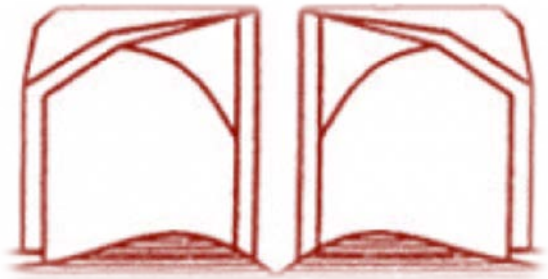


روش‌های یادگیری ماشین در پردازش زبان طبیعی

تشابه‌یابی و بازیابی اطلاعات

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران – دانشکده علوم و فنون نوین

فهرست

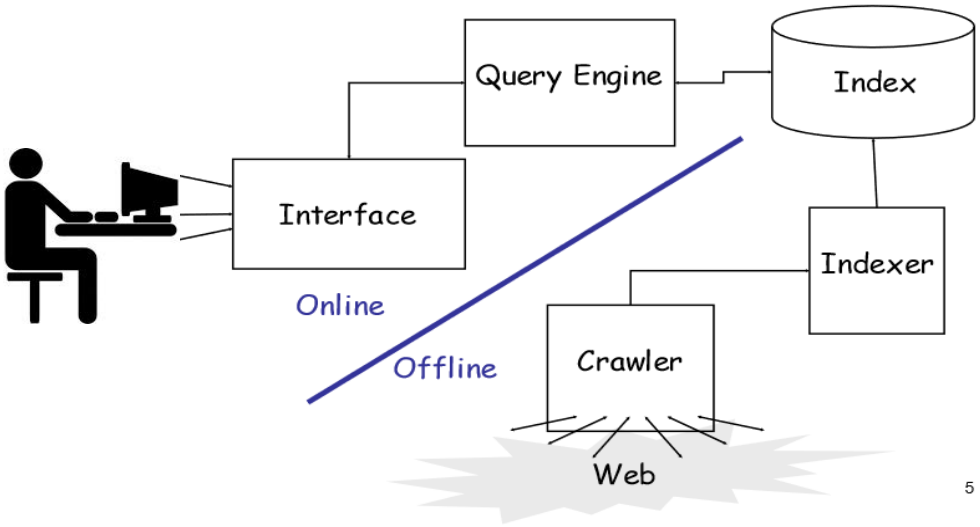
تشابه‌یابی . . .

- محاسبه میزان شباهت دو متن

- کاربردها
  - بازیابی اطلاعات (Information Retrieval) متنی
    - موتورهای جستجو (Search Engines)
    - سیستم های پرسش و پاسخ (Question-Answer Systems)
    - بازیابی اطلاعات غیرمتنی
      - Audio Retrieval
        - Spoken document Retrieval
        - Music Retrieval
      - Image Retrieval
  - تشخیص سرقت علمی (Plagiarism Detection)

تشابه‌یابی: بازیابی اطلاعات . . .

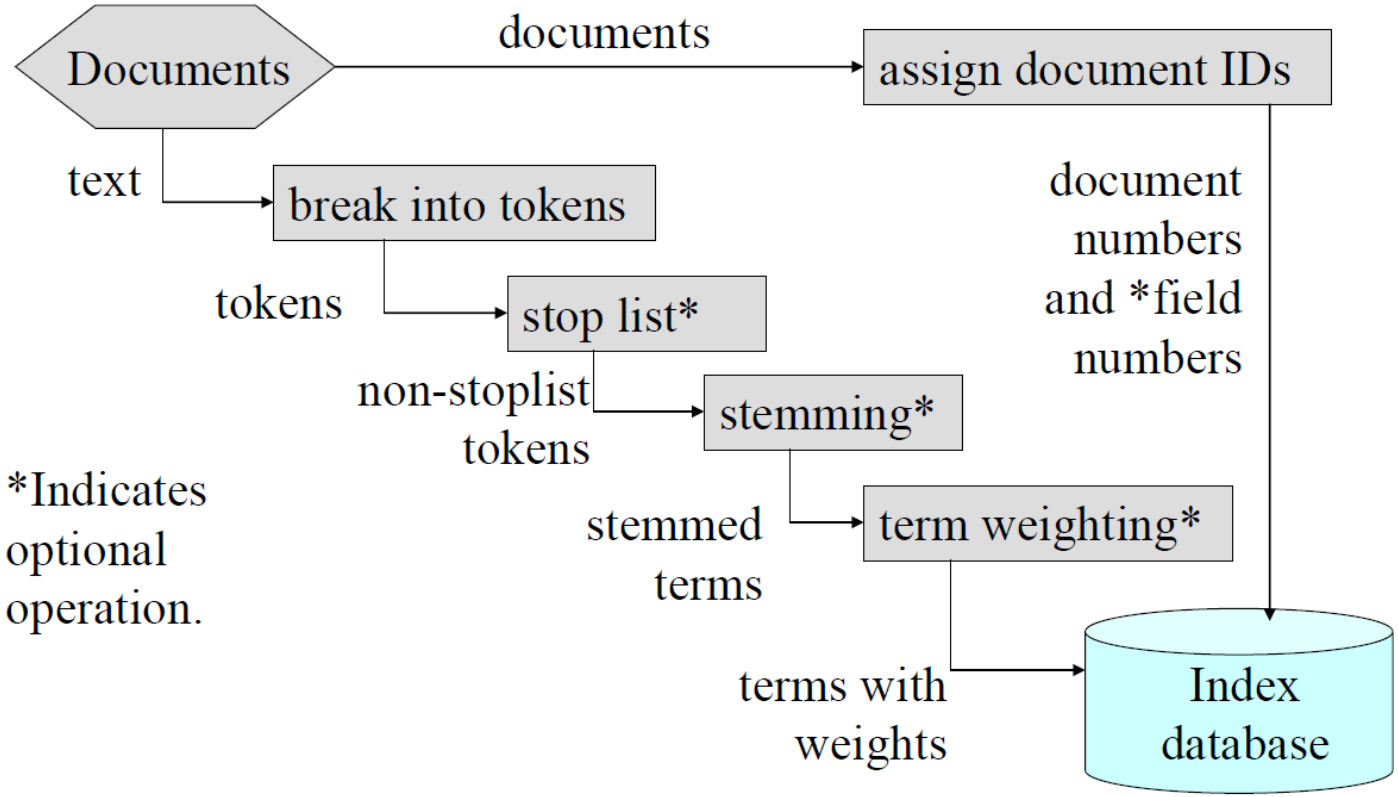
- ساختار کلی



- **واحد نمایه‌گذاری** (Indexing)
  - جمع‌آوری اطلاعات از وب و ساختار دادن به آن
- **واحد پردازش پرسش و جستجو**
  - بررسی پرسش و مقایسه با متون نمایه‌گذاری شده

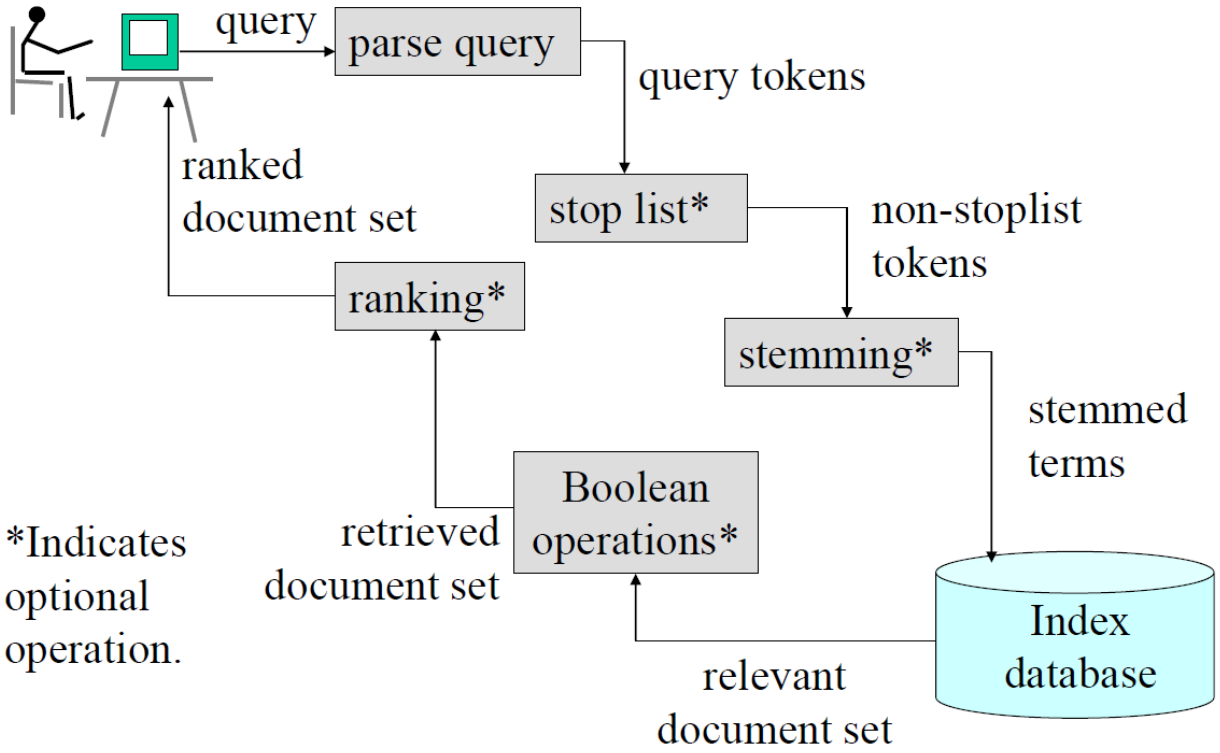
تشابه‌یابی: بازیابی اطلاعات . . .

