NWI-I00041 Information Retrieval
Retrieval Models

Dr. ir. Faegheh Hasibi
f.hasibi@cs.ru.nl

Nijmegen, September 7th, 2020

> Intellectually it is possible for a human to establish the relevance of a document to a query. For a computer to do this we need to construct **a model within which relevance decisions can be quantified**. It is interesting to note that most research in information retrieval can be shown to have been concerned with different aspects of such a model.

Van Rijsbergen, 1976

Retrieval Model

All models are wrong

but some are useful

George E.P. Box

Retrieval Model Features

- A retrieval model should provide is a clear statement about the assumptions upon which it is based

- Ideally, show that given the assumptions, show that the retrieval model will acheive better effectiveness than any other approach

- Such proofs are very hard to come by in information retrieval, Why?

- We are trying to formalize a complex human actvitiy

Retrieval Models

- Vector Space Model (VSM)

- Probabilistic Models
  - BM25
  - Language modeling - query likelihood

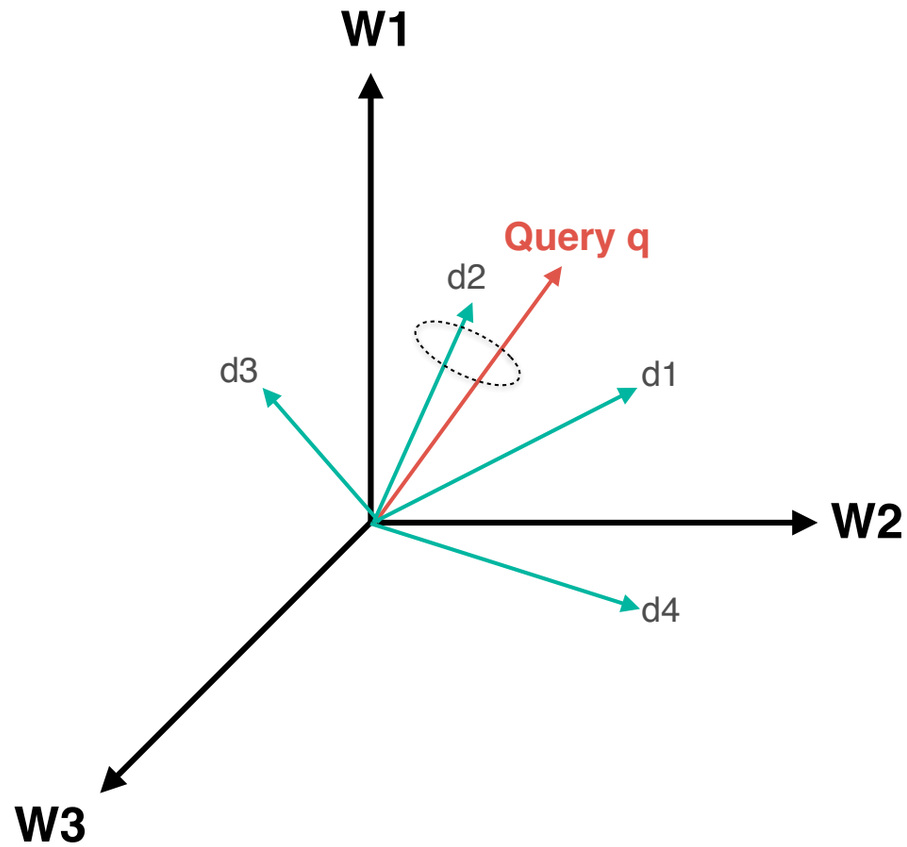
Vector Space Model

- Relevance assumption:
  - Similarity **similarity** between query and document

- If a document is more similar to a query than another document, then the first document would be assumed to be more relevant than the second one.

Vector Space Model

- Each dimension corresponds to a term
- Represent documents as vectors of term magnitudes



Vector Space Model

- Define what represents a concept in geometric space
  - terms, phrases, n-grams?

- Define the vector representation of documents

- Define the similarity function between a query and a document vector

Bag-of-Words Representation

- No word order!
- No markup!
- No decompounding!

