Introduction to

Information Retrieval

فصل شش: نمره گذاری، وزن دهی عبارات و مدل فضای بردار

علی قنبری سرخی

بازیابی رتبه بندی شده

- تاکنون با شاخص های آشنا شدیم که پرس و جوهای بولی را پشتیبانی می کردند
  - یک سند با یک پرس و جو تطبیق داشت یا نداشت.

- در مجموعه های بزرگ اسناد، تعداد نهایی اسناد منطبق می تواند بسیار بیشتر از آن چیزی باشد که کاربر انسانی می تواند آن را غربال کند.
  - بنابراین، ضروری است که موتور جستجو اسناد منطبق با پرس و جو را بر اساس رتبه مرتب کند.
  - برای این منظور، موتور جستجو برای هر سند منطق نمره ای را نسبت به پرس و جو محاسبه می‌کند.

بازیابی رتبه بندی شده

- ایده اصلی این فصل:
  - شاخص های پارامتری و ناحیه ای که دو هدف را دنبال می کند معرفی می کنیم.
  - هدف اول: اجازه می دهد اسناد را با فراداده هایی، مانند زبانی که اسناد با آن نگارش شده شاخص گذاری و بازیابی کنیم.
  - هدف دوم: ابزار ساده برای نمره گذاری (از اینرو رتبه بندی) اسناد در پاسخ به پرس و جو فراهم می کنند.
  - ایده وزن دهی اهمیت یک عبارت در سند را بر اساس آمار وقوع آن عبارت بررسی می کنیم.
  - نمره ای برای یک پرس و جو و هر سند محاسبه می کنیم.

شاخص های ناحیه ای و پارامتری

- تا اینجا سند را به عنوان دنباله ای از عبارات در نظر گرفته ایم.
  - در حقیقت، اکثر اسناد ساختار اضافه ای دارند.
  - در اسناد دیجیتال، اطلاعات فراداده های خاصی که مرتبط به سند باشد کدگذاری شده است.
  - منظور از فراداده، صورت خاصی از داده در مورد یک سند مانند، مولفان، عنوان و تاریخ انتشار است.

  - این فراداده عموما شامل خصایصی مانند تاریخ ایجاد، فرمت سند، همچنین نام مولف و احتمالا عنوان سند است.
  - مقادیر ممکن برای یک خصیصه باید محدود و متناهی باشد.
    - برای مثال، مجموعه تمامی تاریخ های تالیف

شاخص های ناحیه ای و پارامتری

- پرس و جو : "اسنادی را بیابید که توسط ویلیام شکسپیر در ۱۶۰۱ نوشته شده و شامل اصطلاحی مانند alas poor Yorick باشد"
  - پردازش پرس و جو:
    - اشتراک پست ها
    - شاخص های پارامتری (مثلا تاریخ ایجاد)
  - این امر به ما اجازه می دهد تا تنها اسنادی را انتخاب کنیم که تاریخ مشخصی را در پرس و جو تطبیق می دهد.
- ناحیه ها: مانند خصایص هستند با این تفاوت که محتوای ناحیه می تواند متن آزاد اختیاری باشد.
  - خصیصه ممکن است مجموعه نسبتا کوچکی از مقادیر را در برگیرد ناحیه می تواند یک متن اختیاری بی حد باشد.
    - برای نمونه سند و خلاصه سند دو ناحیه هستند.
    - می توانیم شاخص وارونه‌ی جداگانه‌ای برای هر ناحیه از سند بسازیم
    - تا بتوانیم پرس و جوهایی "اسنادی را بیابید merchant در عنوان و william در لیست مولفان و اصطلاح gentle rain در متن آن باشد"

5

نمره گذاری وزن‌دار ناحیه‌ای

- پرس و جو بولی q و سند d، نمره گذاری وزندار ناحیه‌ای، به جفت (q,d) نمره ای بازهی [0,1] با محاسبه ی ترکیب خطی نمره های ناحیه اختصاص می دهد.
  - در آن هر ناحیه از سند به یک مقدار بولی اشاره دارد.
  - اگر l ناحیه داشته باشیم که $g_1, \dots g_l \in [0,1]$ و $\sum_{i=1}^{l} g_i = 1$ باشد.
  - برای $1 \leq i \leq l$، s_i نمره ی بولی که تطبیق (یا عدم تطبیق) بین q و iامین ناحیه را نشان می دهد.
  - برای مثال نمره بولی برای یک ناحیه می تواند ۱ باشد اگر تمامی عبارات پرس و جو در آن ناحیه رخ دهد در غیر این صورت ۰ خواهد بود.
  - نمره وزن دار ناحیه (یا بازیابی بولی رتبه بندی شده) به صورت زیر تعریف می شود.
  - $\sum_{i=1}^{l} g_i s_i$