- واحد نمایه‌گذاری (Indexing)



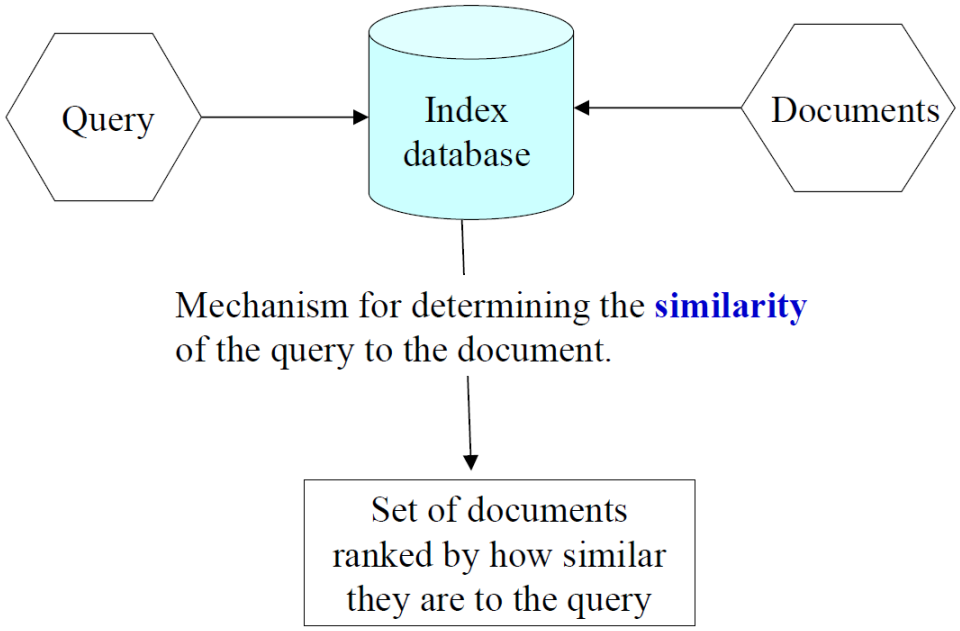
تشابه‌یابی: بازیابی اطلاعات . . .

- واحد پردازش پرسش و جستجو
  - بازیابی سندهای مشابه و مرتبط با پرسش



تشابه‌یابی: بازیابی اطلاعات . . .

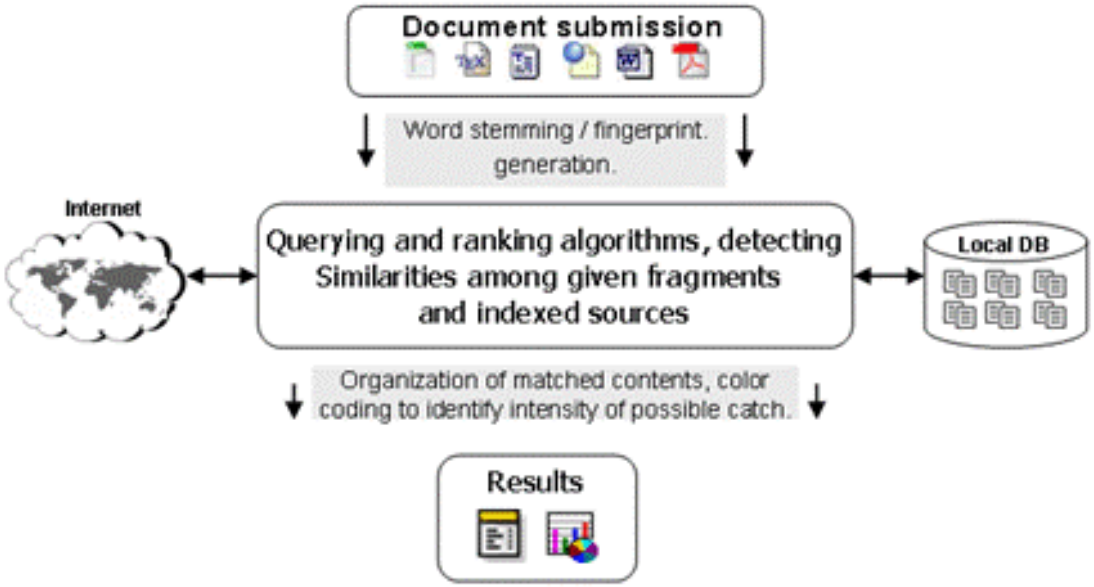
- رتبه‌بندی بر اساس شباهت
  - استفاده از شباهت محاسبه شده برای رتبه‌بندی خروجی‌ها



تشابه‌یابی: سرقت علمی . . .

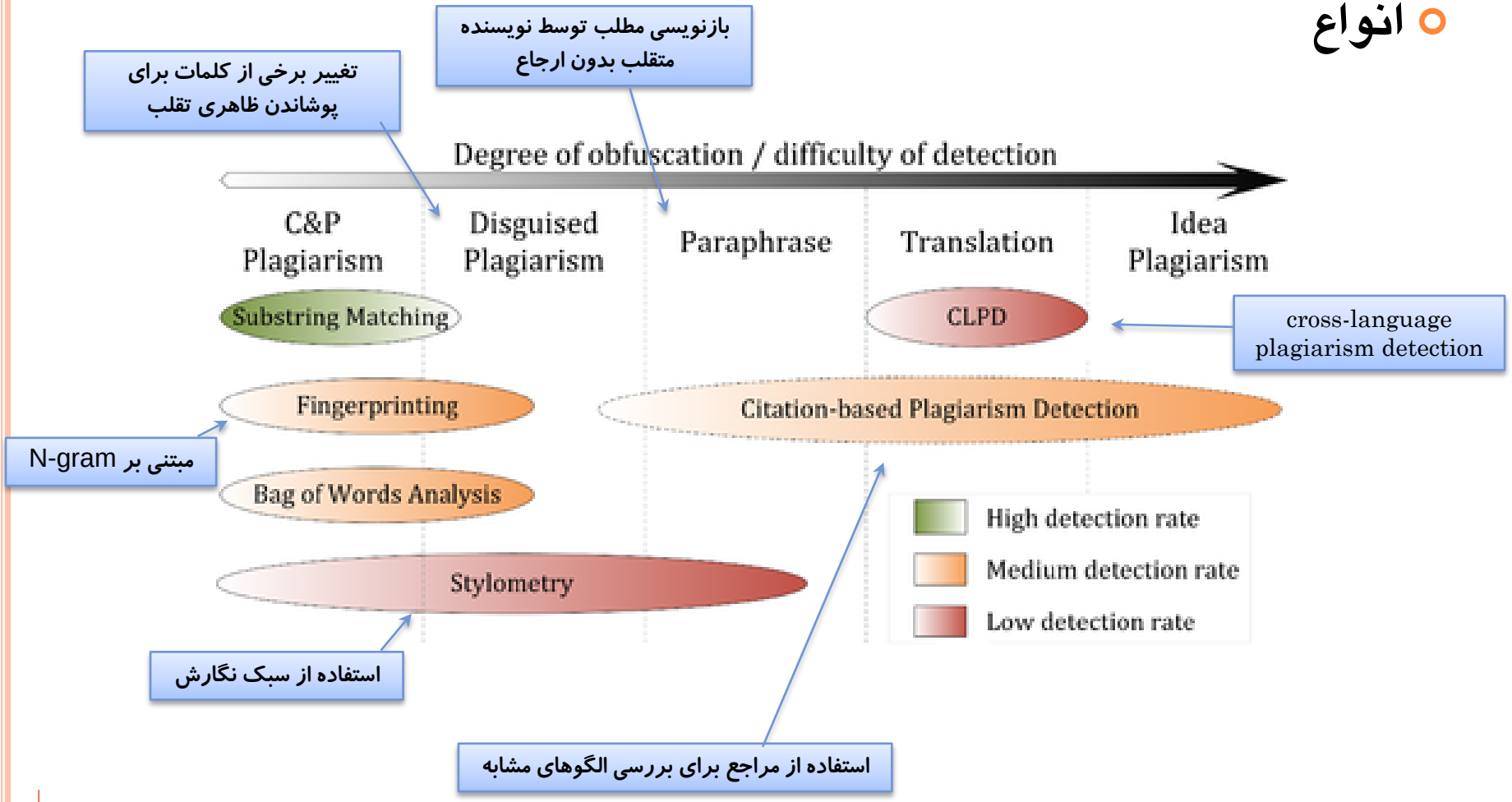
- سرقت علمی در
  - متون علمی (مقاله، پایان نامه، اختراع و ...)
  - کدهای کامپیوتری

