Information Retrieval 1

Term-based Retrieval

Ilya Markov
i.markov@uva.nl

University of Amsterdam

Document representation and matching

Evaluation	**Document representation & matching**	Conversational search
Learning to rank	IR—user interaction	Recommender systems

Outline

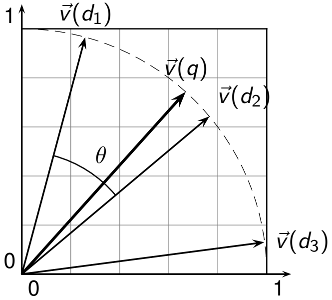
Outline

Documents as vectors

	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	
...							

Manning et al., "Introduction to Information Retrieval"

Match using cosine similarity



$$sim(d, q) = \cos(\vec{v}(d), \vec{v}(q)) = \frac{\vec{v}(d) \cdot \vec{v}(q)}{\|\vec{v}(d)\| \cdot \|\vec{v}(q)\|}$$

$$= \frac{\sum_{i=1}^{|V|} d_i \cdot q_i}{\sqrt{\sum_{i=1}^{|V|} d_i^2} \cdot \sqrt{\sum_{i=1}^{|V|} q_i^2}}$$

Manning et al., "Introduction to Information Retrieval"

Term frequency

	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	
...							

Manning et al., "Introduction to Information Retrieval"

Term frequency

Raw term frequency $tf(t, d)$

Log term frequency $\begin{cases} 1 + \log tf(t, d) & \text{if } tf(t, d) > 0 \\ 0 & \text{otherwise} \end{cases}$

Inverse document frequency

$$idf(t) = \log \frac{N}{df(t)}$$

- $df(t)$ – document frequency of term t
- N – total number of documents in a collection

Inverse document frequency

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

for $N = 1,000,000$ and $\log_{10}$

Manning et al., "Introduction to Information Retrieval"

TF-IDF

$$\text{TF-IDF}(t, d) = tf(t, d) \cdot idf(t)$$

- Term frequency
  - $tf(t, d)$
  - $\begin{cases} 1 + \log tf(t, d) & \text{if } tf(t, d) > 0 \\ 0 & \text{otherwise} \end{cases}$
- Inverse document frequency
  - $\log \frac{N}{df(t)}$
  - $\max\{0, \log \frac{N - df(t)}{df(t)}\}$

Vector space model summary

- Documents and queries as vectors
- Match using cosine similarity
- Weights can be
    1. binary
    2. term frequency
    3. TF-IDF

Outline

1. Vector space model
2. Language modeling in IR
    - Method
    - Smoothing
3. BM25

Outline

2. Language modeling in IR
   - Method
   - Smoothing

Language model

A statistical language model is a probability distribution over sequences of words.

- Given a sequence of length m
- A language model assigns probability $P(w_1, \ldots, w_m)$ to this sequence
- Unigram language model

$$P(w_1, \ldots, w_m) = P(w_1) \ldots P(w_m)$$

- Bi-gram language model

$$P(w_1, \ldots, w_m) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2) \ldots P(w_m \mid w_{m-1})$$

https://en.wikipedia.org/wiki/Language_model

Unigram language model example

Model M_1		Model M_2	
the	0.2	the	0.15
a	0.1	a	0.12
frog	0.01	frog	0.0002
toad	0.01	toad	0.0001
said	0.03	said	0.03
likes	0.02	likes	0.04
that	0.04	that	0.04
dog	0.005	dog	0.01
cat	0.003	cat	0.015
monkey	0.001	monkey	0.002
...	...	...	...

Manning et al., "Introduction to Information Retrieval"

Documents as distributions

- Unigram language model

$$P(t \mid M_d) = \frac{tf(t,d)}{dl(d)}$$

- A document is a multinomial distribution over words
- If some vocabulary terms do not appear in document d, then $P(t \mid M_d) = 0$
- This is addressed by smoothing

Match using query likelihood model (QLM)

- Likelihood of a document given a query

$$P(d \mid q) = \frac{P(q \mid d)P(d)}{P(q)}$$

- The prior distribution over queries $P(q)$ does not affect matching for a particular query

$$P(d \mid q) \stackrel{rank}{=} P(q \mid d)P(d)$$

- Usually, the prior distribution over documents $P(d)$ is assumed to be uniform

$$P(d \mid q) \stackrel{rank}{=} P(q \mid d) = P(q \mid M_d)$$

- "Bag of words" assumption: terms are independent

$$P(q \mid M_d) = \prod_{t \in q} P(t \mid M_d) = \prod_{t \in q} \frac{tf(t,d)}{dl(d)}$$

Match using KL-divergence

$$KL(M_d \| M_q) = \sum_{t \in V} P(t \mid M_q) \log \frac{P(t \mid M_q)}{P(t \mid M_d)}$$