  E.g. “voetbalstadium”

  voet / bal / stadium

In practice:

- Often stop words are removed; e.g., *of, with, and*
- Sometimes stemming is performed; e.g., *reading -> read*

A Simple Instantiation of VSM

- Bag-of-Words representation
  - Every word represents a dimension

- Bit-vector representation
  - **1** if word is present
  - **0** if word is absent

$$d = (d_1, d_2, ..., d_n) \quad d_i \in \{0, 1\}$$

- Dot product as similarity function

$$sim(q, d) = \vec{q} \cdot \vec{d} = q_1 d_1 + q_2 d_2 + .. + q_n d_n$$

$$= \sum q_i d_i$$

A Simple Instantiation of VSM

q: “Apple Phone”

d1: Apple Samsung
d2: Apple Phone Apple
d3: Phone Samsung Phone Phone Apple Samsung

V = {Apple, Samsung, Phone, …}

q = (1, 0, 1)
d1 = (1, 1, 0)
d2 = (1, 0, 1)
d3 = (1, 1, 1)
sim(q, d1) = 1*1 + 0*1 + 1*0 = 1
sim(q, d2) = 1*1 + 0*0 + 1*1 = 2
sim(q, d3) = 1*1 + 0*1 + 1*1 = 2

Improved Vector Representation - TF

- Term Frequency (TF):
  - $TF(w, d) = \text{count}(w, d)$
  - Count of a term in a document

VSM with TF Vector representation

q: “Apple Phone”

d1: Apple Samsung
d2: Apple Phone Apple
d3: Phone Samsung Phone Phone Samsung Apple Samsung

V = {Apple, Samsung, Phone, …}

q = (1, 0, 1)
d1 = (1, 1, 0)
d2 = (2, 0, 1)
d3 = (1, 3, 3)
sim(q, d1) = (1 + 0 + 0) = 1
sim(q, d2) = (2 + 0 + 1) = 3
sim(q, d3) = (1 + 0 + 3) = 4

IDF- Inverse document Frequency

> “We should treat matches on non-frequent terms as more valuable than ones on frequent terms, without disregarding the latter altogether. The natural solution is to correlate a term’s matching value with its collection frequency.

Karen Spärck Jones, 1972

Vector Representation – TF.IDF

- Term Frequency (TF):
  - $TF(w, d) = \text{count}(w, d)$
  - Count of a term in a document

- Inverse Document Frequency (IDF):
  - $\text{IDF}(w) = \log\left(\frac{M+1}{\text{df}(w)}\right)$
  - *M:* total number of documents
  - *df(w):* total number of documents containing *w*

VSM with TF.IDF Vector representation

q: “Apple Phone”

d1: Apple Samsung
d2: Apple Phone Apple
d3: Phone Samsung Phone Phone Samsung Apple Samsung

$M = 1000$
$\mathrm{df}(Apple) = 300$
$\mathrm{df}(Samsung) = 300$
$\mathrm{df}(Phone) = 600$

$$\mathrm{IDF}(Apple) = \log\left(\frac{1001}{300}\right) = 0.5$$
$$\mathrm{IDF}(Samsung) = \log\left(\frac{1001}{300}\right) = 0.5$$
$$\mathrm{IDF}(Phone) = \log\left(\frac{1001}{600}\right) = 0.2$$

V = {Apple, Samsung, Phone, …}
q = (1, 0, 1)
d1 = (1*0.5, 1*0.5, 0*0.2)
d2 = (2*0.5, 0*0.5, 1*0.2)
d3 = (1*0.5, 3*0.5, 3*0.2)
sim(q, d1) = (0.5 + 0 + 0) = 0.5
sim(q, d2) = (1 + 0 + 0.2) = 1.2
sim(q, d3) = (0.5 + 0 + 0.6) = 1.1

Retrieval Models

- Vector Space Model (VSM)

- Probabilistic Models
  - BM25
  - Language modeling - query likelihood

Relevance Assumption of Probabilistic Models

The Probability Ranking Principle:

"If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing **probability of relevance to the user** who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."