- سیستم تشخیص سرقت



تشابه‌یابی: سرقت علمی

- انواع



تشابه‌یابی

- مساله: دو متن داده شده، میزان شباهت آنها چقدر است؟

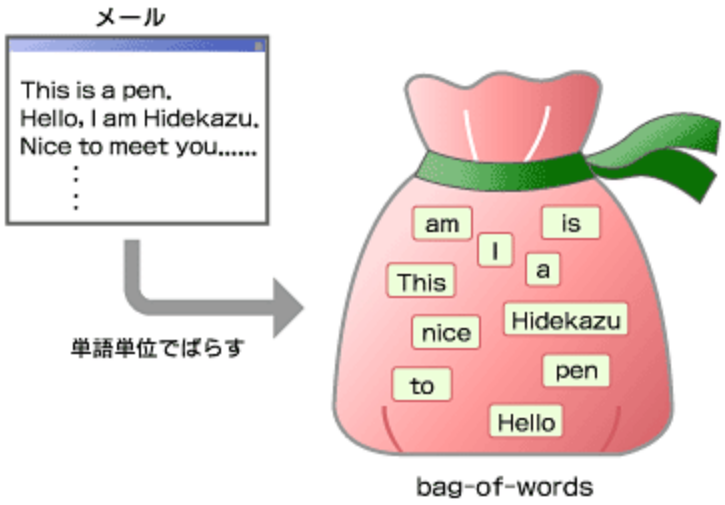
- مثال: سه متن زیر، چقدر به هم شبیه هستند؟

  - d1: ant ant bee
  - d2: dog bee dog hog dog ant dog
  - d3: cat gnu dog eel fox

- ویژگی‌های احتمالی برای محاسبه شباهت
  - طول دو متن
  - تعداد کلمات مشترک دو متن
  - تعداد تکرار کلمات (مشترک) دو متن
  - نوع کلمات: عمومی یا خاص

تبدیل متن به بردار ویژگی . . .

- روش کیسه لغات (Bag of Words)
  - نمایش یک متن (جمله یا سند) تنها بر اساس کلمات آن
  - عدم توجه به ترتیب کلمات و گرامر
    - در بازیابی اطلاعات معمولا این فرض به کار می رود



bag-of-words

- کاربرد
  - تشابه یابی
  - دسته بندی

- مشکل: دو جمله زیر یکی هستند
  - John is quicker than Mary
  - Mary is quicker than John

تبدیل متن به بردار ویژگی . . .

- تبدیل متن (سند) به یک بردار «عددی» از ویژگی‌ها
- گام اول: استخراج کلیه کلمات از کلیه متون مورد مطالعه
  - در کاربردهای واقعی: یک بردار از کلمات (چندهزار بعدی)
- گام دوم: انتساب مقادیر عددی به بردار بر اساس حضور کلمات در متن

تبدیل متن به بردار ویژگی . . .

- بردار دودویی (Binary) . . .
  - برای هر کلمه، اگر آن کلمه در متن باشد، مقدار یک، و در غیر این صورت مقدار صفر
  - مثال

document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

کلمات	ant	bee	cat	dog	eel	fox	gnu	hog
وزن (مقدار)	1	1	0	0	0	0	0	0

- بردار دودویی برای d_1

وزن (مقدار)	1	1	0	1	0	0	0	1

- بردار دودویی برای d_2

تبدیل متن به بردار ویژگی

- **بردار دودویی (Binary)**
  - برای هر کلمه، اگر آن کلمه در متن باشد، مقدار یک، و در غیر این صورت مقدار صفر
  - مثال

document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

- **ماتریس وقوع (Incidence Matrix)**
  - ماتریس Term-document

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	1	1						
d_2	1	1		1				1
d_3			1	1	1	1	1	

اگر سند i حاوی کلمه j باشد، $w_{ij}= 1$
است؛ در غیراینصورت $w_{ij}= 0$

محاسبه تشابه دو متن . . .

- یافتن تشابه بین دو متن بر اساس شباهت دو بردار
  - وجود معیارهای مختلف
- معیار شباهت: فاصله کسینوسی
  - محاسبه زاویه بین دو بردار
  - هرکدام از بردارها: n بعدی

ضرب داخلی (نقطه‌ای)

$$\mathbf{d}_1.\mathbf{d}_2 = x_{11}x_{21} + x_{12}x_{22} + x_{13}x_{23} + \ldots + x_{1n}x_{2n}$$

$$\cos(\theta) = \frac{\mathbf{d}_1.\mathbf{d}_2}{|\mathbf{d}_1|\,|\mathbf{d}_2|}$$

زاویه بین دو بردار

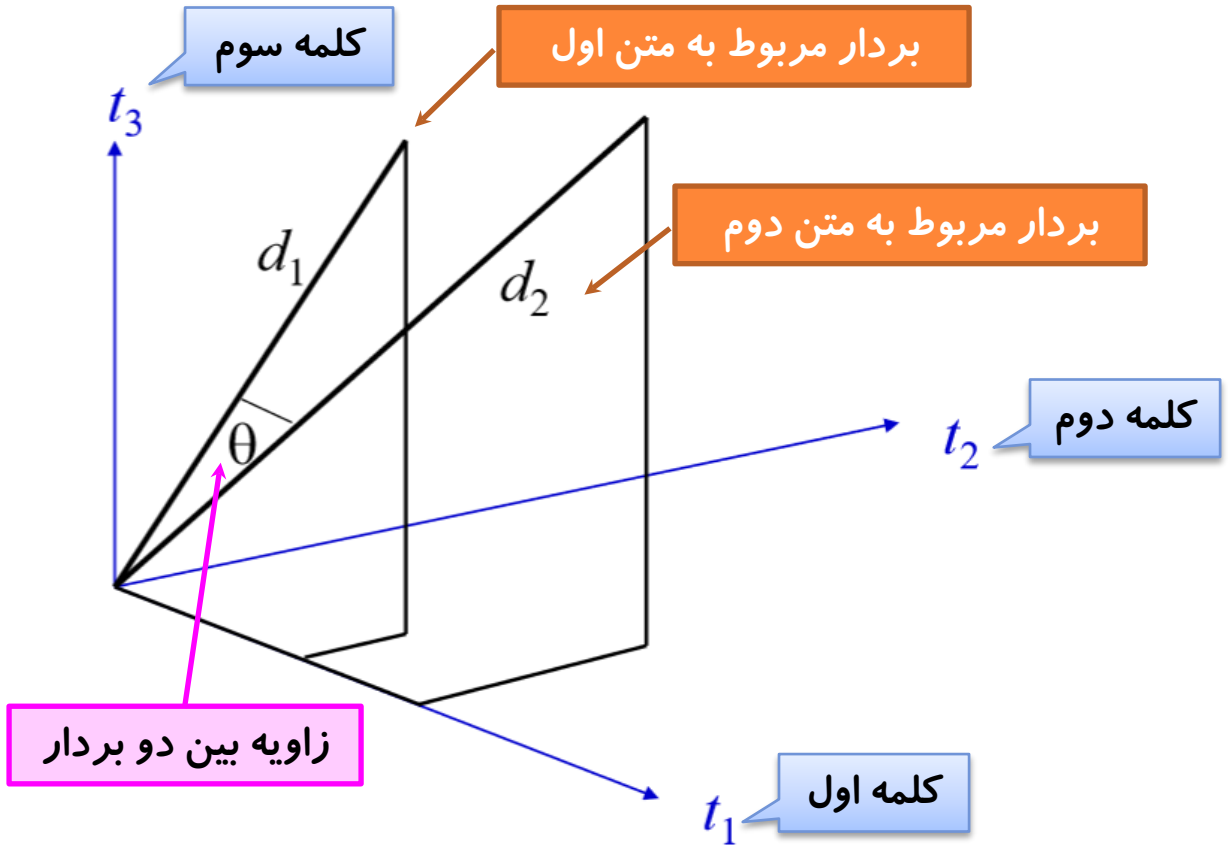
اندازه بردار

$$|\mathbf{d}|^2 = x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2$$

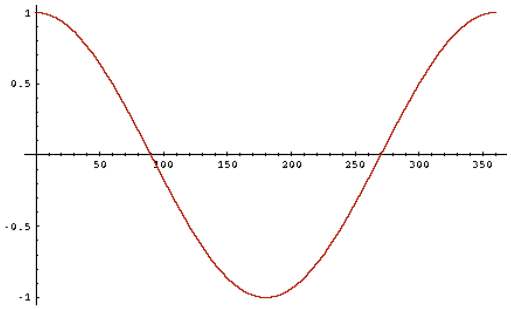
$$|\mathbf{d}| = (x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2)^{1/2}$$

محاسبه تشابه دو متن . . .

- معیار شباهت فاصله کسینوسی در فاصله هندسی
  - زاویه بین دو بردار d1 و d2



محاسبه تشابه دو متن . . .



