

Ranking-II

Temporal Representation and Retrieval Models

Ranking in Information Retrieval

- Ranking documents important for information overload, quickly finding documents which are “relevant” for the query
- Notion of relevance : In IR documents are ranked according to how relevant they are to the issued query
 - Higher the rank, more the relevance of the document
 - Given a query, find a ordered sequence or ranking of relevant documents based on the notion of relevance
- Interpretations and Modelling of relevance
 - Geometric Interpretation — Vector Space Similarity
 - Probabilistic interpretation — Probabilistic IR, Statistical LM

q

1) d1
2) d5
3) d3
4) d2

.

.

Vector Space Similarity

- Given a document collection $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots\}$, and a query \mathbf{q}
- Represent a document \mathbf{d} as a vector where the dimensions are the vocabulary of the collection \mathbf{V} (all distinct word in the collection)
 - Weights of the dimensions are boolean (term presence), integral (tf), scalar (tf-idf or BM-25)
- Represent the query also as a vector on the same “vocabulary” space
- Relevance of the document \mathbf{d} given \mathbf{q} is given by the cosine similarity (dot product) between both these vectors

Vector Space Similarity

q = “apple”

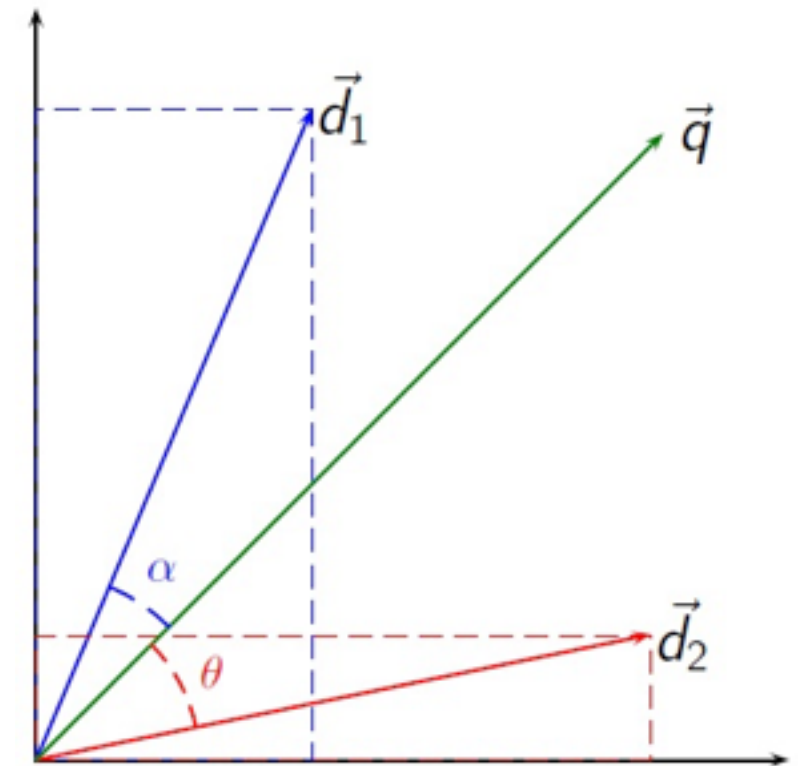
“aadvark”	“apple”	“zzz”
0	1		0

d1

“aadvark”	“apple”	“zzz”
2	0		0

d2

“aadvark”	“apple”	“zzz”
0	4		1



$$sim(d, q) = \frac{\sum_{i \in V} w_d(i) \cdot w_q(i)}{\sqrt{\sum_{i \in V} w_d(i)^2} \sqrt{\sum_{i \in V} w_q(i)^2}}$$


i.e. tf in doc.
 tf in query

Term Weighting

- Boolean retrieval — 0,1 encoding presence or absence of term
- Weighted retrieval — tf-idf score (idf for discriminative nature of term)
- Okapi-BM25 : probabilistic notion, with length normalisation

$$\sum_{t \in q} \log \left[\frac{N}{\text{df}_t} \right] \cdot \frac{(k_1 + 1) \text{tf}_{td}}{k_1((1 - b) + b \times (L_d / L_{\text{ave}})) + \text{tf}_{td}}$$

idf *tf*
length normalization

$$\text{sim}(d, q) = \frac{\sum_{i \in V} w_d(i) \cdot w_q(i)}{\sqrt{\sum_{i \in V} w_d(q_i)^2} \sqrt{\sum_{i \in V} w_q(i)^2}}$$