Outline

2. Language modeling in IR
   - Method
   - Smoothing

Jelinek-Mercer smoothing

$$
\begin{aligned}
P_s(t \mid M_d) &= \lambda P(t \mid M_d) + (1 - \lambda) P(t \mid M_c) \\
&= \lambda \frac{tf(t,d)}{dl(d)} + (1 - \lambda) \frac{cf(t)}{cl}
\end{aligned}
$$

- $cf(t)$ – collection frequency of term t
- cl – collection length

Dirichlet smoothing

- A unigram language model can be seen as a multinomial distribution over words $\mathcal{L}_d(n_1, \ldots, n_k \mid p_1, \ldots, p_k)$
  - $n_i = tf(t_i, d)$
  - $p_i = P(t_i \mid M_d)$
- The conjugate prior for multinomial is the Dirichlet distribution $P_{prior}(p_1, \ldots, p_k; \alpha_1^{pr}, \ldots, \alpha_k^{pr})$
  - $\alpha_i^{pr} = \mu P(t_i \mid M_c)$
  - μ is a smoothing parameter ($\lambda = \frac{dl}{dl+\mu}$)
- The posterior is the Dirichlet distribution with parameters $\alpha_i^{po} = n_i + \alpha_i^{pr} = tf(t_i, d) + \mu P(t_i \mid M_c)$
- Dirichlet smoothing

$$P_s(t \mid M_d) = \frac{tf(t_i, d) + \mu P(t_i \mid M_c)}{dl(d) + \mu}$$

Language modeling for IR summary

- Documents and queries as distributions
- Match using QLM or KL-divergence
- Smoothing
  - Jelinek-Mercer smoothing
  - Dirichlet smoothing

Outline

BM25

$$BM25 = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t, d)}{k_1 \cdot \left[(1 - b) + b \cdot \frac{dl(d)}{dl_{avg}} \right] + tf(t, d)}$$

- $k_1, \ b$ – parameters
- $dl(d)$ – length of document d
- dl_{avg} – average document length

BM25

$$BM25 = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t, d)}{k_1 \cdot \left[(1 - b) + b \cdot \frac{dl(d)}{dl_{avg}} \right] + tf(t, d)}$$

- What if $k_1 \in \{0, \infty\}$?
- What of $b \in \{0, 1\}$?
- What if $tf(t, d)$ is small/large? $k_1 \in [1.2, 2], b = 0.75$

BM25 for long queries

$$BM25 = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t,d)}{k_1 \cdot \left[(1-b) + b \cdot \frac{dl(d)}{dl_{ave}} \right] + tf(t,d)} \cdot \frac{(k_3 + 1) tf(t,q)}{k_3 + tf(t,q)}$$

Experimental comparison

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
Trec8 T	Okapi BM25	Okapi	0.2292	0.2820	0.4380
	JM	$\lambda = 0.7$	0.2310 (p=0.8181)	0.2889 (p=0.3495)	0.4220 (p=0.3824)
	Dir	$\mu = 2,000$	**0.2470** (p=0.0757)	0.2911 (p=0.3739)	**0.4560** (p=0.3710)
	Dis	$\delta = 0.7$	0.2384 (p=0.0686)	0.2935 (p=0.0776)	0.4440 (p=0.6727)
	Two-Stage	auto	0.2406 (p=0.0650)	**0.2953** (p=0.0369)	0.4260 (p=0.4282)

Figure: TREC-8 Newswire, ad-hoc track, queries 401–450, title-only

G. Bennett, "A Comparative Study of Probabilistic and Language Models for Information Retrieval"

Experimental comparison

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
TREC-2001 T	Okapi BM25	Okapi	0.1522	0.2056	0.2918
	JM	$\lambda = 0.7$	0.1113 (p=0.0003)	0.1505 (p=0.0037)	0.2122 (p=0.0003)
	Dir	$\mu = 2,000$	**0.1774** (p=0.0307)	**0.2238** (p=0.3236)	**0.3184** (p=0.3165)
	Dis	$\delta = 0.7$	0.1370 (p=0.0511)	0.1906 (p=0.053)	0.2653 (p=0.1348)
	Two-Stage	auto	0.1441 (p=0.2963)	0.1934 (p=0.3992)	0.2898 (p=0.8962)

Figure: TREC-2001 Web data, ad-hoc track, queries 501–550, title-only

G. Bennett, "A Comparative Study of Probabilistic and Language Models for Information Retrieval"

Content-based retrieval summary

- Vector space model
  - Documents and queries as vectors
  - Match using cosine similarity
- Language modeling in IR
  - Documents and queries as discrtibutions
  - Match using QLM or KL-divergence
- BM25

Materials

- Manning et al., Chapters 6, 9, 11, 12
- Croft et al., Chapter 7