Information Retrieval

Dhruv Gupta

dhruv.gupta@ntnu.no

06-September-2022

NTNU | Norwegian University of Science and Technology

Announcements

- Reference Group: volunteers needed for feedback regarding course.
    - Interested? Please contact me by email!

References

- Text and diagrams of some slides are based on the material from the book: Baeza-Yates and Ribeiro-Neto, “Modern Information Retrieval", Second Edition. Pearson Education Limited, 2011.

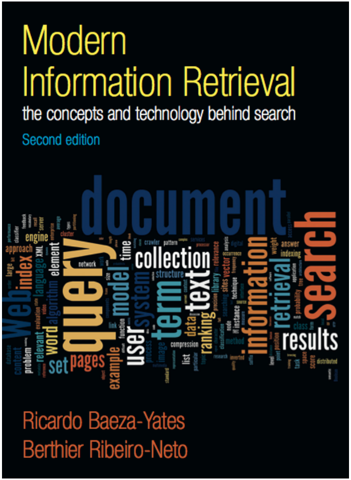
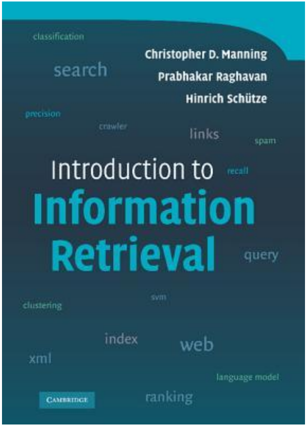


Image Credit: `http://grupoweb.upf.es/mir2ed/`

References

- Text and diagrams of some slides are based on the material from the book: Manning et al., “Introduction to Information Retrieval", First Edition. Cambridge University Press, 2008.
- Slides for statistical language models largely adapted from Hinrich Schütze’s lectures at ESSIR 2011.[1]



[1] ESSIR 2011: https://nlp.stanford.edu/IR-book/essir2011/

Image Credit: https://www.goodreads.com/book/show/3278309-introduction-to-information-retrieval

References

- Text and diagrams of some slides are based on the material from the book: Büttcher et al., "Information Retrieval," MIT Press, 2010.

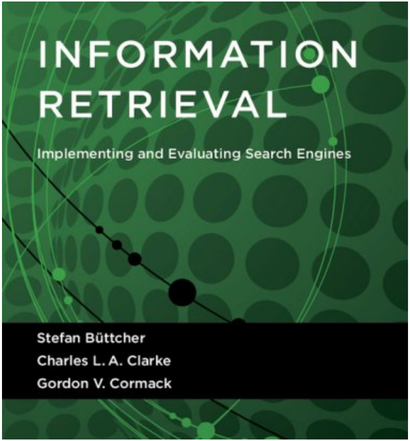


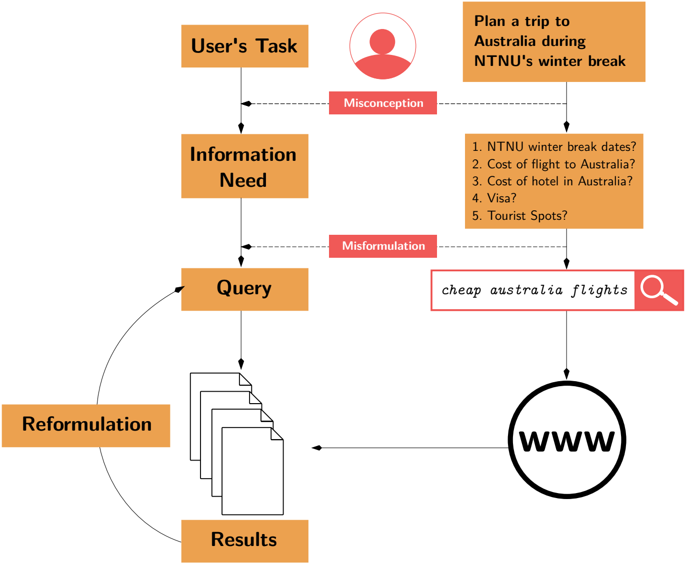
Image Credit: http://www.ir.uwaterloo.ca/book/

Recap — The IR Problem

- An IR Model can be defined by a quadruple $\langle \mathcal{D}, \mathcal{Q}, \mathcal{F}, R(q, d) \rangle$, where
    - $\mathcal{D} = \{d_1, d_2, \ldots, d_N\} \equiv$ document collection,
    - $\mathcal{Q} = \{q_1, q_2, \ldots, q_M\} \equiv$ query collection reflecting user's information needs,
    - $\mathcal{F} \equiv$ framework for modeling documents d, queries q, and their relationships.
    - $R(q, d) \equiv$ ranking function.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Recap — The IR Problem



