inference networks ...)

...

Term Independence Assumption

- Wszystkie klasyczne modele IR bazują na założeniu, że termy indeksujące są niezależne, czyli na podstawie znajomości wagi a_{ij} przyporządkowanej parze (k_i, d_j) nie możemy nic powiedzieć o wadze a_{lj} dla pary (k_l, d_j): $i \neq l$
- Założenie o niezależności słów kluczowych jest uproszczeniem dyktowanym:
  - efektywnością i prostotą obliczeń
  - trudnością w modelowaniu związków między słowami (zależność od konkretnych kontekstów)
  - zadowalającymi wynikami mimo tego założenia

Latent Semantic Indexing (LSI)*

- Motywacja: wyrazy są niejednoznaczne (synonimy, homonimy) – zamiast dopasowania *wyrazów* lepiej dokonywać dopasowania *pojęć* (*concept matching* vs. *index term matching*)
- Model VSM nie dostarcza mechanizmów pozwalających na rozróżnienie kilku znaczeń tego samego termu (jak np. pączek) i przeciwnie – synonimy, czy wyrazy bliskoznaczne traktuje jako niezależne, ortogonalne wymiary w przestrzeni termów

*) LSI a.k.a. LSA – Latent Semantic Analysis
Deerwester et *al*. – patent w 1988 r.

IR Models – notation

- Niech T będzie liczbą termów indeksujących (rozmiarem słownika), k_i – i-tym słowem kluczowym, D – kolekcją, czyli zbiorem wszystkich dostępnych dokumentów, Q – zbiorem zapytań
- $K=\{ k_1, k_2, ..., k_T \}$ jest zbiorem wszystkich termów indeksujących
- Z każdym słowem kluczowym k_i dokumentu d_j związana jest waga $a_{ij} > 0$ (ew. $a_{ij} \geq 0$), dla słów kluczowych niewystępujących w tekście dokumentu $a_{ij} = 0$
- Stąd każdemu dokumentowi przyporządkowany jest wektor $d_j = (a_{1j}, a_{2j}, ..., a_{Tj})$
- Niech g_i będzie funkcją, która zwraca wagę związaną ze słowem kluczowym k_i dowolnego T-wymiarowego wektora, np.: $g_i(d_j) = a_{ij}$
- $sim(q, d_j)$ jest funkcją rangującą, która przyporządkowuje wartości rzeczywiste parom (q, d_j): $q \in Q$, $d_j \in \mathrm{D}$
- Funkcja sim definiuje uporządkowanie (ranking) dokumentów względem zapytania

Term-Document Matrix

M	d_1	d_2	...	d_N
k_1	a_{11}	a_{12}	...	a_{1N}
k_2	a_{21}	a_{22}	...	a_{2N}
k_3	a_{31}	a_{32}	...	a_{3N}
...	...	...		...
k_T	a_{T1}	a_{T2}	...	a_{TN}