معیار شباهت فاصله کسینوسی

- مقدار بین صفر و یک
- اگر دو بردار (متن) یکسان باشد، مقدار فاصله کسینوسی برابر است با ۱
- اگر دو بردار (متن) کاملا متفاوت باشد، مقدار فاصله کسینوسی برابر است با ۰

- مثال: فاصله بین دو بردار d1 و d2
  - بردار d1= (1, 1, 0, 0, 0, 0, 0, 0)
  - بردار d2= (1, 1, 0, 1, 0, 0, 0, 1)

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	1	1						
d_2	1	1		1				1
d_3			1	1	1	1	1	

$$cos(\theta) = \frac{\boldsymbol{d}_1 . \boldsymbol{d}_2}{|\boldsymbol{d}_1|\,|\boldsymbol{d}_2|} = \frac{1\times1+1\times1+0\times0+0\times1+0\times0+0\times0+0\times0+0\times1}{\sqrt{1^2+1^2}\times\sqrt{1^2+1^2+1^2+1^2}} = \frac{2}{\sqrt{8}} = 0.71$$

محاسبه تشابه دو متن

- معیار شباهت فاصله کسینوسی- مثال

document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	1	1						
d_2	1	1		1				1
d_3			1	1	1	1	1	

$$\cos(\theta) = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{|\mathbf{d}_1|\,|\mathbf{d}_2|}$$

	d_1	d_2	d_3
d_1	1	0.71	0
d_2	0.71	1	0.22
d_3	0	0.22	1

تبدیل متن به بردار ویژگی . . .

- بردار فراوانی عبارت (TF: Term-Frequency) . . .
  - منظور از Frequency در پردازش متن، «تعداد» است.
  - برای هر کلمه، تعداد وقوع آن کلمه در متن به عنوان ویژگی استفاده می‌شود

document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

W_{ij}= i تعداد تکرار کلمه j در سند

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	2	1						
d_2	1	1		4				1
d_3			1	1	1	1	1	

تبدیل متن به بردار ویژگی . . .

- بردار فراوانی عبارت (TF: Term-Frequency) . . .
  - مثال: فاصله بین دو متن d1 و d2 (با فاصله کسینوسی)
    - بردار d1= (2, 1, 0, 0, 0, 0, 0, 0)
    - بردار d2= (1, 1, 0, 4 0, 0, 0, 1)

	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	2	1						
d_2	1	1		4				1
d_3			1	1	1	1	1	

$$cos(\theta) = \frac{\boldsymbol{d}_1 . \boldsymbol{d}_2}{|\boldsymbol{d}_1|\,|\boldsymbol{d}_2|} = \frac{2\times1+1\times1+0\times0+0\times4+0\times0+0\times0+0\times0+0\times1}{\sqrt{2^2+1^2}\times\sqrt{1^2+1^2+4^2+1^2}} = \frac{3}{\sqrt{5*19}} = 0.31$$

- فاصله همه سندها با همدیگر
  - نتایج متفاوت با بردار دودویی

	d_1	d_2	d_3
d_1	1	0.31	0
d_2	0.31	1	0.41
d_3	0	0.41	1

تبدیل متن به بردار ویژگی . . .

- **بردار فراوانی عبارت (TF: Term-Frequency)**
  - **مشکل: در صورت طولانی بودن یک متن (نسبت به دیگری) بردار به سمت متن طولانی بایاس می‌شود.**
  - **راه حل: نرمال کردن تعداد هر کلمه**
    - به تعداد کل کلمات در یک متن = بردار احتمال (Monogram)
      - N= تعداد کل کلمات سند i

تعداد تکرار کلمه j در سند i

جمع تکرار همه کلمه ها در سند i

$$TF_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{N} n_{i,k}}$$

- به بیشینه تعداد کلمات

$$TF_{i,j} = \frac{n_{i,j}}{\max_k(n_{i,k})}$$

- در مقیاس لگاریتم

$$TF_{i,j} = \log(1+n_{i,j})$$

- نرمال‌سازی ۰.۵

$$TF_{i,j} = 0.5 + 0.5\frac{n_{i,j}}{\max_k(n_{i,k})}$$

تبدیل متن به بردار ویژگی . . .

- **معکوس فراوانی سند (IDF: Inverse Document Frequency) . . .**
  - **اگر یک کلمه در تعداد کمی از سندها حضور داشته باشد، تفکیک بیشتری بین سندها ایجاد می‌کند (در مقایسه با کلماتی که در همه سندها وجود دارند).**
  - **برای کلمه jام (t_j) = نسبت تعداد کل سندها بر تعداد سندهایی که حاوی کلمه t_j است**

$$IDF_j = \log\left(\frac{N}{|\{i: t_j \in d_i\}|}\right)$$

تعداد کل سندها

تعداد سندهایی که حاوی کلمه t_j هستند

  - **بیانگر کمیاب بودن یک کلمه در مجموعه سندهاست.**
    - اگر یک کلمه در همه سندها حضور داشته باشد، مقدار IDF آن صفر است

تبدیل متن به بردار ویژگی . . .

- معکوس فراوانی سند (IDF: Inverse Document Frequency) . . .
  - مثال
    - یک پیکره با ۱.۰۰۰.۰۰۰ سند داریم
    - بنابراین

$$IDF_{sunday} = log_{10}\left(\frac{1.000.000}{1000}\right) = 3$$

تبدیل متن به بردار ویژگی . . .

- **معکوس فراوانی سند (IDF: Inverse Document Frequency)**
  - **اگر کلمه jام در هیچ سندی موجود نباشد، مخرج صفر می‌شود**
    - رفع مشکل با اضافه کردن عدد ۱ به مخرج

$$IDF_j = \log\left(\frac{N}{\left|\{\mathrm{i}: t_j \in d_i\}\right| + 1}\right)$$

- **نرمال‌سازی هموار**

$$IDF_j = \log\left(\frac{N}{\left|\{\mathrm{i}: t_j \in d_i\}\right|} + 1\right)$$

- **نرمال‌سازی بیشینه**

$$IDF_j = \log\left(1 + \frac{\max_k(N_k)}{N_j}\right)$$

  - که $N_j = \left|\{\mathrm{i}: t_j \in d_i\}\right|$ بیانگر تعداد سندهای حاوی کلمه j

تبدیل متن به بردار ویژگی . . .

- **فراوانی عبارت–معکوس فراوانی سند** (TF-IDF: Term Freq-Inverse Document Freq) . . .

  - برای هر کلمه، تعداد وقوع آن کلمه در متن در معکوس فراوانی سند ضرب می‌شود

$$TFIDF_{i,j} = TF_{i,j} \times IDF_j = \frac{n_{i,j}}{\sum_{k=1}^{N} n_{i,k}} \log\left(\frac{N}{\left|\{i: t_j \in d_i\}\right| + 1}\right)$$

برای کلمه j در سند i

  - نوعی دیگر (نرمال شده)

$$TFIDF_{i,j} = TF_{i,j} \times IDF_j = \left(0.5 + 0.5\frac{n_{i,j}}{\max_k(n_{i,k})}\right) \log\left(\frac{N}{\left|\{i: t_j \in d_i\}\right| + 1}\right)$$

  - از روش‌های بسیار رایج در پردازش متن

تبدیل متن به بردار ویژگی

- فراوانی عبارت–معکوس فراوانی سند (TF-IDF: Term Freq-Inverse Document Freq)

$$TFIDF_{i,j} = TF_{i,j} \times IDF_j = \frac{n_{i,j}}{\sum_{k=1}^{N} n_{i,k}} \log\left(\frac{N}{\left|\{i: t_j \in d_i\}\right|}\right)$$

- مثال
  - مجموعه متن (انگلیسی) Reuters با 806,791 سند
  - کلمه Insurance در 3,997 سند تکرار شده است
  - تعداد تکرار کلمه Insurance در چند نمونه سند

Term	Doc1	Doc2	Doc3
Insurance	3	0	1
# all Terms	100	79	83

- هدف: محاسبه $TFIDF(\text{"Insurance"}, Doc1)$

$$TF(\text{Insurance}, Doc1) = \frac{3}{100} = 0.03$$

بدون نرمال‌سازی ⟵ $TF(\text{Insurance}, Doc1) = 3$

$$IDF(\text{Insurance}, D) = log_e\left(\frac{806{,}791}{3{,}997}\right) = 5.31$$

$$TFIDF(\text{Insurance}, Doc1) = 0.03 \times 5.31 = 0.16$$

بدون نرمال‌سازی TF ⟵ $TFIDF(\text{Insurance}, Doc1) = 3 \times 5.31 = 16$

Hadi Veisi (h.veisi@ut.ac.ir)

تبدیل متن به بردار ویژگی . . .