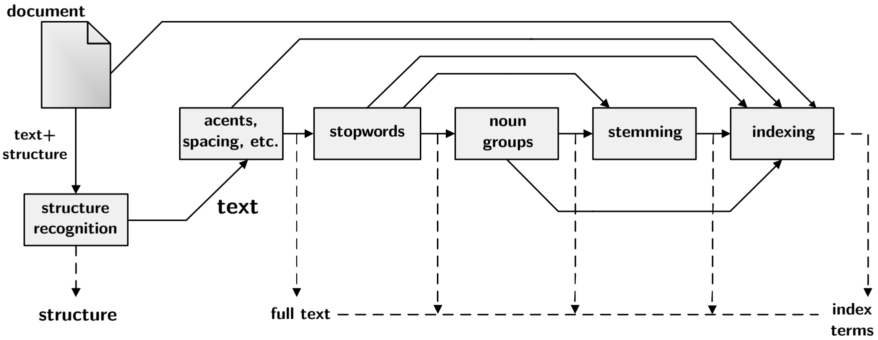
All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Recap — Modeling Documents

	w_1	w_2	w_3	w_4	w_5	$\cdots$	$w_{\|\mathcal{V}\|}$
d_1	1	0	1	1	0	$\cdots$	1
d_2	1	1	0	0	1	$\cdots$	0
d_3	0	0	0	1	0	$\cdots$	0
d_4	0	1	0	0	1	$\cdots$	1
d_5	0	0	1	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
d_N	1	1	0	1	1	$\cdots$	0

Recap — Modeling Documents

- Logical view of a document: from full text to a set of index terms



All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Recap — Boolean Retrieval

- Consider a Boolean query: $q = w_1 \wedge (w_2 \vee \neg w_3)$.
- Term vector for $w_1 = \langle 1, 1, 0, 0, 0 \rangle$.
- Term vector for $w_2 = \langle 0, 1, 0, 1, 0 \rangle$.
- Term vector for $w_3 = \langle 1, 0, 0, 0, 1 \rangle$.

	1	2	3	4	5
w_3	1	0	0	0	1
$\neg w_3$	0	1	1	1	0

	1	2	3	4	5
$\neg w_3$	0	1	1	1	0
w_2	0	1	0	1	0
OR	0	1	1	1	0

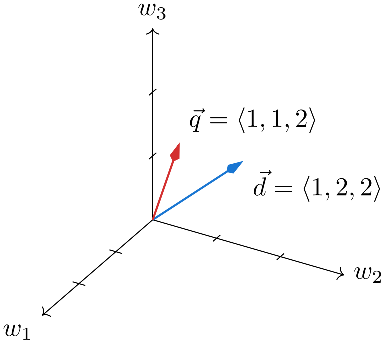
	1	2	3	4	5
$w_2 \vee \neg w_3$	0	1	1	1	0
w_1	1	1	0	0	0
AND	0	1	0	0	0

	w_1	w_2	w_3	w_4	w_5
d_1	1	0	1	1	0
d_2	1	1	0	0	1
d_3	0	0	0	1	0
d_4	0	1	0	0	1
d_5	0	0	1	0	0

Recap — The Vector Space Model

- They are represented as unit vectors of a $|\mathcal{V}|$-dimensionsal space.
- The representations of document d and query q are $|\mathcal{V}|$-dimensional vectors given by:

$$\vec{d}_i = \langle m_{i,1}, m_{i,2}, \ldots, m_{i,|\mathcal{V}|} \rangle$$
$$\vec{q} = \langle m_{q,1}, m_{q,2}, \ldots, m_{q,|\mathcal{V}|} \rangle$$



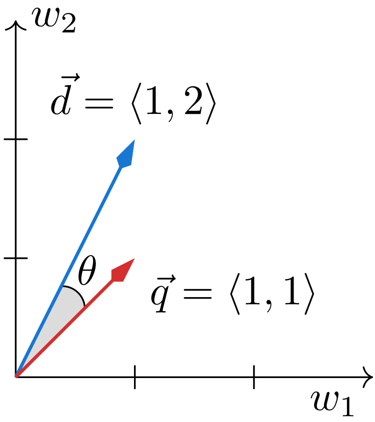
All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Recap — The Vector Space Model

- Similarity between a document and query $\text{sim}(\vec{d}, \vec{q})$ is equated to its cosine-similarity:

$$\cos(\theta) = \frac{\vec{d}_i \cdot \vec{q}}{|\vec{d}| \cdot |\vec{q}|}$$

$$\cos(\theta) = \frac{\sum_{j=1}^{|\mathcal{V}|} m_{i,j} \cdot m_{q,j}}{\sqrt{\sum_{j=1}^{|\mathcal{V}|} m_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{|\mathcal{V}|} m_{i,q}^2}}$$



All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model

- The probabilistic model captures the IR problem using a probabilistic framework.
- Given a user query, there is an ideal answer set for this query.
- Given a description of this ideal answer set, we could retrieve the relevant documents.
- Querying is seen as a specification of the properties of this ideal answer set. But, what are these properties?