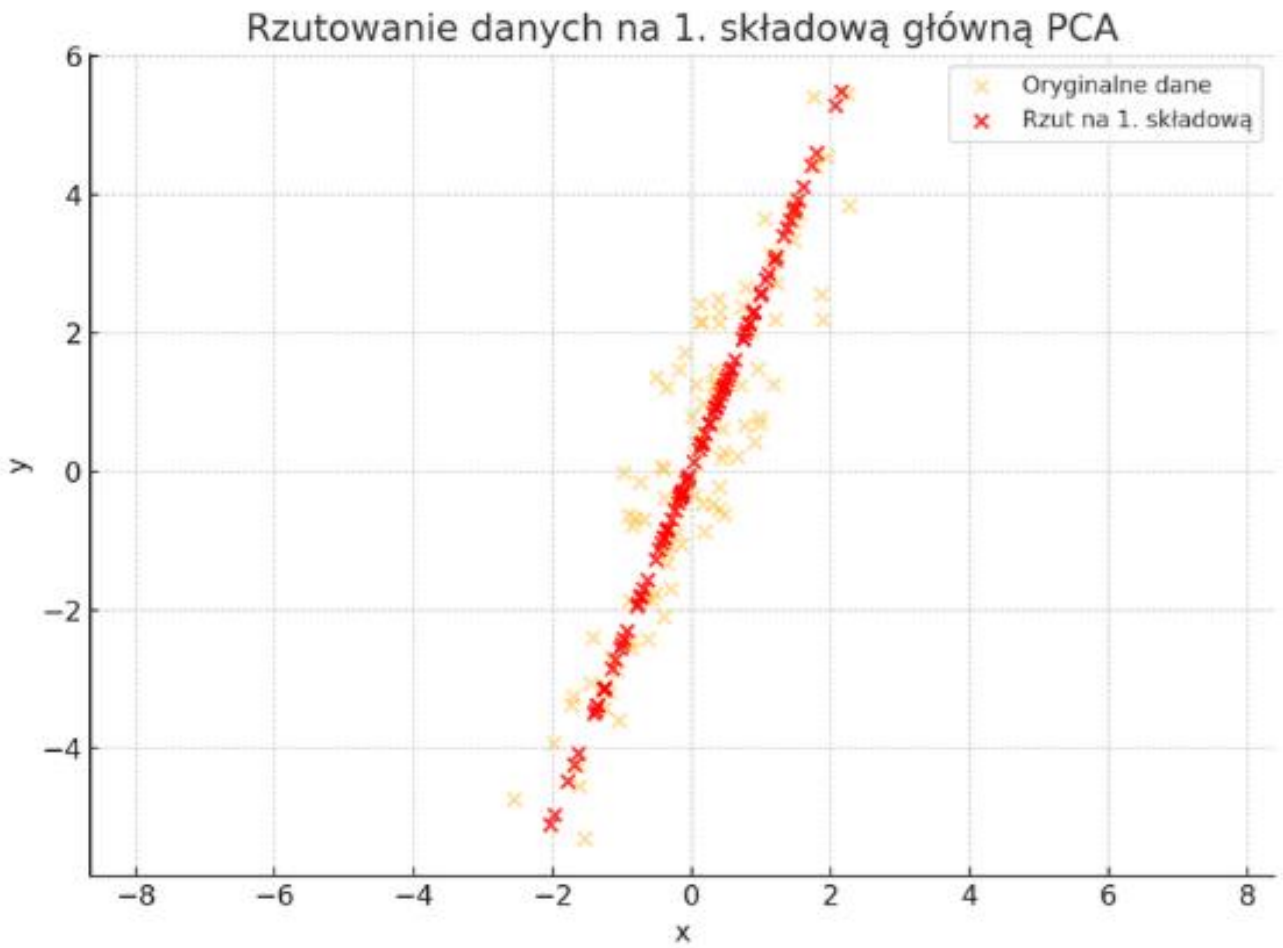
Latent Semantic Indexing (LSI)

- W oryginalnej macierzy M asocjacji *term-document* każdy element odpowiada wadze a_{ij} wyrazu k_i w dokumencie d_j (zwykła częstość *tf* lub *tf-idf*) $M[T \times N]$
- Główna idea: dokonać odwzorowania wektorów związanych z dokumentami i zapytaniami do przestrzeni o mniejszej (niż T) liczbie wymiarów
- Nowa przestrzeń ma być przestrzenią pojęć (zamiast termów) a reprezentacja dokumentów w nowej przestrzeni ma zachować jak najwięcej informacji z oryginalnej reprezentacji

Principal Component Analysis (PCA)

- Technika analizy danych stosowana do redukcji wymiarów danych poprzez wyodrębnienie składowych głównych, które najlepiej opisują zmienność w zbiorze danych
- Polega na obliczeniu składowych głównych (eigenvectors) macierzy kowariancji danych i wykorzystaniu ich do przekształcenia danych na nowy zestaw ortogonalnych współrzędnych
- Poprzez rzutowanie danych na pierwsze kilka składowych głównych PCA pozwala uzyskać dane o niższym wymiarze (mniejsza przestrzeń) jednocześnie zachowując jak najwięcej informacji odnośnie zmienności danych