Statistical Language models for Ranking

- Notion of relevance: How likely is a document **D** is generated from **Q**

$$P(D|Q) \propto P(Q|D).P(D)$$

$$P(Q|D) = \prod_{w_i \in Q} \lambda.P(w_i|D) + (1 - \lambda).P(w_i|C)$$

doc. contrib.

laplace smoothing

$$P(Q|D) = \prod_{w_i \in Q} \frac{tf(w_i; D) + \mu.P(w_i|C)}{|D| + \mu}$$

doc. contrib. corpus contrib.

dirichlet smoothing

How do we improve retrieval models by adding temporal information ?

Temporal Representation

- Valid time vs Transaction time
 - Valid time: related to the period of time during which the events occur in real time
 - Transaction time refers to the time when fact was stored (say in DB or corpus)
- Focus time : time mentioned or implicitly referred to in the content
 - multiple focus times per document possible
- Reading time: In web search, reading time is assumed to be the same as the time a search query was issued

Temporal Expressions

- Temporal expressions are Natural language text which have a temporal meaning

“..in the 1980's...”
- Explicit temporal expressions : denote a precise moment in time and can be anchored on timelines without further knowledge
 - “...december 2014...”
- Implicit Temporal Expressions : associated with events carrying an implicit temporal nature
 - “...christmas day...”
 - Need to look at text proximity or explicit features for disambiguation
- Relative temporal expressions : Implicit mentions dependent on a time-point (publication date) or previous explicit mention
 - “..yesterday...”, “..next monday...”, “..in 2 hours...”

Temporal Expressions

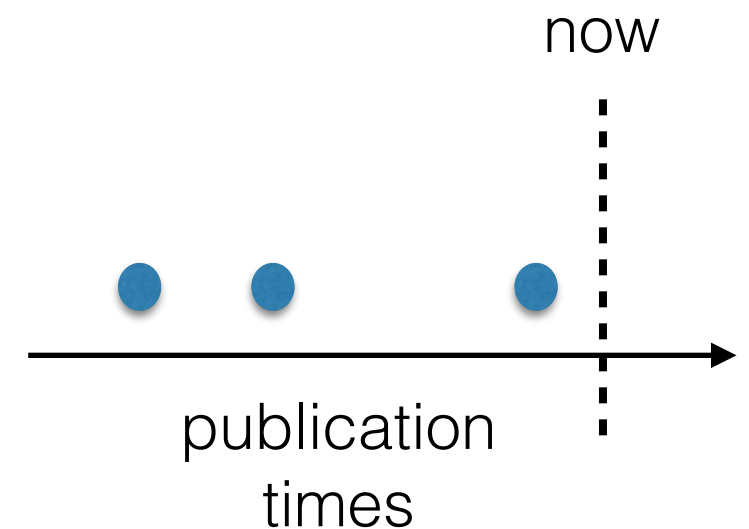
- Temporal expressions are Natural language text which have a temporal meaning

“..in the 1980's...”
- Explicit temporal expressions : denote a precise moment in time and can be anchored on timelines without further knowledge
 - “...december 2014...”
- Implicit Temporal Expressions : associated with events carrying an implicit temporal nature
 - “...christmas day...”
 - Need to look at text proximity or explicit features for disambiguation
- Relative temporal expressions : Implicit mentions dependent on a time-point (publication date) or previous explicit mention
 - “..yesterday...”, “..next monday...”, “..in 2 hours...”