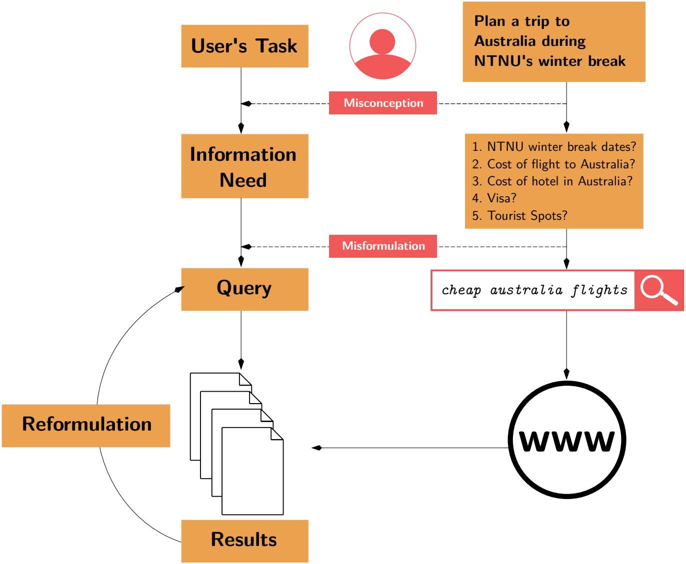
All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model — Relevance Feedback from User

- An initial set of documents is retrieved somehow.
- The user inspects these docs looking for the relevant ones (in truth, only top 10-20 need to be inspected).
- The IR system uses this information to refine the description of the ideal answer set.
- By repeating this process, it is expected that the description of the ideal answer set will improve.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model



All from Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

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.

All from: Manning et al., "Introduction to Information Retrieval", First Edition. Cambridge University Press, 2008.

The Probabilistic Model

- The Probabilistic Model:
  - Tries to estimate the probability that a document will be relevant to a user query.
  - Assumes that this probability depends on the query and document representations only.
  - The ideal answer set, referred to as R, should maximize the probability of relevance.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model — Framework

- Let, R be the set of relevant documents to query q.
- Let, $\bar{R}$ be the set of non-relevant documents to query q.
- Let, $P(R|\vec{d}_i)$ be the probability that d_i is relevant to the query q.
- Let, $P(\bar{R}|\vec{d}_i)$ be the probability that d_i is non-relevant to q.
- The similarity $\text{sim}(\vec{d}_i, q)$ can be defined as:

$$\text{sim}(d_i, q) = \frac{P(R|\vec{d}_i, q)}{P(\bar{R}|\vec{d}_i, q)}.$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model — Framework

- Using Bayes' Rule:

$$\mathrm{sim}(d_i, q) = \frac{P(R|\vec{d}_i, q)}{P(\bar{R}|\vec{d}_i, q)} = \frac{P(\vec{d}_i|R, q) \cdot P(R|q)}{P(\vec{d}_i|\bar{R}, q) \cdot P(\bar{R}|q)} \sim \frac{P(\vec{d}_i|R, q)}{P(\vec{d}_i|\bar{R}, q)}.$$

- where, $P(\vec{d}_i|R, q)$ represents the probability of randomly selecting the document d_i from the set R.
- where, $P(R|q)$ represents the probability that a document randomly selected from the entire collection is relevant to query q.
- where, $P(\vec{d}_i|\bar{R}, q)$ and $P(\bar{R}|q)$ represents same definitions above w.r.t. non-relevant document set $\bar{R}$.
- Since, $P(R|q)/P(\bar{R}|q)$ is independent of d, we can ignore this and assume the simplification as rank equivalent.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

and Büttcher et al., "Information Retrieval," MIT Press.

The Binary Independence Model

- Binary Independence Model Assumption 1: the weights $m_{i,j}$ are all binary and assuming independence among the index terms, we have

$$\mathrm{sim}(d_i, q) \sim \frac{P(\vec{d}_i|R, q)}{P(\vec{d}_i|\bar{R}, q)}$$

$$\sim \frac{(\prod_{w_j|m_{i,j}=1} P(w_j|R, q)) \cdot (\prod_{w_j|m_{i,j}=0} P(\bar{w}_j|R, q))}{(\prod_{w_j|m_{i,j}=1} P(w_j|\bar{R}, q)) \cdot (\prod_{w_j|m_{i,j}=0} P(w_j|\bar{R}, q))}$$

- where, $P(w_j|R, q)$ represents the probability that the term w_j is present in a document randomly selected from the set R.
- where, $P(\bar{w}_j|R, q)$ represents the probability that the term w_j is not present in a document randomly selected from the set R.
- where, $P(\bar{w}_j|\bar{R}, q)$ and $P(\bar{w}_j|\bar{R}, q)$ have analogous meanings to the ones described for $\bar{R}$.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- To simplify notation, take:
  - $p_{jR} = P(w_j|R, q)$.
  - $q_{jR} = P(w_j|\bar{R}, q)$.
- Also,
  - $P(w_j|R, q) + P(\bar{w}_j|R, q) = 1$.
  - $P(w_j|\bar{R}, q) + P(\bar{w}_j|\bar{R}, q) = 1$.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- We have,

$$\begin{aligned} \mathrm{sim}(d_i, q) &\sim \frac{\left(\prod_{w_j|m_{i,j}=1} P(w_j|R, q)\right) \cdot \left(\prod_{w_j|m_{i,j}=0} P(\bar{w}_j|R, q)\right)}{\left(\prod_{w_j|m_{i,j}=1} P(w_j|\bar{R}, q)\right) \cdot \left(\prod_{w_j|m_{i,j}=0} P(w_j|\bar{R}, q)\right)} \\ &\sim \frac{\left(\prod_{w_j|m_{i,j}=1} p_{jR}\right) \cdot \left(\prod_{w_j|m_{i,j}=0}(1 - p_{jR})\right)}{\left(\prod_{w_j|m_{i,j}=1} q_{jR}\right) \cdot \left(\prod_{w_j|m_{i,j}=0}(1 - q_{jR})\right)}. \end{aligned}$$