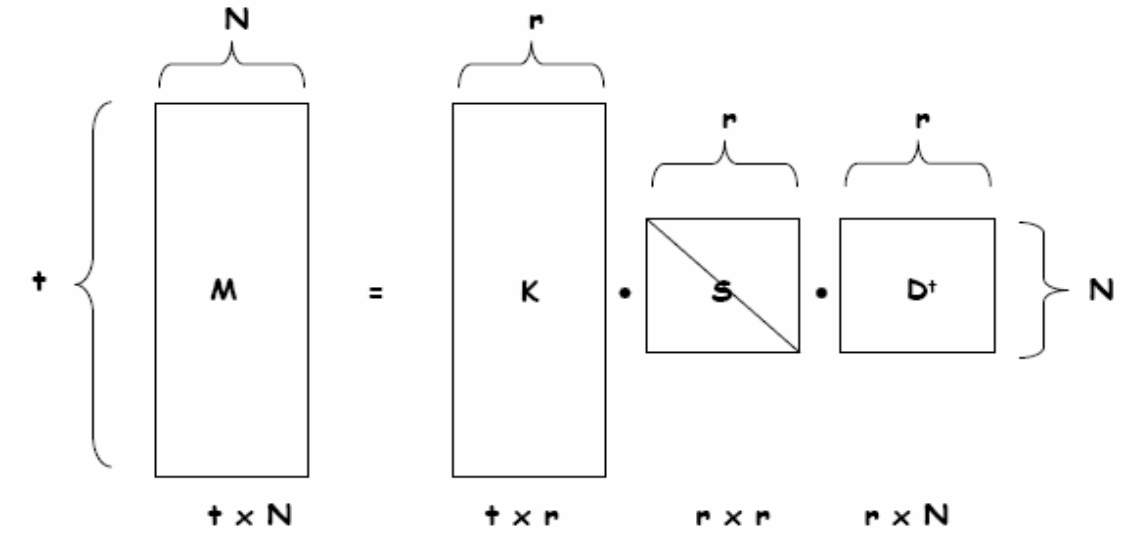
Principal Component Analysis (PCA)



Singular-Value Decomposition (SVD)

- Dekompozycja SVD (czyli dekompozycja na składowe osobliwe) macierzy M
  $M = KSD^T$
- Macierz K to macierz wektorów własnych uzyskana z macierzy MM^T (macierz korelacji term - to-term)
- Macierz D to macierz wektorów własnych uzyskana z macierzy M^TM (macierz korelacji document-to-document)
- Macierz $S[r \times r]$ to diagonalna macierz wartości osobliwych, gdzie $r=min(T, N)$ to rząd macierzy M

Singular-Value Decomposition (SVD)



- Macierz K to macierz wektorów własnych uzyskana z macierzy MM^T (macierz korelacji term-to-term)
- Macierz D to macierz wektorów własnych uzyskana z macierzy M^TM (macierz korelacji document-to-document)
- Macierz $S[r \times r]$ to diagonalna macierz wartości osobliwych, gdzie $r=min(T, N)$ to rząd macierzy M

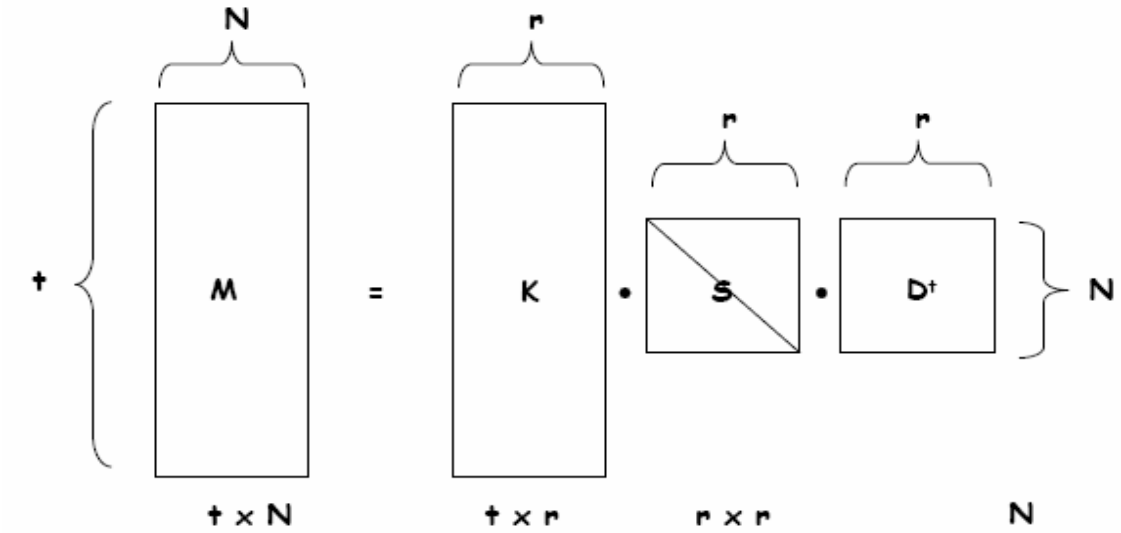
Singular-Value Decomposition (SVD)