آنتروپی نرمال شده

- نوعی از TF با نرمال‌سازی بر اساس آنتروپی
- در نظر گرفتن اهمیت هر کلمه در سندها
- بیانگر اهمیت کلمه iام در پیکره از نظر میزان رخداد در سندها

$$w_{ij} = (1 - \alpha_i) \frac{n_{i,j}}{\sum_{k=1}^{N} n_{k,j}}$$

$n_{i,j}$: تعداد تکرار کلمه iام در سند d_j

$\sum_{k=1}^{N} n_{k,j}$: تعداد کل همه کلمه ها در سند d_j

w_{ij}: آنتروپی نرمال شده کلمه iام در پیکره

$$\alpha_i = \frac{-1}{\log(N)} \sum_{j=1}^{N} \left(\frac{n_{i,j}}{N_i} log \frac{n_{i,j}}{N_i} \right)$$

N_i: تعداد کل کلمه iام در پیکره

- مقدار آنتروپی α_i عددی بین صفر و یک
  - نزدیک ۱ = رخداد کلمه در همه سندهای پیکره = کم اهمیت
  - نزدیک ۰ = رخداد در تعداد کمی از سندها= اهمیت زیاد

Hadi Veisi (h.veisi@ut.ac.ir)

معیارهای محاسبه شباهت . . .

- کسینوسی (Cosine)
  - برای دو بردار (رشته) d_1 و d_2

$$\frac{|d_1 \cap d_2|}{|d_1| \times |d_2|} \equiv \frac{d_1 . d_2}{|d_1|\,|d_2|}$$

ضرب داخلی (نقطه‌ای)

اندازه اقلیدسی بردار

- جاکارد (Jaccard)

$$\frac{|d_1 \cap d_2|}{|d_1 \cup d_2|} \equiv \frac{|d_1 \cap d_2|}{|d_1| + |d_2| - |d_1 \cap d_2|} \equiv \frac{d_1 . d_2}{|d_1| + |d_2| - d_1 . d_2}$$

- سورنسن–دایس (Sørensen–Dice)

$$\frac{2|d_1 \cap d_2|}{|d_1| + |d_2|} \equiv \frac{2 \times (d_1 . d_2)}{|d_1| + |d_2|}$$

- همپوشانی (Overlap)

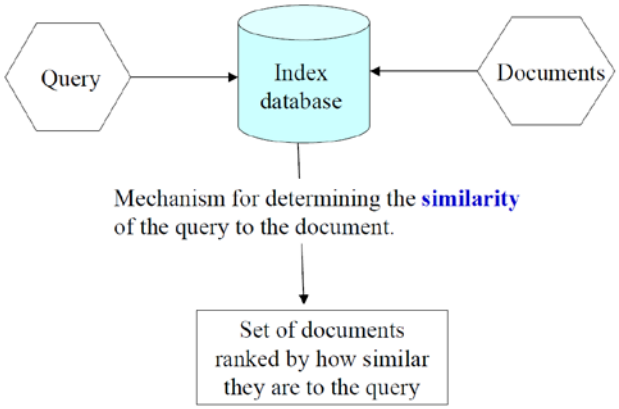
$$\frac{|d_1 \cap d_2|}{\min(|d_1|, |d_2|)} \equiv \frac{d_1 . d_2}{\min(|d_1|, |d_2|)}$$

معیارهای محاسبه شباهت . . .

- نیاز به استفاده از معیارهای شباهت مختلف در کاربردهای مختلف
  - چون هنوز هیچ‌کدام از معیارها در همه کاربردها بهینه نیستند
- استفاده از معیارهای شباهت برای محاسبه شباهت بین
  - دو رشته متنی یا دو بردار
    - دو سند (سرقت علمی)
    - یک سند و یک پرسش (سیستم‌های بازیابی اطلاعات، پرسش و پاسخ)
- استفاده از مقدار شباهت بدست آمده برای
  - صرف‌نظر کردن از برخی پاسخ‌ها (آنهایی که مقدار شباهتشان از مقدار آستانه کمتر است)
  - رتبه دادن به سندهای پاسخ (بر اساس مقدار شباهت)
- محاسبه فاصله بر اساس مقدار شباهت: Distance=1-Similarity

معیارهای محاسبه شباهت . . .

- در بازیابی اطلاعات
  - برای پرسش Q، تشابه آن با همه سندها محاسبه می‌شود
    - پاسخ = سندهایی که میزان شباهت آنها با پرسش از مقداری مشخص مانند ۰.۷ بیشتر باشد
  - رتبه‌بندی پاسخ‌ها: بر اساس مقدار شباهت



- در تشخیص سرقت علمی
  - محاسبه شباهت سند مشکوک به سرقت با همه سندهای موجود
    - پاسخ = سندهایی که میزان شباهت آنها با سند مشکوک از مقداری مشخص مانند ۰.۸ بیشتر باشد

معیارهای محاسبه شباهت

- مثال (در بازیابی اطلاعات)

query		
q	*ant dog*	
document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

	ant	bee	cat	dog	eel	fox	gnu	hog
q	1			1				
d_1	2	1						
d_2	1	1		4				1
d_3			1	1	1	1	1	

ویژگی TF و شباهت کسینوسی

	d_1	d_2	d_3
q	$2/\sqrt{10}$	$5/\sqrt{38}$	$1/\sqrt{10}$
	0.63	0.81	0.32

- رتبه‌بندی: d2, d1, d3

تبدیل متن به بردار ویژگی

- **وابسته به کاربرد است**
  - **برای استخراج کلمات کلیدی از متن: روش TF معیار خوبی است**
  - **برای تشخیص عنوان (موضوع) متن: معیار TFIDF معیار مناسبی است**
  - **برای تشخیص جنسیت نویسنده متن**
    - شمارش تعداد: رنگ‌ها، کل کلمات یکتا، کلمات یک/دو تکراره، کلمات رکیک، صفات مثبت/منفی، ضمایر
  - **برای عقیده کاوی**
    - شمارش تعداد صفات مثبت/منفی
  - **در کاربردهای دیگر**
    - تعداد نویسه‌ها، تعداد اعداد، تعداد علائم سجاوندی، میانگین طول کلمه (برحسب نویسه)، طول جمله (برحسب کلمه)، تعداد حروف ربط/اضافه، تعداد اسم/صفت/فعل، ...

تشابه‌یابی و بازیابی اطلاعات . . .

- میزان خوب بودن پاسخ سیستم
  - Precision: درصدی از پاسخ‌های بازگردانده شده توسط سیستم که مرتبط با پرسش کاربر است

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

  - Recall: درصدی از سندهای مرتبط در پایگاه داده که توسط سیستم به عنوان پاسخ بازگردانده شده‌اند

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

	truly relevant	truly irrelevant
retrieved	true positive (tp)	false positive (fp)
not retrieved	false negative (fn)	true negative (tn)

تشابه‌یابی و بازیابی اطلاعات . . .

ریشه‌یابی

- **می‌توان ریشه (Stem) مربوط به کلمات (Term) را استخراج کرد و شکل‌های مختلف صرفی یک کلمه را با هم یکی درنظر گرفت.**
  - مزیت: سندهایی که شامل صورت‌های مختلف صرفی یک کلمه هستند، بازیابی می شوند.
  - عیب: در نظر نگرفتن وجه تمایز صورت‌های مختلف صرفی یک کلمه دقت بازیابی را پایین می آورد.

ایست واژه‌ها (Stop Word)

- **از مجموعهٔ کلمات حذف می‌شوند**
  - مزیت: ایست واژگان تعداد رخداد بالا و بار معنایی کمی دارند و حذف آنها حجم محاسبات را کاهش می دهد و کارایی را بهبود می‌دهد.
  - عیب: جستجوی پرسش‌هایی که شامل ایست واژگان هستند، دقت پایین دارد.

تبدیل متن به بردار ویژگی: مشکل . . .

- در نظر نگرفتن وابستگی آماری کلمات در متن
  - بررسی مستقل واژه‌ها

- عدم توجه به معنی و مفهوم
  - مترادف (Synonym): کلمات مجزایی که معانی یکسانی دارند
    - فارسی: خودرو و اتومبیل
    - انگلیسی: big و large
    - باعث کاهش Recall می‌شود
  - چندمعنایی (Polysemy): کلماتی با یک صورت نوشتاری و معانی مجزا
    - فارسی: شیر (حیوان، خوراکی، وسیله مکانیکی) یا روان (جاری، روح)
    - انگلیسی: Bank (موسسه مالی، ساحل) یا Book (کتاب، رزرو کردن)
    - باعث کاهش Precision می‌شود

تبدیل متن به بردار ویژگی: مشکل

- محاسبات بالا
  - در کاربردهای واقعی پیکره‌ها بزرگ هستند
    - یک میلیارد کلمه
      - یک میلیون سند هر کدام حدود ۱۰۰۰ کلمه
    - تعداد واژه‌های یکتا = ۵۰۰ هزار کلمه
    - ماتریس وقوع (Term-Document)
      - ماتریسی ۵۰۰/۰۰۰ در ۱/۰۰۰/۰۰۰
      - ماتریس به شدت تنک است (بیشتر عناصر صفر هستند)

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **یکی از روش‌های جبری-آماری در تبدیل سندها/کلمات به بردار ویژگی**
  - **سعی در یافتن معنی پنهان کلمات در سندها**
  - **در یک سند، کلمات قابل مشاهده هستند ولی عنوان آن پنهان (Latent) است**

- **فرضیات**
  - کلماتی که به‌طور همزمان در یک سند (با موضوع مشخص) می‌آیند، از نظر معنایی به هم مرتبط هستند
  - سندهایی که دارای موضوع مشابهی هستند، حاوی کلمات مشابهی هستند