- where, $P(w_j|R, q)$ represents the probability that the term w_j is present in a document randomly selected from the set R.
- where, $P(\bar{w}_j|R, q)$ represents the probability that the term w_j is not present in a document randomly selected from the set R.
- where, $P(\bar{w}_j|\bar{R}, q)$ and $P(\bar{w}_j|\bar{R}, q)$ have analogous meanings to the ones described for $\bar{R}$.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- Taking logarithms, we write

$$
\begin{aligned}
\mathrm{sim}(d_i, q) \sim \quad & \log\left[\frac{\left(\prod_{w_j|m_{i,j}=1} P(w_j|R,q)\right)\cdot\left(\prod_{w_j|m_{i,j}=0} P(\bar{w}_j|R,q)\right)}{\left(\prod_{w_j|m_{i,j}=1} P(w_j|\bar{R},q)\right)\cdot\left(\prod_{w_j|m_{i,j}=0} P(w_j|\bar{R},q)\right)}\right] \\
\sim \quad & \log\left[\frac{\left(\prod_{w_j|m_{i,j}=1} p_{jR}\right)\cdot\left(\prod_{w_j|m_{i,j}=0}(1-p_{jR})\right)}{\left(\prod_{w_j|m_{i,j}=1} q_{jR}\right)\cdot\left(\prod_{w_j|m_{i,j}=0}(1-q_{jR})\right)}\right] \\
\sim \quad & \log\prod_{w_j|m_{i,j}=1} p_{jR} + \log\prod_{w_j|m_{i,j}=0}(1-p_{jR}) \\
& -\log\prod_{w_j|m_{i,j}=1} q_{jR} - \log\prod_{w_j|m_{i,j}=0}(1-q_{jR})
\end{aligned}
$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- Summing up terms that cancel each other,

$$\begin{aligned}
\mathrm{sim}(d_i, q) \sim \quad & \log \prod_{w_j|m_{i,j}=1} p_{jR} + \log \prod_{w_j|m_{i,j}=0} (1 - p_{jR}) \\
& - \log \prod_{w_j|m_{i,j}=1} (1 - p_{jR}) + \log \prod_{w_j|m_{i,j}=1} (1 - p_{jR}) \\
& - \log \prod_{w_j|m_{i,j}=1} q_{jR} - \log \prod_{w_j|m_{i,j}=0} (1 - q_{jR}) \\
& + \log \prod_{w_j|m_{i,j}=1} (1 - q_{jR}) - \log \prod_{w_j|m_{i,j}=1} (1 - q_{jR})
\end{aligned}$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- Further re-arranging using logarithm operations,

$$\begin{aligned} sim(d_i, q) \sim \quad & \log \prod_{w_j | m_{i,j}=1} \frac{p_{jR}}{(1 - p_{jR})} + \log \prod_{w_j} (1 - p_{jR}) \\ & + \log \prod_{w_j | m_{i,j}=1} \frac{(1 - q_{jR})}{q_{jR}} - \log \prod_{w_j} (1 - q_{jR}) \end{aligned}$$

- Notice that two of the factors in the formula above are a function of all index terms and do not depend on document d_i. They are constants for a given query and can be disregarded for the purpose of ranking.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Binary Independence Model

- Binary Independence Model Assumption 2: $\forall w_j \notin q, p_{jR} = q_{jR}$.
- Converting log products into sums of logs, we have

$$\text{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{p_{jR}}{1 - p_{jR}}\right] + \log\left[\frac{1 - q_{jR}}{q_{jR}}\right].$$

- The above formula is a key expression for ranking computation in the probabilistic model.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Ranking Formula

- Let, N be the number of documents in the collection.
- Let, n_j be the number of documents that contain term w_j.
- Let, R be the total number of relevant documents to query q.
- Let, r_j be the number of relevant documents that contain term w_j.
- Based on these variables, we can build the following contingency table.

	Relevant	**Non-Relevant**	**All Docs**
Docs that Contain w_j	r_j	$n_j - r_j$	n_j
Docs that Do Not Contain w_j	$R - r_j$	$N - n_j - (R - r_j)$	$N - n_j$
All Docs	R	$N - R$	N

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Ranking Formula

- If information on the contingency table were available for a given query, we could write

$$p_{jR} = P(w_j|R, q) = \frac{r_j}{R}$$

$$q_{jR} = P(w_j|\bar{R}, q) = \frac{n_j - r_j}{N - R}$$

- Then, the equation for ranking computation in the probabilistic model could be rewritten as

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{r_j}{R - r_j} \cdot \frac{N - n_j - R + r_j}{n_j - r_j}\right).$$

	Relevant	**Non-Relevant**	**All Docs**
Docs that Contain w_j	r_j	$n_j - r_j$	n_j
Docs that Do Not Contain w_j	$R - r_j$	$N - n_j - (R - r_j)$	$N - n_j$
All Docs	R	$N - R$	N

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Ranking Formula

- In the previous formula, we are still dependent on an estimation of the relevant docs for the query.
- For handling small values of r_j, we add 0.5 to each of the terms in the formula above, which changes $\mathrm{sim}(d_i, q)$ into

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{r_j + 0.5}{R - r_j + 0.5} \cdot \frac{N - n_j - R + r_j + 0.5}{n_j - r_j + 0.5}\right).$$

- This formula is considered as the classic ranking equation for the probabilistic model and is known as the Robertson-Sparck Jones Equation.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Ranking Formula

- The previous equation cannot be computed without estimates of r_j and R.
- One possibility is to assume $R = r_j = 0$, as a way to bootstrap the ranking equation, which leads to:

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{N - n_j + 0.5}{n_j + 0.5}\right).$$

- This equation provides an idf-like ranking computation.
- In the absence of relevance information, this is the equation for ranking in the probabilistic model.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Probabilistic Ranking Example

- Query: *to do*.