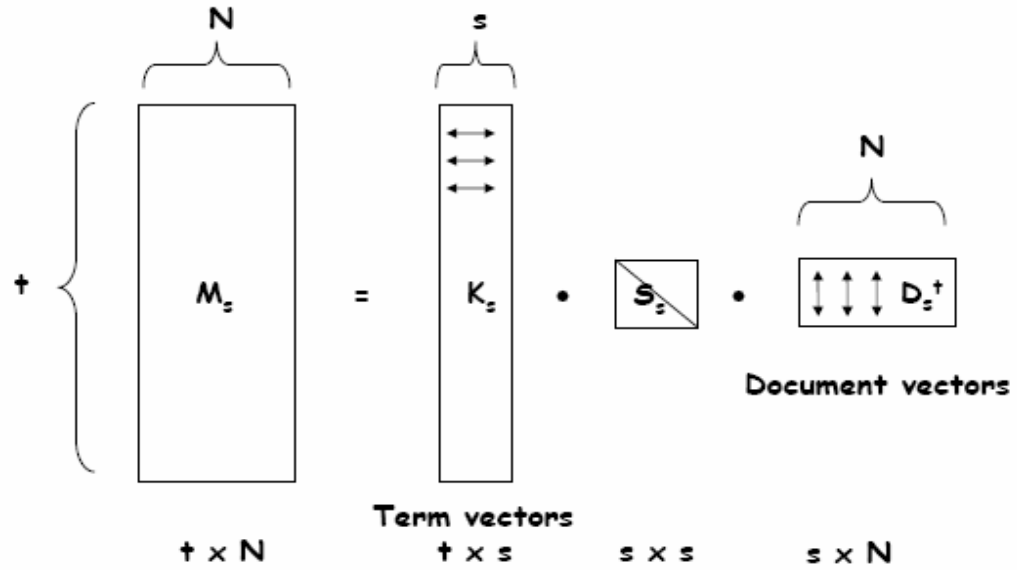
- Redukcja reprezentacji polega na zachowaniu jedynie pewnej liczby s największych wartości z macierzy diagonalnej S (gdzie $s < r$)
- Reszta jest usuwana wraz z odpowiadającymi im kolumnami K i D^T

  – jest to równoważne zastąpieniu macierzy S macierzą S_s, w której wszystkie mniejsze wartości na przekątnej zostają wyzerowane: $S_{s+1,s+1}=0$; $S_{s+2,s+2}=0$; ... ; $S_{r,r}=0$

  $M_s = K_s S_s D_s^T$
- s to wymiar nowej przestrzeni – przestrzeni pojęć (*concept space*)

Singular-Value Decomposition (SVD)





Patrz: przykład w pliku *xlsx*

Ranking in LSI Model

- Związek między dowolnymi termami kolekcji w zredukowanej przestrzeni można odczytać z odpowiedniej pozycji macierzy $M_s M_s^T$
- Związek między dowolnymi dwoma dokumentami (lub **dokumentem i zapytaniem**) w zredukowanej przestrzeni można odczytać z odpowiedniej pozycji macierzy $M_s^T M_s$

$$M_s^T M_s = (K_s S_s D_s^T)^T K_s S_s D_s^T$$
$$= D_s S_s K_s^T K_s S_s D_s^T$$
$$= D_s S_s S_s D_s^T$$
$$= (D_s S_s)(D_s S_s)^T$$