- **یک روش کاهش ابعاد بردارهای و یژگی**
  - متون و کلمات را به یک فضای معنایی مشترک تصویر می‌کند
  - مبتنی بر تجزیه مقادیر تکین (SVD)
  - استفاده از ایده روش تحلیل اجزای اصلی (PCA)
  - اسم دیگر Latent Semantic Indexing (LSI)

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **فرض کنید در پیکره متنی، داریم:**
    - تعداد M کلمه $W=\{w_1,w_2,\ldots,w_M\}$
    - تعداد N سند $D=\{d_1,d_2,\ldots,d_N\}$

- **گام ۱: ساخت ماتریس رخداد کلمات در سندها (ماتریس X)**
  - ماتریسی M×N است که عنصر x_{ij} از آن بیانگر شمارش وزن‌دار کلمه iاُم (w_i) در سند jاُم (d_j) است
    - عناصر ماتریس می‌توانند TF یا TFIDF یا ویژگی‌های مشابه باشند
  - هر ردیف ماتریس X بیانگر بردار یک کلمه در همه سندهاست
  - هر ستون، بیانگر بردار یک سند است

$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} x_{1,1} & \ldots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \ldots & x_{m,n} \end{bmatrix} \quad \mathbf{d}_j \downarrow$$

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **گام ۲: تجزیه ماتریس تنک X با SVD . . .**
    - X = ماتریس M×N «کلمه-سند» است که حاوی فراوانی (نرمال شده) یا بردار دودویی کلمات در سندها است
    - $B=XX^T$ ماتریس M×M «کلمه-کلمه» است
      - اگر کلمه i و کلمه j با هم در b سند مشترک آمده باشند، آنگاه B(i,j)=b (اگر محتوای X بیانگر بردار دودویی باشند)
    - $C=X^TX$ ماتریس N×N «سند-سند» است
      - اگر سند i و سند j دارای c کلمه مشترک باشند، آنگاه C(i,j)=c (اگر محتوای X بیانگر بردار دودویی باشند)
    - ماتریس‌های B و C مربعی و متقارن هستند
    - می‌توان ماتریس X را با روش SVD به صورت زیر تجزیه کرد

$$X = U\Sigma V^T$$

      - U = ماتریسی است که حاوی «بردارهای ویژه» ماتریس B است
      - V = ماتریسی است که حاوی «بردارهای ویژه» ماتریس C است
      - Σ = ماتریسی قطری است که عناصر قطر اصلی آن جذر «مقادیر ویژه» ماتریس B (و C) است
        - مقادیر ویژه ماتریس های B و C با هم برابرند (تعداد M-N مقدار ویژه صفر است)

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **گام ۲: تجزیه ماتریس تنک X با SVD**

$$X = U\Sigma V^T$$

$$\underset{(\mathbf{t}_i^T)\rightarrow}{\begin{matrix} X \\ (\mathbf{d}_j) \\ \downarrow \end{matrix}} \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} = \underset{(\hat{\mathbf{t}}_i^T)\rightarrow}{} \begin{matrix} U \\ \end{matrix} \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} & \dots & \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{matrix} \Sigma \\ \end{matrix} \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{matrix} V^T \\ (\hat{\mathbf{d}}_j) \\ \downarrow \end{matrix} \begin{bmatrix} [& &] \\ & \mathbf{v}_1 & \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}$$

  - مقادیر ویژه ماتریس Σ بیانگر اهمیت هر کدام از بردارهای ویژه است

  - **گام ۳: کاهش بعد**
    - تعداد محدودی از مقادیر ویژه، اعداد بزرگ و مابقی اعداد کوچک (نزدیک صفر) هستند
    - از میان مقادیر ویژه، فقط k مقدار بزرگ‌تر و بردارهای ویژه معادل را حفظ می‌کنیم
    - مرتب کردن بر اساس بزرگی مقدار ویژه
    - کاهش بعد به k بعد (k خیلی کوچک تر از M و N)
      - تعداد k معادل تعداد «مفاهیم پنهان» موجود در کلمات و سندهاست

$$X_{M\times N} \cong U_{M\times k}\Sigma_{k\times k}V^t_{k\times N}$$

$$A_k = U_k\Sigma_k V_k^t$$

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **گام ۴: محاسبه بردار سند یا بردار کلمه**
    - بردار هر کلمه: ردیف‌های ماتریس $U_{M\times k}\Sigma_{k\times k}$
      - هر بردار k بعدی است
      - کلماتی که بردارهای آن‌ها در زیرفضای کاهش‌بعدیافته به هم نزدیک می‌باشند همان کلماتی هستند که از نظر معنایی به هم مرتبط هستند
    - بردارهای هر سند: ستون‌های ماتریس $\Sigma_{k\times k}V^t_{k\times N}$
      - هر بردار k بعدی است
      - سندهایی با بردار مشابه دارای موضوع مشابه هستند
    - از بردارها بدست آمده (برای کلمات یا سندها) می‌توان در کاربردهای مختلف مانند دسته‌بندی کلمات و متون استفاده کرد

تبدیل متن به بردار ویژگی . . .

- **تحلیل معنایی پنهان (LSA: Latent Semantic Analysis) . . .**
  - **محاسبه بردار سند برای اسناد جدید (خارج از مجموعه آموزش $X_{M\times N}$)**
    - سند جدید = d_{new}

    - گام ۱: محاسبه شمارش وزن‌دار کلمه iاُم (w_i) در سند (d_{new}) با همان واژگان $W=\{w_1,w_2,\ldots,w_M\}$
      - بردار حاصل، یک بردار ستونی M بعدی است
      - استفاده از همان روشی که ماتریس $X_{M\times N}$ بر اساس آن ساخته شده است (TF، TFIDF یا آنتروپی نرمال شده)

    - گام ۲: سند تبدیل شده برابر است با $\hat{d}_{new}=\Sigma_k^{-1}U_k^t d_{new}$
      - معکوس ماتریس قطری Σ_k با معکوس کردن عناصر روی قطر اصلی قابل محاسبه است

    - در بازیابی اطلاعات، با آمدن پرسش q، باید این تبدیل بر روی پرسش (به عنوان یک سند جدید) اعمال شود

تبدیل متن به بردار ویژگی . . .

- تحلیل معنایی پنهان (LSA): مثال . . .
  - ۵ سند و ۸ کلمه
    - مقادیر ماتریس کلمه-سند بیانگر TF هستند

- گام ۱: ماتریس کلمه-سند X

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

- گام ۲: تجزیه $X = U\Sigma V^T$

$$\Sigma = \begin{bmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{bmatrix}$$

تبدیل متن به بردار ویژگی . . .

- تحلیل معنایی پنهان (LSA): مثال . . .
  - گام ۳: کاهش بعد $A_k = U_k \Sigma_k V_k^t$
    - از ۵ مقدار ویژه، ۲ مقدار بزرگ‌تر را نگه می‌داریم، یعنی k=2

$$U_2 = \begin{array}{r} romeo \\ juliet \\ happy \\ dagger \\ live \\ die \\ free \\ new\text{-}hampshire \end{array} \begin{bmatrix} -0.396 & 0.280 \\ -0.314 & 0.450 \\ -0.178 & 0.269 \\ -0.438 & 0.369 \\ -0.264 & -0.346 \\ -0.524 & -0.246 \\ -0.264 & -0.346 \\ -0.326 & -0.460 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix}$$

$$V_2^t = \begin{bmatrix} -0.311 & -0.407 & -0.594 & -0.603 & -0.143 \\ 0.363 & 0.541 & 0.200 & -0.695 & -0.229 \end{bmatrix}$$

  - گام ۴: محاسبه بردار سند یا بردار کلمه
    - بردار هر کلمه: ردیف‌های ماتریس $U_{M\times k}\Sigma_{k\times k}$
    - بردارهای هر سند: ستون‌های ماتریس $\Sigma_{k\times k}V_{k\times N}^t$

تبدیل متن به بردار ویژگی . . .