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{N - n_j + 0.5}{n_j + 0.5}\right).$$

D1: *To do is to be. To be is to do.*

D2: *To be or not to be. I am what I am.*

D3: *I think therefore I am. Do be do be do.*

D4: *Do do do, da da da. Let it be, let it be.*

doc	rank computation	rank
d_1	$\log \frac{4-2+0.5}{2+0.5} + \log \frac{4-3+0.5}{3+0.5}$	- 1.222
d_2	$\log \frac{4-2+0.5}{2+0.5}$	0
d_3	$\log \frac{4-3+0.5}{3+0.5}$	- 1.222
d_4	$\log \frac{4-3+0.5}{3+0.5}$	- 1.222

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model — Ranking Formula

- The ranking computation led to negative weights because of the term `do`.
- Actually, the probabilistic ranking equation produces negative terms whenever $n_j > N/2$.
- One possible artifact to contain the effect of negative weights is to change the previous equation to:

$$\text{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{N + 0.5}{n_j + 0.5}\right).$$

- By doing so, a term that occurs in all documents $(n_j = N)$ produces a weight equal to zero.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Probabilistic Ranking Example

- Query: *to do*.

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left(\frac{N+0.5}{n_j+0.5}\right).$$

D1

To do is to be. To be is to do.

D2

To be or not to be. I am what I am.

D3

I think therefore I am. Do be do be do.

D4

Do do do, da da da. Let it be, let it be.

doc	rank computation	rank
d_1	$\log \frac{4+0.5}{2+0.5} + \log \frac{4+0.5}{3+0.5}$	1.210
d_2	$\log \frac{4+0.5}{2+0.5}$	0.847
d_3	$\log \frac{4+0.5}{3+0.5}$	0.362
d_4	$\log \frac{4+0.5}{3+0.5}$	0.362

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Estimating r_j and R

- Our examples above considered that $r_j = R = 0$.
- An alternative is to estimate r_j and R performing an initial search.
  - Select the top 10-20 ranked documents.
  - Inspect them to gather new estimates for r_j and R.
  - Remove the 10-20 documents used from the collection.
  - Re-run the query with the estimates obtained for r_j and R.
- Unfortunately, procedures such as these require human intervention to initially select the relevant documents.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model — Estimating r_j and R

- Consider the equation,

$$\text{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{p_{jR}}{1 - p_{jR}}\right] + \log\left[\frac{1 - q_{jR}}{p_{jR}}\right].$$

- How to obtain the probabilities p_{jR} and q_{jR}?
- Estimates based on assumptions:
  - $p_{jR} = 0.5$.
  - $q_{jR} = \frac{n_j}{N}$ where, n_j is the number of docs that w_j.
  - Use this initial guess to retrieve an initial ranking.
  - Improve upon this initial ranking.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Estimating r_j and R

- Substituting p_{jR} and q_{jR} into the previous Equation, we obtain:

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{N - n_j}{n_j}\right].$$

- That is the equation used when no relevance information is provided, without the 0.5 correction factor.
- Given this initial guess, we can provide an initial probabilistic ranking.
- After that, we can attempt to improve this initial ranking as follows.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Improving Estimates for r_j and R

- We can attempt to improve this initial ranking as follows.
- Let,
  - D : set of documents initially retrieved.
  - D_j : subset of documents retrieved that contain w_j.
- Re-evaluate estimates:
  - $p_{jR} = D$
  - $q_{jR} = \frac{n_j - D_j}{N - D}$
- Relevance Feedback: this process can then be repeated recursively.
- Pseudo-Relevance Feedback: if the interaction with the user is absent, then the top$-k$ documents retrieved are chosen as relevant for ranking improvement.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

The Probabilistic Model

- The probabilistic ranking formula:

$$\mathrm{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{N - n_j}{n_j}\right].$$

- To avoid problems with $D = 1$ and $D_j = 0$:

$$p_{jR} = \frac{D_j + 0.5}{D + 1} \text{ and } q_{jR} = \frac{n_j - D_j + 0.5}{N - D + 1}.$$

- Also,

$$p_{jR} = \frac{D_j + \frac{n_j}{N}}{D + 1} \text{ and } q_{jR} = \frac{n_j - D_j + \frac{n_j}{N}}{N - D + 1}.$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Advantages and Disadvantages

- Advantage: documents are ranked in decreasing order of their probability of being relevant.
- Disadvantage:
  - Need to guess initial separation of documents in to relevant and non-relevant sets.
  - Index terms are assumed to be occurring independent to each other in the document.
  - No accounting of term frequency (all weights are binary).
  - There is a lack of document length normalization.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Comparison of the Probabilistic Model to Classic Models

- Boolean model does not provide for partial matches and is considered to be the weakest classic model.
- There is some controversy as to whether the probabilistic model outperforms the vector model.
- Croft [2] suggested that the probabilistic model provides a better retrieval performance.
- However, Salton et al [3] showed that the vector model outperforms it with general collections.
- This also seems to be the dominant thought among researchers and practitioners of IR.