یادگیری وزن‌ها

- چگونه وزن g_i را برای نمره گذاری و وزندار ناحیه ای تعیین کنیم؟
  - توسط متخصص

- براساس مثال های آموزشی که به صورت ویراستاری شده‌اند یاد گرفته شود

وزن دهی و فراوانی عبارت

- نمره گذاری بر اساس پرس و جو در یک ناحیه از سند موجود است یا خیر

- گام منطقی بعدی
  - سند یا ناحیه ای که که یک عبارت پرس و جو را بارها دربردارد بیشتر به آن پرس و جو پرداخته و از اینرو نمره بالاتری دریافت میکند.
  - برای این هدف، برای هر عبارت در سند وزنی را انتخاب می کنیم که به تعداد وقوع‌های آن عبارت در سند بستگی دارد.
- نمره بین عبارت پرس و جو t و سند d بر اساس وزن t در d محاسبه می‌شود.
- ساده ترین روش، انتساب وزن برابر با تعداد وقوع‌های عبارت t در d است. به این روش فراوانی عبارت گفته می شود.
- فراموایین عبارت را با $tf_{t,f}$ نشان داده می شود.

8

فراوانی وارونه سند

- فراوانی عبارت مشکل عمده ای دارد:
  - تمامی عبارات از اهمیت یکسانی برخوردارند. در حقیقت عبارت معین، در تعیین مرتبط بودن کم اثر یا بی اثر است.
    - برای نمونه، اسناد مرتبط به صنعت خودکار، عبارت "خودکار" را در هر سند دارا است.
- ایده دیگر:
  - وزن های عبارت را با فراوانی مجموعه تنظیم کنیم.
  - فراوانی مجموعه: تعداد کل وقوع عبارت در مجموعه است.
  - ایده این است که وزن tf عبارت را با فاکتورهای که با فراوانی مجموعه افزایش می‌یابد کاهش دهیم.
  - فراوانی سند (df_t) را به کار بریم که به صورت تعداد اسنادی در مجموعه که شامل عبارت t هستند تعریف شود.

9

فراوانی وارونه سند

- فراوانی وارونه سند

$$idf_t = log \frac{N}{df_t}$$

- N تعداد اسناد در یک مجموعه

10

Binary incidence matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	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
. . .						

Each document is represented as a binary vector $\in \{0, 1\}^{|V|}$.

11

Binary incidence matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	157	73	0	0	0	1
BRUTUS	4	157	0	2	0	0
CAESAR	232	227	0	2	1	0
CALPURNIA	0	10	0	0	0	0
CLEOPATRA	57	0	0	0	0	0
MERCY	2	0	3	8	5	8
WORSER	2	0	1	1	1	5
. . .						

Each document is now represented as a count vector $\in \mathbb{N}^{|V|}$.

12

Examples for idf

- Compute idf_t using the formula: $\text{idf}_t = \log_{10} \frac{1{,}000{,}000}{\text{df}_t}$

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

Collection frequency vs. Document frequency

word	collection frequency	document frequency
INSURANCE	10440	3997
TRY	10422	8760

- Collection frequency of *t*: number of tokens of *t* in the collection
- Document frequency of *t*: number of documents *t* occurs in
- Why these numbers?
- Which word is a better search term (and should get a higher weight)?
- This example suggests that df (and idf) is better for weighting than cf (and "icf").

14

وزن دهی tf-idf

- با ترکیب تعاریف فراوانی عبارت و فراوانی وازونه سند، وزن مرکبی را برای هر عبارت در هر سند ایجاد می کنیم.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

- این روش وزن دهی، به عبارت t در سند d وزنی می‌دهد که:
  - بالاترین است زمانیکه t به دفعات زیاد درون تعداد اندکی از اسناد رخ دهد.
    - قدرت تمایز بالایی برای آن اسناد ایجاد می کند.
  - پایینتر است زمانیه عبارت به دفعات کمتری در یک سند رخ دهد و یا در بسیاری از اسناد رخ داده باشد.
  - کمترین است زمانیکه عبارت در تمامی اسناد وجود داشته باشد.