Van Rijsbergen, 1979

BM25 Model

- Derived based on Binary Independence Model (BIM):
  - $R \in \{0, 1\}$ is a binary random variable denoting relevance
  - Estimates the probability of relevance $P(R = 1 \mid d, q)$
  - Modeled as ordering by $P(R = 1|d, q)$, rather than estimating this probability directly
- Based on probabilistic arguments and experimental validation, but it is not a formal model.

BM25 Model

Derived based on Binary Independence Model (BIM):

(check [1] if interested)

$$score(q,d) = \sum_{i=1}^{n} \frac{(k+1)c(q_i,d)}{c(q_i,d) + k(1-b+b\ \frac{|d|}{avdl})} \log\left(\frac{M+1}{df(q_i)}\right)$$

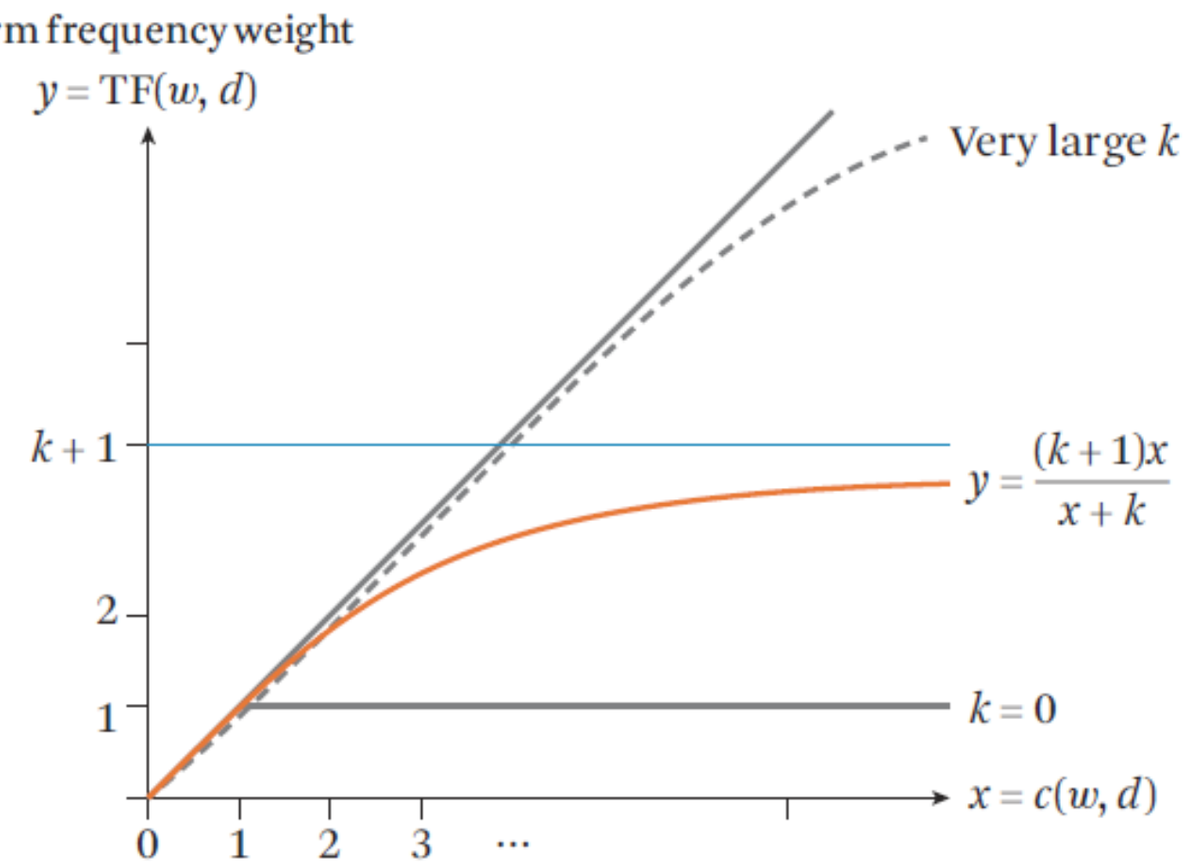
$$b \in [0,1], \quad k \in [0,+\infty)$$

|*d*|*:* document length
avdl: average document length
k, b: free parameters

[1] “Search Engines: Information Retrieval in Practice”, W. B. Croft, D. Metzler, T. Strohman