[2] http://portal.acm.org/citation.cfm?id=106765.106784

[3] http://portal.acm.org/citation.cfm?id=866292

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM25 (Best Match 25)

- BM25 was created as the result of a series of experiments on variations of the probabilistic model.
- A good term weighting is based on three principles:
  - Inverse Document Frequency
  - Term Frequency
  - Document Length Normalization
- The classic probabilistic model covers only the first of these principles.
- This reasoning led to a series of experiments with the Okapi system, which led to the BM25 ranking formula.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM25 (Best Match 25)

- Unlike the probabilistic model, the BM25 formula can be computed without relevance information.
- There is consensus that BM25 outperforms the classic vector model for general collections.
- Thus, it has been used as a baseline for evaluating new ranking functions, in substitution to the classic vector model.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- At first, the Okapi system used the Equation below as ranking formula:

$$\text{sim}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log \frac{N - n_j + 0.5}{n_j + 0.5}.$$

- which is the equation used in the probabilistic model, when no relevance information is provided.
- It was referred to as the BM1 formula (Best Match 1).

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- The first idea for improving the ranking was to introduce a term-frequency factor $\mathcal{F}_{i,j}$ in the BM1 formula.
- This factor, after some changes, evolved to become

$$\mathcal{F}_{i,j} = S_1 \cdot \frac{tf_{i,j}}{K_1 + tf_{i,j}}.$$

- where,
  - $\mathrm{tf}_{i,j}$ is the frequency of term w_j within document d_i.
  - K_1 is a constant setup experimentally for each collection.
  - S_1 is a scaling constant, normally set to $S_1 = (K_1 + 1)$.
- If $K_1 = 0$, this whole factor becomes equal to 1 and bears no effect in the ranking.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- The next step was to modify the $\mathcal{F}_{i,j}$ factor by adding document length normalization to it, as follows:

$$\mathcal{F}'_{i,j} = S_1 \cdot \frac{tf_{i,j}}{\frac{K_1 \cdot \text{len}(d_i)}{\text{avg\_doclen}} + tf_{i,j}}.$$

- where,
  - $\text{len}(d_i)$ is the length of document d_i (computed, for instance, as the number of terms in the document).
  - avg_doclen is the average document length for the collection.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- Next, a correction factor $\mathcal{G}_{i,q}$ dependent on the document and query lengths was added:

$$\mathcal{G}_{i,q} = K_2 \cdot \text{len}(q) \cdot \frac{\text{avg\_doclen} - \text{len}(d_i)}{\text{avg\_doclen} + \text{len}(d_i)}.$$

- where,
  - $\text{len}(q)$ is the query length (number of terms in the query).
  - K_2 is a constant.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- A third additional factor, aimed at taking into account term frequencies within queries, was defined as:

$$\mathcal{F}_{q,j} = S_3 \cdot \frac{\mathrm{tf}_{q,j}}{K_3 + \mathrm{tf}_{q,j}}.$$

- where,
  - $\mathrm{tf}_{q,j}$ is the frequency of term w_j within query q.
  - K_3 is a constant.
  - S_3 is a scaling constant related K_3, normally set to $S_3 = (K_3 + 1)$.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- Introduction of these three factors led to various BM (Best Matching) formulas, as follows:

$$\text{sim}_{\text{BM1}}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

$$\text{sim}_{\text{BM15}}(d_i, q) \sim \mathcal{G}_{i,q} + \sum_{w_j \in q \wedge w_j \in d_i} \mathcal{F}_{i,j} \cdot \mathcal{F}_{q,j} \cdot \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

$$\text{sim}_{\text{BM11}}(d_i, q) \sim \mathcal{G}_{i,q} + \sum_{w_j \in q \wedge w_j \in d_i} \mathcal{F}'_{i,j} \cdot \mathcal{F}_{q,j} \cdot \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- Experiments using TREC data have shown that BM11 outperforms BM15.
- Further, empirical considerations can be used to simplify the previous equations, as follows:
  - Empirical evidence suggests that a best value of K_2 is 0, which eliminates the $\mathcal{G}_{i,q}$ factor from these equations.
  - Further, good estimates for the scaling constants S_1 and S_3 are $K_1 + 1$ and $K_3 + 1$, respectively.
  - Empirical evidence also suggests that making K_3 very large is better. As a result, the $\mathcal{F}_{q,j}$ factor is reduced simply to $tf_{q,j}$.
  - For short queries, we can assume that $\mathrm{tf}_{q,j}$ is 1 for all terms.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM1, BM11 and BM15 Formulas

- These considerations lead to simpler equations as follows:

$$\text{sim}_{\text{BM1}}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

$$\text{sim}_{\text{BM15}}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \frac{(K_1 + 1) \cdot \text{tf}_{i,j}}{(K_1 + 1) + \text{tf}_{i,j}} \cdot \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

$$\text{sim}_{\text{BM11}}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \frac{(K_1 + 1) \cdot \text{tf}_{i,j}}{\left(\frac{K_1 \cdot \text{len}(d_i)}{\text{avg\_doclen}} + \text{tf}_{i,j}\right) + \text{tf}_{i,j}} \cdot \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM25 Ranking Formula