وزن دهی tf-idf

- در اینجا هر سند به عنوان یک بردار یا مولفه در نظر گرفته می شود. که متناظر با هر عبارت در لغت نامه است و این سند را با وزنی که برای هر مولفه بدست میآید در نظر خواهیم داشت.
- برای عبارات لغت نامه که در سند رخ نمی دهند، این وزن صفر خواهد بود.
- نمره هم پوشانی
  - نمره یک سند d، روی تمامی عبارات پرس و جو، مجموعه تعداد دفعاتی است که هر عبارت در پرس و جو d رخ داده است.
  - تصحیح این ایده: تعداد وقوع هر عبارت پرس و جو t را در d جمع نکرده و و در عوض وزن tf-idf هر عبارت در d را جمع ببندیم.

$$Score(q,d) = \sum_{t \in q} tf - idf_{t,d}$$

Exercise: Term, collection and document frequency

Quantity	Symbol	Definition
term frequency	$tf_{t,d}$	number of occurrences of *t* in *d*
document frequency	df_t	number of documents in the collection that *t* occurs in
collection frequency	cf_t	total number of occurrences of *t* in the collection

- Relationship between df and cf?
- Relationship between tf and cf?
- Relationship between tf and df?

17

Binary incidence matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	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
. . .						

Each document is represented as a binary vector $\in \{0, 1\}^{|V|}$.

18

Count matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	157	73	0	0	0	1
BRUTUS	4	157	0	2	0	0
CAESAR	232	227	0	2	1	0
CALPURNIA	0	10	0	0	0	0
CLEOPATRA	57	0	0	0	0	0
MERCY	2	0	3	8	5	8
WORSER	2	0	1	1	1	5
. . .						

Each document is now represented as a count vector $\in \mathbb{N}^{|V|}$.

19

Binary → count → weight matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	5.25	3.18	0.0	0.0	0.0	0.35
BRUTUS	1.21	6.10	0.0	1.0	0.0	0.0
CAESAR	8.59	2.54	0.0	1.51	0.25	0.0
CALPURNIA	0.0	1.54	0.0	0.0	0.0	0.0
CLEOPATRA	2.85	0.0	0.0	0.0	0.0	0.0
MERCY	1.51	0.0	1.90	0.12	5.25	0.88
WORSER	1.37	0.0	0.11	4.15	0.25	1.95
. . .						

Each document is now represented as a real-valued vector of tf idf weights $\in R^{|V|}$.

20

مدل فضای بردار برای نمره گذاری

- نمایش مجموعه‌ی اسناد، به عنوان بردارهایی در یک فضای متعارف بردار، مدل فضای بردار نامیه می شود.
- این مدل برای یک میزابان عملیات بازیابی اطلاعات شامل نمره گذاری اسناد در یک پزس و جو، دسته بندی اسناد و خوشه بندی اسناد ضروری است.
- ابتدا، ایده های اساسی نمره گذاری فضای بردار شرح داده می شود.
- یک گام محوری در نظر گرفتن پرس وجوها به عنوان بردار در فضای بردار مشابه با مجموعه اسناد است.

ضرب نقطه ای

- منظور از $\overrightarrow{V}(d)$ برداری است که از سند d مشتق شده و در آن برای هر عبارت لغت نامه یک مولفه وجود دارد.
- یک مجموعه اسناد به عنوان مجموعه ی بردارها در یک فضای بردار نشان داده می شود که در آن یک محور برای هر عبارت وجود دارد.
  - در این نمایش، ترتیب نسبی عبارات در هر سند را از دست می دهد.

- محاسبه شباهت دو سند در فضای برداری:
- روش اول: بزرگی تفاوت بین دو بردار سند را محاسبه کنیم:
- مشکل اساسی: دو سند با محتوای مشابه می توانند تفاوت برداری قابل توجهی داشته باشند، چون به راحتی ممکن است یکی از دیگیری طولانی تر باشد.
- بنابراین توزیع نسبی عبارارت ممکن است در دو سند یکسان بوده اما فراوانی یک عبارت یکی ممکن است بزرگتر باشد.

22

ضرب نقطه ای

- برای جبران تاثیر طول سند، روش استاندارد اندازه گیری شباهت بین دو سند d1 و d2 محاسبه ی شاهت کسینوسی نمایش برداری $\overrightarrow{V}(d_1)$ و $\overrightarrow{V}(d_2)$ است:

$$sim(d_1, d_2) = \frac{\vec{v}(d_1).\vec{v}(d_2)}{|\vec{v}(d_1)||\vec{v}(d_2)|}$$

- صورت کسر نشان دهنده ضرب نقطه ای بردار ها می باشد.
- مخرج کسر ضرب طول اقلیدسی انها است.

Length normalization

- How do we compute the cosine?
- A vector can be (length-) normalized by dividing each of its components by its length – here we use the L_2 norm:
  $||x||_2 = \sqrt{\sum_i x_i^2}$
- This maps vectors onto the unit sphere . . .
- . . . since after normalization: $||x||_2 = \sqrt{\sum_i x_i^2} = 1.0$
- As a result, longer documents and shorter documents have weights of the same order of magnitude.
- Effect on the two documents *d* and *d'* (*d* appended to itself) from earlier slide: they have identical vectors after length-normalization.