Important Elements of BM25

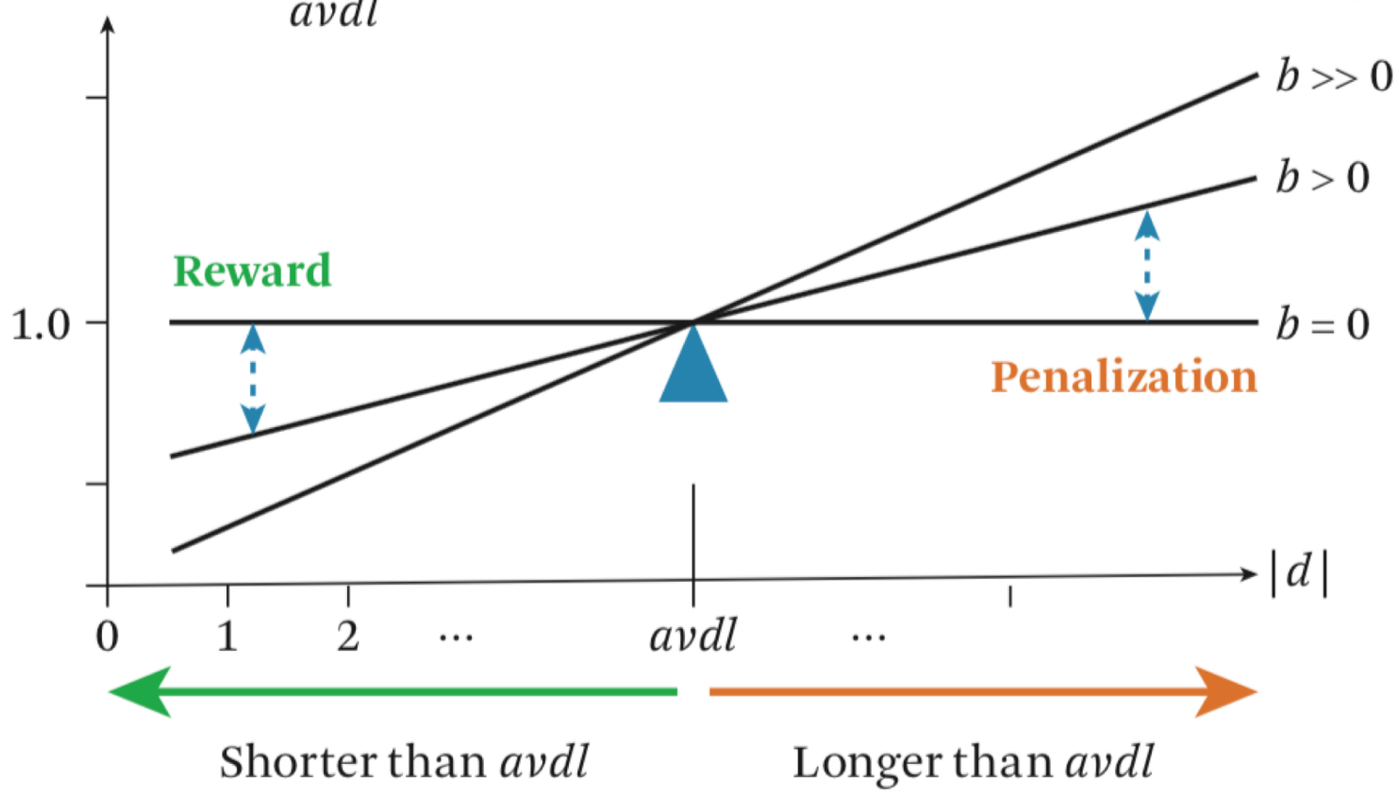
- IDF
- Transferred TF
- Document Length Normalization