- BM25: combination of the BM11 and BM15.
- The motivation was to combine the BM11 and BM15 term frequency factors as follows.

$$\mathcal{B}_{i,j} = \frac{(K_1 + 1) \cdot \mathrm{tf}_{i,j}}{K_1 \cdot \left[(1 - b) + b \cdot \frac{\mathrm{len}(d_i)}{\mathrm{avg\_doclen}}\right] + \mathrm{tf}_{i,j}}.$$

- where, b is a constant with values in the interval $[0, 1]$.
  - If $b = 0$, it reduces to the BM15 term frequency factor.
  - If $b = 1$, it reduces to the BM11 term frequency factor.
  - For values of $b \in (0, 1)$, the equation provides a combination of BM11 and BM15.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

BM25 Ranking Formula

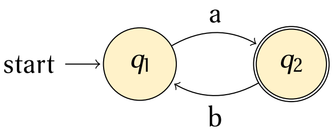
- The ranking equation for the BM25 model can then be written as:

$$\text{sim}_{\text{BM25}}(d_i, q) \sim \sum_{w_j \in q \wedge w_j \in d_i} \mathcal{B}_{i,j} \cdot \log\left[\frac{N - n_j + 0.5}{n_j + 0.5}\right]$$

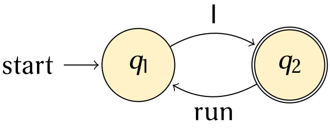
- where, K_1 and b are empirical constants.
  - $K_1 = 1$ works well with real collections.
  - b should be kept closer to 1 to emphasize the document length normalization effect present in the BM11 formula.
  - For instance, $b = 0.75$ is a reasonable assumption.
  - Constants values can be fine tuned for particular collections through proper experimentation.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

- We can view a finite state automaton as a deterministic language model.



$\boxed{a(ba)^*}$

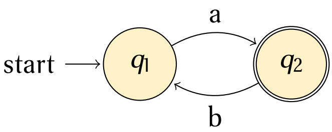


$\boxed{\text{I (run I)}^*}$

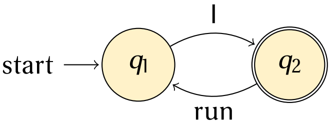
All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- We can view a finite state automaton as a deterministic language model.



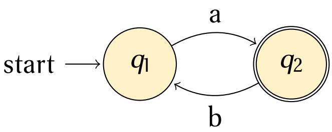
$\boxed{a(ba)^*}$



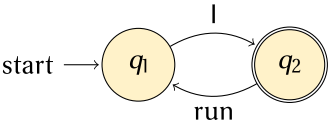
$\boxed{\text{I run I}}$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

- We can view a finite state automaton as a deterministic language model.



$\boxed{a(ba)^*}$

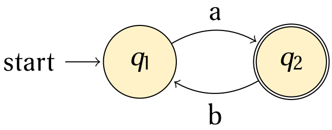


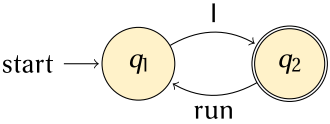
$\boxed{\text{I run I run I}}$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- We can view a finite state automaton as a deterministic language model.
- Cannot generate: “run I run”.
- Our basic model: each document was generated by a different automaton like this except that these automata are probabilistic.



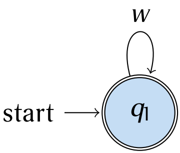


$\boxed{a(ba)^*}$

$\boxed{\text{I run I run I run I}}$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy



- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.

w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

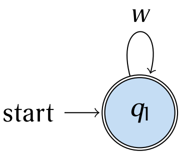
frog

$P(\text{string}) = 0.01$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w \mid q_1)$	w	$P(w \mid q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

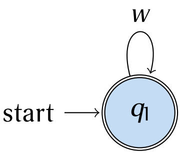
frog said

$$P(\text{string}) = 0.01 \cdot 0.03$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

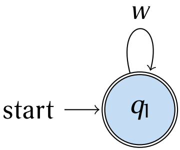
frog said that

$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

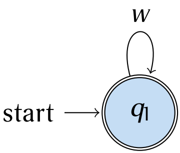
frog said that toad

$$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

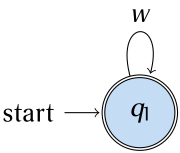
frog said that toad likes

$$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

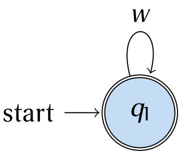
frog said that toad likes frog

$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

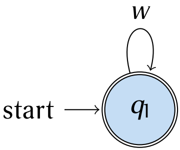
frog said that toad likes frog STOP

$$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

- This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1.
- STOP is not a word, but a special symbol indicating that the automaton stops.



w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

frog said that toad likes frog STOP

$$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2 = 0.000000000048$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

Language Model of d_1			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.01
the	0.20	said	0.03
a	0.10	likes	0.02
frog	0.01	that	0.04
		...	...