- تحلیل معنایی پنهان (LSA): مثال . . .
  - گام ۴: محاسبه بردار سند یا بردار کلمه
    - بردار هر کلمه: ردیف‌های ماتریس $U_{M\times k}\Sigma_{k\times k}$

$$romeo = \begin{bmatrix} -0.905 \\ 0.563 \end{bmatrix}, \; juliet = \begin{bmatrix} -0.717 \\ 0.905 \end{bmatrix}, \; happy = \begin{bmatrix} -0.407 \\ 0.541 \end{bmatrix}, \; dagger = \begin{bmatrix} -1.001 \\ 0.742 \end{bmatrix}$$

$$live = \begin{bmatrix} -0.603 \\ -0.695 \end{bmatrix}, \; die = \begin{bmatrix} -1.197 \\ -0.494 \end{bmatrix}, \; free = \begin{bmatrix} -0.603 \\ -0.695 \end{bmatrix}, \; new\text{-}hampshire = \begin{bmatrix} -0.745 \\ -0.925 \end{bmatrix}$$

    - بردارهای هر سند: ستون‌های ماتریس $\Sigma_{k\times k}V^t_{k\times N}$

$$d_1 = \begin{bmatrix} -0.711 \\ 0.730 \end{bmatrix}, \; d_2 = \begin{bmatrix} -0.930 \\ 1.087 \end{bmatrix}, \; d_3 = \begin{bmatrix} -1.357 \\ 0.402 \end{bmatrix}, \; d_4 = \begin{bmatrix} -1.378 \\ -1.397 \end{bmatrix}, \; d_5 = \begin{bmatrix} -0.327 \\ -0.460 \end{bmatrix}$$

Hadi Veisi (h.veisi@ut.ac.ir)

تبدیل متن به بردار ویژگی . . .

$$romeo = \begin{bmatrix} -0.905 \\ 0.563 \end{bmatrix}, \; juliet = \begin{bmatrix} -0.717 \\ 0.905 \end{bmatrix}, \; happy = \begin{bmatrix} -0.407 \\ 0.541 \end{bmatrix}, \; dagger = \begin{bmatrix} -1.001 \\ 0.742 \end{bmatrix}$$

$$live = \begin{bmatrix} -0.603 \\ -0.695 \end{bmatrix}, \; die = \begin{bmatrix} -1.197 \\ -0.494 \end{bmatrix}, \; free = \begin{bmatrix} -0.603 \\ -0.695 \end{bmatrix}, \; new\text{-}hampshire = \begin{bmatrix} -0.745 \\ -0.925 \end{bmatrix}$$

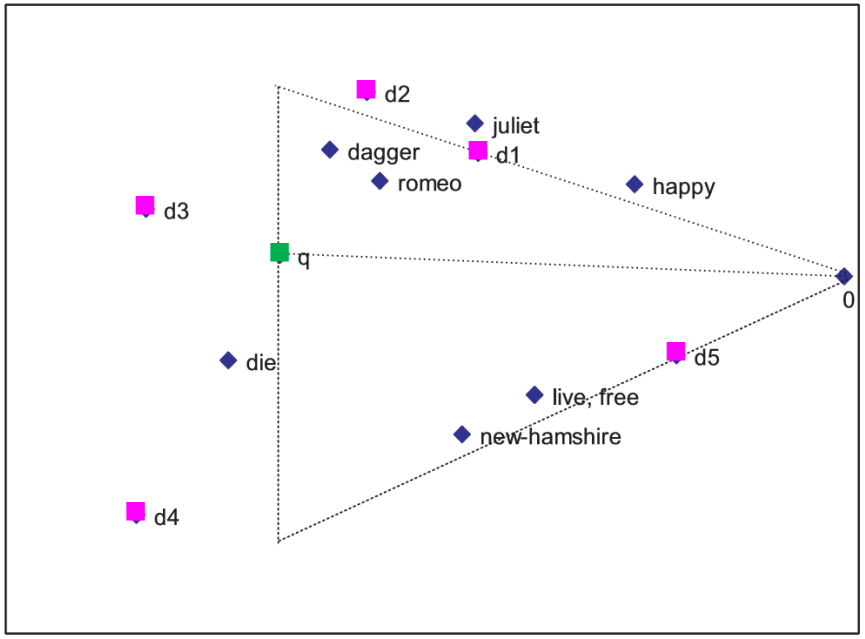
تحلیل معنایی پنهان (LSA): مثال

- گام (استفاده): محاسبه شباهت پرسش «die, dagger» به اسناد
  - محاسبه بردار پرسش

$$q = \frac{\begin{bmatrix} -1.197 \\ -0.494 \end{bmatrix} + \begin{bmatrix} -1.001 \\ 0.742 \end{bmatrix}}{2} = \begin{bmatrix} -1.099 \\ 0.124 \end{bmatrix}$$

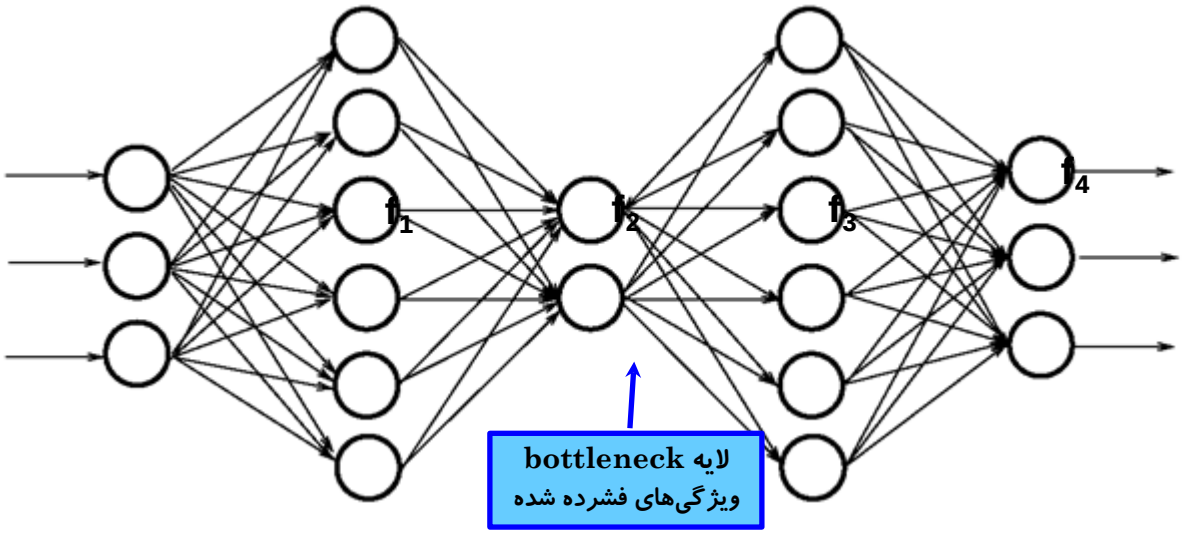
  - محاسبه شباهت کسینوسی با همه سندها
    - محاسبه $\frac{d_i \cdot q}{|d_i||q|}$ بین همه بردار سندها با بردار پرسش
    - سند d_1 نزدیک‌تر از d_5 به q است
      - Romeo و Juliet با dagger کشته شده‌اند
    - سند d_1 مقدار کمی نزدیک‌تر از d_2 به q است
      - سند d_1 حاوی کلمات مرتبط Romeo و Juliet با همدیگر است، هرچند d_2 حاوی یکی از کلمات پرسش است (پاسخ شبیه پاسخ احتمالی انسان است)

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1



تبدیل متن به بردار ویژگی

- سایر روش‌ها
  - **تحلیل معنایی پنهان احتمالاتی** (PLSA: Probabilistic LSA)
    - نسخه بهبود یافته PLSA
    - مرتبط کردن متغیرهای پنهان (عنوان) با متغیرهای قابل‌مشاهده (سندها و کلمات)
  - **مبتنی بر شبکه عصبی**
    - قابل استفاده هم برای استخراج بردار سند و هم برای بردار کلمه
      - Word2Vec
      - FastText



تبدیل متن به بردار ویژگی: کلمات . . .

- روش‌های بیان شده برای تبدیل یک سند به یک بردار بودند
- تبدیل کلمات به بردار
  - در برخی کاربردها نیاز است هر واژه به یک بردار تبدیل شود
    - در POS Tagging: استفاده از بردار کلمات (خود کلمه و کلمات قبلی)
  - بردار دودیی: تبدیل هر کلمه به یک بردار دودیی
    - هر کلمه معادل یک بردار N بعدی است: کل کلمات موردنظر در این حالت 2^N است
      - برای کد کردن ۱۰۲۴ کلمه، ۱۰ بیت کافیست (هر کلمه یک بردار ۱۰ بعدی)

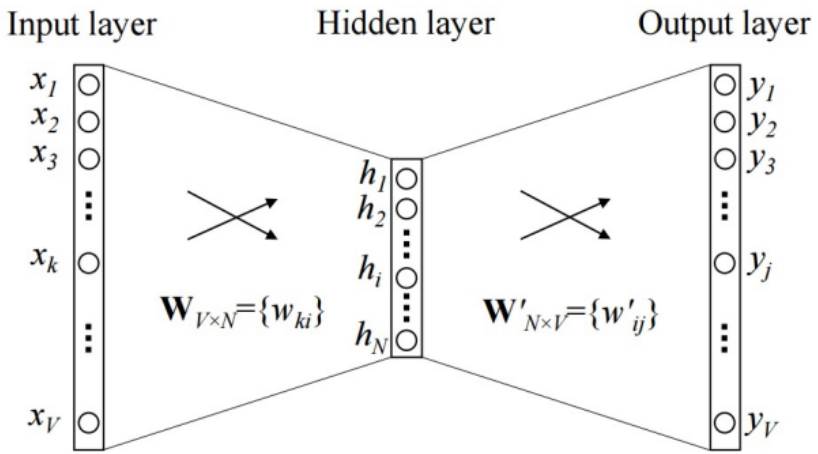
شماره کلمه	کلمه	بردار کلمه									
۱	آب	0	0	0	0	0	0	0	0	0	1
۲	آبرو	0	0	0	0	0	0	0	0	1	0
..	...	...									
۱۰۲۳	ویروس	1	1	1	1	1	1	1	1	1	1

  - بردارهای حاوی معنی: روش‌های مبتنی بر شبکه عصبی
    - یادگیری ارتباط نحوی و معنایی کلمات از روی داده‌ها

motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0] AND
hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0

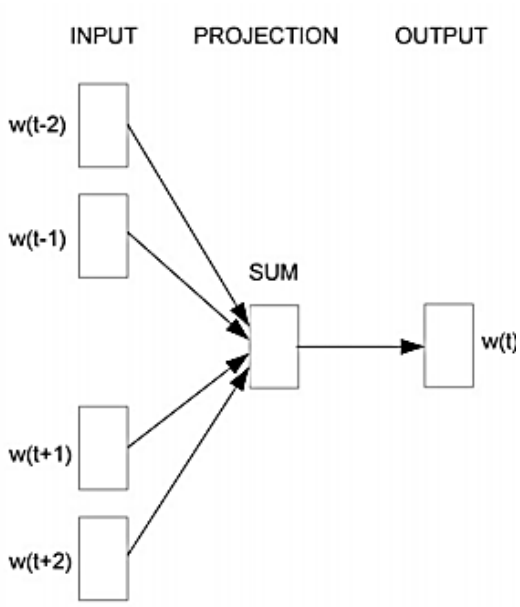
تبدیل متن به بردار ویژگی: کلمات . . .