Important Elements of BM25

- IDF
- Transferred TF
- Document Length Normalization

$$\text{normalizer} = 1 - b + b\frac{|d|}{avdl}$$



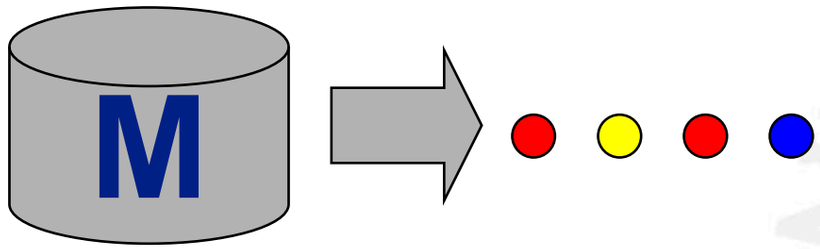
Retrieval Models

- Vector Space Model (VSM)

- Probabilistic Models
  - BM25
  - Language modeling - query likelihood

Language Modeling

- Assigns probability to a sequence of words drawn from some vocabulary



$$P(\bullet\ \bullet\ \bullet\ \bullet \mid M) = P(\bullet \mid M)\, P(\bullet \mid M, \bullet)\, P(\bullet \mid M, \bullet\ \bullet)\, P(\bullet \mid M, \bullet\ \bullet\ \bullet)$$

Unigram and Higher Order Language Models

- General From (based on chain rule)

$$P(t_1 t_2 t_3 t_4) = P(t_1)P(t_2|t_1)P(t_3|t_1 t_2)P(t_4|t_1 t_2 t_3)$$

- Unigram

$$P(t_1 t_2 t_3 t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$$

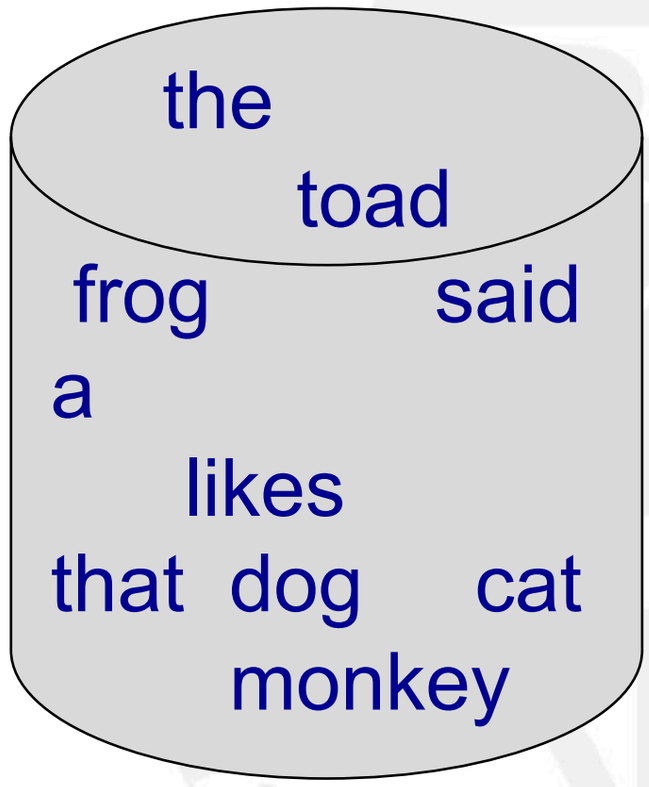
- Bigram

$$P(t_1 t_2 t_3 t_4) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)$$

- Other N-gram models, grammar-based, etc.

Unigram Language Model

Model M	
P(the)	= 0.2
P(a)	= 0.1
P(frog)	= 0.01
P(toad)	= 0.01
P(said)	= 0.03
P(likes)	= 0.02
P(that)	= 0.04
P(dog)	= 0.005
P(cat)	= 0.003
P(monkey)	= 0.001



P(frog said that toad likes frog) =
0.01 * 0.03 * 0.04 * 0.01 * 0.02 * 0.01