24

Cosine similarity between query and document

$$\cos(\vec{q},\vec{d}) = \text{SIM}(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

- q_i is the tf-idf weight of term *i* in the query.
- d_i is the tf-idf weight of term *i* in the document.
- $|\vec{q}|$ and $|\vec{d}|$ are the lengths of $\vec{q}$ and $\vec{d}$.
- This is the cosine similarity of $\vec{q}$ and $\vec{d}$ . . . . . or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$.

Cosine for normalized vectors

- For normalized vectors, the cosine is equivalent to the dot product or scalar product.

$$\cos(\vec{q}, \vec{d}) = \vec{q} \cdot \vec{d} = \sum_i q_i \cdot d_i$$

  - (if $\vec{q}$ and $\vec{d}$ are length-normalized).

26

Cosine: Example

How similar are these novels? SaS: Sense and Sensibility PaP: Pride and Prejudice WH: Wuthering Heights

term frequencies (counts)

term	SaS	PaP	WH
AFFECTION	115	58	20
JEALOUS	10	7	11
GOSSIP	2	0	6
WUTHERING	0	0	38

27

Cosine: Example

term frequencies (counts)

term	SaS	PaP	WH
AFFECTION	115	58	20
JEALOUS	10	7	11
GOSSIP	2	0	6
WUTHERING	0	0	38

log frequency weighting

term	SaS	PaP	WH
AFFECTION	3.06	2.76	2.30
JEALOUS	2.0	1.85	2.04
GOSSIP	1.30	0	1.78
WUTHERING	0	0	2.58

(To simplify this example, we don't do idf weighting.)

28

Cosine: Example

log frequency weighting

term	SaS	PaP	WH
AFFECTION	3.06	2.76	2.30
JEALOUS	2.0	1.85	2.04
GOSSIP	1.30	0	1.78
WUTHERING	0	0	2.58

log frequency weighting & cosine normalization

term	SaS	PaP	WH
AFFECTION	0.789	0.832	0.524
JEALOUS	0.515	0.555	0.465
GOSSIP	0.335	0.0	0.405
WUTHERING	0.0	0.0	0.588

- cos(SaS,PaP) ≈
  $0.789 * 0.832 + 0.515 * 0.555 + 0.335 * 0.0 + 0.0 * 0.0 \approx 0.94.$
- cos(SaS,WH) ≈ 0.79
- cos(PaP,WH) ≈ 0.69
- Why do we have cos(SaS,PaP) > cos(SAS,WH)?

29

Computing the cosine score

```
CosineScore(q)
 1  float Scores[N] = 0
 2  float Length[N]
 3  for each query term t
 4  do calculate w_{t,q} and fetch postings list for t
 5      for each pair(d, tf_{t,d}) in postings list
 6      do Scores[d] += w_{t,d} × w_{t,q}
 7  Read the array Length
 8  for each d
 9  do Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```

Components of tf-idf weighting

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

31

tf-idf example

- We often use different weightings for queries and documents.
- Notation: ddd.qqq
- Example: lnc.ltn
- document: logarithmic tf, no df weighting, cosine normalization
- query: logarithmic tf, idf, no normalization
- Isn't it bad to not idf-weight the document?
- Example query: "best car insurance"
- Example document: "car insurance auto insurance"

tf-idf example: lnc.ltn

Query: “best car insurance”. Document: “car insurance auto insurance”.

word	query					document				product
	tf-raw	tf-wght	df	idf	weight	tf-raw	tf-wght	weight	n'lized	
auto	0	0	5000	2.3	0	1	1	1	0.52	0
best	1	1	50000	1.3	1.3	0	0	0	0	0
car	1	1	10000	2.0	2.0	1	1	1	0.52	1.04
insurance	1	1	1000	3.0	3.0	2	1.3	1.3	0.68	2.04

Key to columns: tf-raw: raw (unweighted) term frequency, tf-wght: logarithmically weighted term frequency, df: document frequency, idf: inverse document frequency, weight: the final weight of the term in the query or document, n’lized: document weights after cosine normalization, product: the product of final query weight and final document weight

$\sqrt{1^2 + 0^2 + 1^2 + 1.3^2} \approx 1.92$

1/1.92 ≈ 0.52

1.3/1.92 ≈ 0.68 Final similarity score between query and document: $\sum_i w_{qi} \cdot w_{di} = 0 + 0 + 1.04 + 2.04 = 3.08$ Questions?

33

Summary: Ranked retrieval in the vector space model

- Represent the query as a weighted tf-idf vector
- Represent each document as a weighted tf-idf vector
- Compute the cosine similarity between the query vector and each document vector
- Rank documents with respect to the query
- Return the top K (e.g., $K = 10$) to the user

34