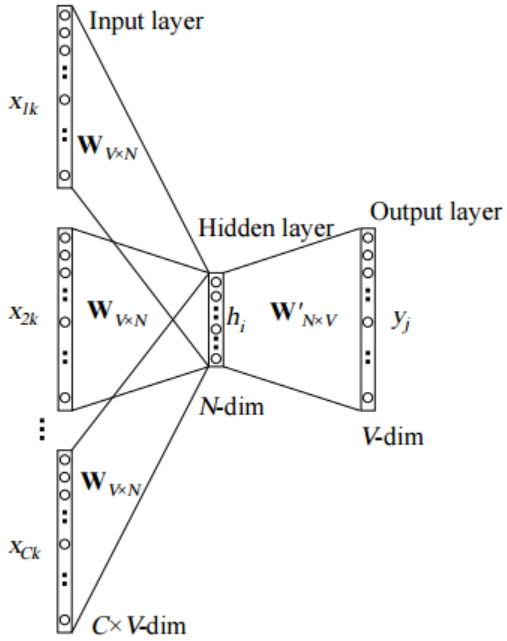
- بردار کلمات با شبکه عصبی . . .
  - در نظر گرفتن یک بردار چند بعدی (مثلا ۲۰۰ بعد)
  - مقدار اولیه: تصادفی
  - آموزش بردار کلمه با پیکره متنی و در نظر گرفتن بافت کلمه (کلمات اطراف)
  - حالت ساده: دادن یک کلمه به ورودی شبکه و پیش بینی کلمه بعد از آن
    - مشابه دوتایی (Bi-gram)



تبدیل متن به بردار ویژگی: کلمات . . .

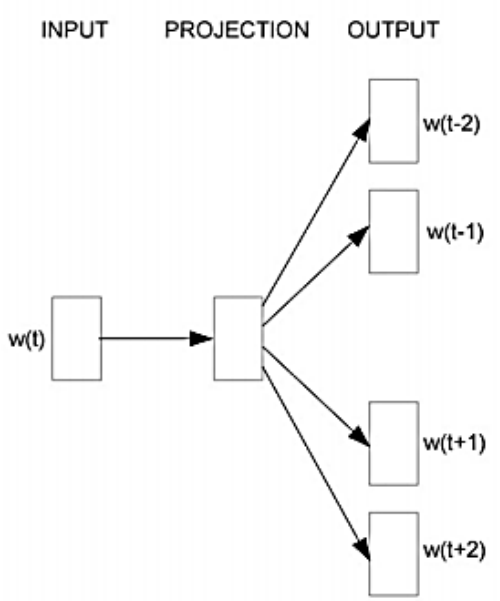
- بردار کلمات با شبکه عصبی . . .
  - آموزش بردار کلمه با پیکره متنی و در نظر گرفتن بافت کلمه (کلمات اطراف)
  - با پنجره کلمات بزرگ‌تر (بیش از یک کلمه اطراف)
    - روش کیسه کلمات پیوسته (CBOW: continuous bag-of-words)
      - پیش بینی کلمه با در نظر دو (چند) کلمه قبل و دو (چند) کلمه بعد





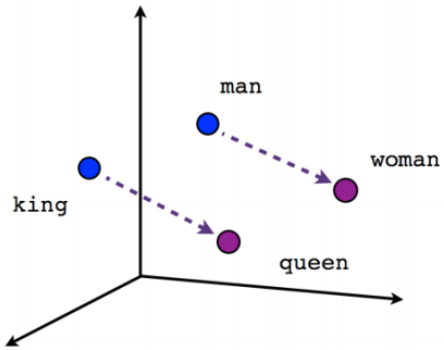
تبدیل متن به بردار ویژگی: کلمات . . .

- بردار کلمات با شبکه عصبی . . .
  - آموزش بردار کلمه با پیکره متنی و در نظر گرفتن بافت کلمه (کلمات اطراف)
  - با پنجره کلمات بزرگ‌تر(بیش از یک کلمه اطراف)
    - روش پرش چندتایی (skip-gram)
      - پیش بینی کلمات (دو یا چند کلمه قبل و دو یا چند کلمه بعد) از روی کلمه فعلی
        - در عمل احتمال همه کلمات

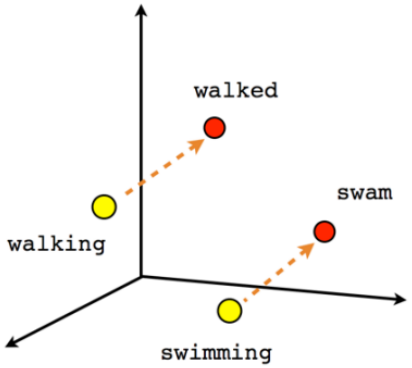


تبدیل متن به بردار ویژگی: کلمات . . .

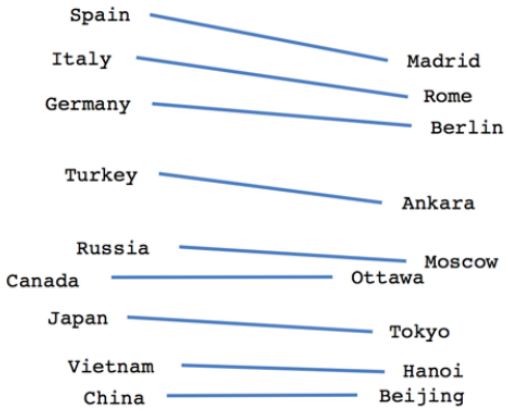
- بردار کلمات با شبکه عصبی . . .
  - در نظر گرفتن ارتباط بین کلمات (معنا) در یافتن مقدار بردار کلمه: بردارهای با معنی
  - بردار کوچک و محاسبات سریع



Male-Female



Verb tense



Country-Capital

تبدیل متن به بردار ویژگی: کلمات . . .

- بردار کلمات با شبکه عصبی . . .

$$X_{shirt} - X_{clothing} \approx X_{chair} - X_{furniture}$$

$$X_{apple} - X_{apples} \approx X_{car} - X_{cars} \approx X_{family} - X_{families}$$

- **ترکیب بردارها**
  - اصفهان+ رودخانه ➔زاینده رود

- **جستجو کنید:** Word2vec ،word vector representation ،GloVe

تبدیل متن به بردار ویژگی

- **بردار جمله/پاراگراف/متن از روی بردار کلمات (با شبکه عصبی)**
  - **ترکیب بردارهای کلمات جمله/پاراگراف/متن**
    - با میانگین‌گیری!
    - با روش‌های غیرخطی مانند شبکه عصبی
  - **استفاده در تشابه‌یابی**
    - مقایسه دو بردار برای دو جمله/پاراگراف/متن با میعارهای تشابه مانند فاصله کسینوسی
      - رتبه دوم مسابقه PAN 2016 فارسی

Rank	Team	Plagdet	Granularity	Precision	Recall
1	Fatemeh Mashhadi, Mehrnoush Shamsfard Shahid Beheshti University, NLP Research Lab	**0.92204**	1.00146	0.92688	0.91919
2	Hadi Veisi, Kayvan Bijari, Kiarash Zahirnia, Erfaneh Gharavi University of Tehran, Data & Signal processing Lab	**0.90593**	1	0.95927	0.85820
3	Mozhgan Momtaz, Kayvan Bijari, Davood Heidarpour University of Tehran, COIN Lab	**0.87103**	1	0.89258	0.85049
4	Mahdi Niknam University of Qom	**0.83015**	1.03968	0.92034	0.79602
5	Faezeh Esteki, Faramarz Safi Esfahani Najafabad Branch, Islamic Azad University	**0.80083**	1.0	0.93337	0.70124
6	Alireza Talebpour, Mohammad Shirzadi, Zahra Aminolroaya, Mohammad Adibi, Ahmad Mahmoudi-Aznaveh Shahid Beheshti University, Content lab /cyberspace research institute	**0.77496**	1.22759	0.96383	0.83615