Ranking in LSI Model

- Aby nie powtarzać kosztownej analizy SVD dla każdego nadchodzącego zapytania *q,* poddaje się je tej samej transformacji, która mapuje *M* na *D:*

$$M = KSD^T$$
$$K^{-1}M = K^{-1}KSD^T$$
$$S^{-1}\,K^T\,M = D^T$$
$$(D^T)^T = (S^{-1}\,K^T\,M)^T = M^TK\,(S^{-1})^T = M^TK\,S^{-1}$$
$$D = M^TK\,S^{-1} \Rightarrow q' = q^T\,K_sS_s^{-1}$$

- Ranking dla *q* można teraz wyznaczyć licząc miarę kosinusową w nowej przestrzeni dokumentów, czyli porównując q' z kolumnami d_j macierzy D_s^T

Patrz: przykład w pliku *xlsx*

LSI – summary

- Podobne dokumenty mają podobny rozkład (częstości) termów
- Ogromny zbiór termów zawiera potencjalnie nieistotne lub nadmiarowe elementy, które warto odfiltrować
- Użycie SVD do redukcji rozmiarów macierzy częstości pozwala na zachowanie jedynie pewnej liczby najistotniejszych *wymiarów semantycznych*, które odpowiadają głównym wzorcom współwystępowania w danych

LSI – summary

- SVD rozkłada oryginalną macierz na komponenty ortogonalne, które najlepiej wyjaśniają zmienność danych
- Wybierając tylko kilka głównych komponentów, zachowujemy największe, globalne wzorce współwystępowania słów, kosztem odrzucenia drobnych, lokalnych zależności
- Dzięki temu model generalizuje – przykładowo umieszcza synonimy blisko siebie nawet wtedy, gdy w macierzy oryginalnej nigdy nie wystąpiły w tym samym dokumencie

LSI – summary

- Wymuszenie redukcji liczby wymiarów reprezentacji term-dokument powinno poskutkować zbliżeniem termów, które wykazują podobieństwa występowania
- Dokumenty o reprezentacji bliskiej zapytaniu w oryginalnej przestrzeni pozostają bliskie także po redukcji
- Jakość wyników wyszukiwania nie tylko nie ucierpi w wyniku redukcji liczby wymiarów, ale może wręcz ulec polepszeniu dzięki wyłonieniu dokumentów podobnych tematycznie, ale wykorzystujących inne termy niż te użyte w zapytaniu (przez co niepodobnych w klasycznym modelu VSM)

Embeddings

- W kontekście analizy danych i przetwarzania języka naturalnego, *embedding* oznacza przekształcenie obiektów (np. wyrazów, dokumentów, obrazów) na wektory liczbowe w przestrzeni o niższym wymiarze, tak aby zachować ich istotne właściwości lub relacje
- *pol.*: osadzenie, zanurzenie, reprezentacja wektorowa
- Prostym statycznym *embeddingiem* będzie np. **wektor *s*-wymiarowy opisujący dany wyraz** – wyekstrahowany na drodze LSI z dużej macierzy wyrazów i dokumentów aby zawierać esencję jego znaczenia i odzwierciedlać znaczeniowe podobieństwo między wyrazami

Embeddings

Method	Static/Contextual	Model Type	Key Strength
Word2Vec	Static	Neural Network	Simple, efficient
GloVe	Static	Matrix Factorization	Captures global + local relationships
FastText	Static	Subword-based	Handles rare/misspelled words
ELMo	Contextual	BiLSTM	Dynamic embeddings for polysemy
BERT	Contextual	Transformer (Bidirectional)	Deep understanding of context
GPT	Contextual	Transformer (Unidirectional)	Great for text generation
T5	Contextual	Text-to-Text Transformer	Versatile for multiple NLP tasks