Recency Aware Ranking

- Freshness of documents could be utilised for improving ranking
- How do we include freshness ?

$$P(D|Q) \propto P(Q|D).P(D)$$

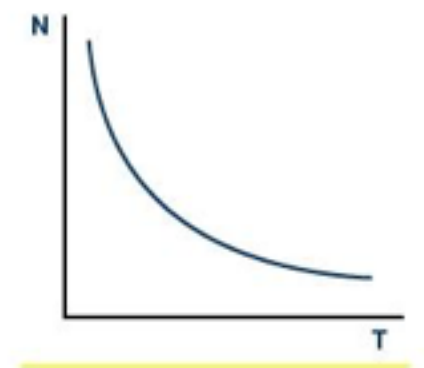


- Querying time or hitting time : Time when the query was issues
- Recently published documents are more relevant that older documents
- Exponential decay to model old documents

$$P(D) = \lambda e^{-\lambda \cdot (t_{now} - t_D)}$$

freshness param.

pub. time



Recency Aware Ranking - Timeliness

- Equal decay to all queries

$$P(D|Q) \propto P(Q|D).P(D) \longrightarrow P(D) = \lambda e^{-\lambda.(t_{now}-t_D)}$$

- Some queries are more “timely” than others
 - “hairstyle fashion trends”, “olympics”, “house of cards”
- Estimate different parameters for different queries
- Parameter typically estimated from user assessments (next lecture)

$$P(D) = \lambda_Q e^{-\lambda_Q.(t_{now}-t_D)}$$

refer to “Estimation Methods for Ranking Recent Information”, Miles Efron and Gene Golovchinsky, SIGIR ‘11

Exploiting Temporal References for Ranking

- Temporal information from text good indicators for focus time of the doc.

FIFA World Cup tournaments of the 1990's

Movies that won an Academy Award in 2007

Crusades of the 12th century

London Summer Olympics 2012

- Four-tuple representation: $T = (tb_l, tb_u, te_l, te_u)$
 - Upper and Lower bounds for begin and end times
 - in 1998, e.g., is represented as (1998/01/01, 1998/12/31, 1998/01/01, 1998/12/31)

Exploiting Temporal References for Ranking

- Distinguish between temporal and textual part of the document

$$P(Q|D) = P(Q_{text}|D_{text}).P(Q_{time}|D_{time})$$

- Independent generation of temporal query expressions

$$P(Q_{time}|D_{time}) = \prod_{q \in Q_{time}} P(q|D_{time})$$

- 2 Step Generation

- Draw a temporal expression at random
- Generate q from T

$$P(q|D_{time}) = \frac{1}{|D_{time}|} \prod_{T \in D_{time}} P(q|T)$$



Matching Temporal Expressions

- How do you match two temporal expressions
 - Exact Match: Intervals should match exactly (after normalization)
 - query: football in the 80's
 - temporal expression: ...within 1980 to 1990....
 - $p(q|T) = 1$ or 0
 - Might miss almost similar expressions

- Partial Match:

- query: football in the 80's

- temporal expression: ...in 1982 to 1989....

$$P(q|D_{time}) = \frac{1}{|D_{time}|} \prod_{T \in D_{time}} P(q|T)$$

$$P(q|T) = \frac{|T \cap q|}{|T| \cdot |q|}$$

Summary

$$P(D|Q) \propto P(Q|D) \cdot P(D)$$

$$P(Q|D) = P(Q_{text}|D_{text}) \cdot P(Q_{time}|D_{time})$$

$$P(Q_{time}|D_{time}) = \prod_{q \in Q_{time}} P(q|D_{time})$$

$$P(q|D_{time}) = \frac{1}{|D_{time}|} \prod_{T \in D_{time}} P(q|T)$$

exploiting temporal
expressions

$$P(D) = \lambda_Q e^{-\lambda_Q \cdot (t_{now} - t_D)}$$

exploiting freshness
from publication
times

- Rank documents using $P(D|Q)$ as scores
- Index scores in a temporal index for faster retrieval

References and Further Readings

- Information retrieval: (<http://www.ir.uwaterloo.ca/book/>)
 - Stefan Büttcher, Google Inc. , Charles L. A. Clarke, Univ. of Waterloo, Gordon V. Cormack, Univ. of Waterloo
- Foundations of Information retrieval: Manning, Schutze, Raghavan
- “Estimation Methods for Ranking Recent Information”, Miles Efron and Gene Golovchinsky, SIGIR '11
- Temporal Search in Web Archives, Klaus Berberich, 2010.