Language Model of d_2			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.02
the	0.15	said	0.03
a	0.08	likes	0.02
frog	0.01	that	0.05
		...	...

query: frog said that toad likes frog STOP

$$P(\text{query}|M_{d_1}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2.$$

$$P(\text{query}|M_{d_1}) = 0.0000000000048.$$

$$P(\text{query}|M_{d_1}) = 4.8 \cdot 10^{-12}.$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

Language Model of d_1			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.01
the	0.20	said	0.03
a	0.10	likes	0.02
frog	0.01	that	0.04
		...	...

Language Model of d_2			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.02
the	0.15	said	0.03
a	0.08	likes	0.02
frog	0.01	that	0.05
		...	...

query: frog said that toad likes frog STOP

$$P(\text{query}|M_{d_2}) = 0.01 \cdot 0.03 \cdot 0.05 \cdot 0.02 \cdot 0.02 \cdot 0.01 \cdot 0.2.$$

$$P(\text{query}|M_{d_2}) = 0.0000000000120.$$

$$P(\text{query}|M_{d_2}) = 12 \cdot 10^{-12}.$$

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

Language Model of d_1			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.01
the	0.20	said	0.03
a	0.10	likes	0.02
frog	0.01	that	0.04
		...	...

Language Model of d_2			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.02
the	0.15	said	0.03
a	0.08	likes	0.02
frog	0.01	that	0.05
		...	...

query: frog said that toad likes frog STOP

$P(\text{query}|M_{d_1}) = 4.8 \cdot 10^{-12}$.

$P(\text{query}|M_{d_2}) = 12 \cdot 10^{-12}$.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models — Analogy

Language Model of d_1			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.01
the	0.20	said	0.03
a	0.10	likes	0.02
frog	0.01	that	0.04
		...	...

Language Model of d_2			
w	$P(w\|q_1)$	w	$P(w\|q_1)$
STOP	0.20	toad	0.02
the	0.15	said	0.03
a	0.08	likes	0.02
frog	0.01	that	0.05
		...	...

- query: frog said that toad likes frog STOP.
- $\left(P(\text{query}|M_{d_1}) = 4.8 \cdot 10^{-12}\right) < \left(P(\text{query}|M_{d_1}) = 12 \cdot 10^{-12}\right)$.
- Thus, document d_2 is more relevant to the query than d_1 is.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Statistical Language Models in IR

- Let S be a sequence of r consecutive terms that occur in a document of the collection:

$$S = \langle w_1, w_2, \ldots, w_s \rangle.$$

- An n-gram language model uses a Markov process to assign a probability of occurrence to S:

$$P_n(S) = \prod_{i=1}^{r} P\big(w_i | w_{i-1}, w_{i-2}, \ldots, w_{i-(n-1)}\big).$$

- where, n is the order of the Markov process.
- The occurrence of a term depends on observing the $n-1$ terms that precede it in the text.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Statistical Language Models in IR

- Unigram language model ($n = 1$): the estimates are based on the occurrence of individual words.
- Bigram language model ($n = 2$): the estimates are based on the co-occurrence of pairs of words.
- Higher order models such as Trigram language models ($n = 3$) are usually adopted for speech recognition.
- Term independence assumption: in the case of IR, the impact of word order is less clear.
- As a result, unigram language models have been used extensively in IR.

All from: Baeza-Yates and Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley.

Statistical Language Models in IR

- Each document is treated as (the basis for) a language model.
- Given a query q.
- Rank documents based on $P(d|q)$.

$$P(d|q) = \frac{P(q|d) \cdot P(d)}{P(q)} \propto P(q|d) \cdot P(d)$$

- $P(q)$ is the same for all documents, so ignore.
- $P(d)$ is the prior – often treated as the same for all d.
  - But we can give a higher prior to "high-quality" documents, e.g., those with high PageRank.
- $P(q|d)$ is the probability of q given d.
- Under the assumptions we made, ranking documents according to $P(q|d) \cdot P(d)$ or $P(d|q)$ is considered equivalent.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Computing $P(q|d)$

- We will make the same conditional independence assumption as in BIM.

$$P(q|M_d) = P(\langle t_1, \ldots, t_{|q|} \rangle | M_d) = \prod_{1 \leq k \leq |q|} P(t_k|M_d)$$

- where,
  - $|q|$ is length of query q.
  - t_k is the token occurring at position k in q.
- This is equivalent to:

$$P(q|M_d) = \prod_{\text{distinct term } t \text{ in } q} P(t|M_d)^{\mathrm{tf}_{q,t}}.$$

- where,
  - $\mathrm{tf}_{q,t}$ term frequency of t in q.
  - also known as, Multinomial Model (omitting constant factor).

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Parameter Estimation

- Missing piece: Where do the parameters $P(t|M_d)$ come from?
- Start with maximum likelihood estimates:

$$\hat{P}(t|M_d) = \frac{\mathrm{tf}_{d,t}}{|d|}.$$

- where,
  - $|d|$ is length of document d.
  - $\mathrm{tf}_{d,t}$ term frequency of t in d.
- But, there is a problem here!

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Parameter Estimation

- Start with maximum likelihood estimates:

$$\hat{P}(t|M_d) = \frac{\mathrm{tf}_{d,t}}{|d|}.$$

- We have a problem with zeros!
- A single t in the query with $P(t|M_d) = 0$ will make $P(q|M_d) = \prod P(t|M_d)$ zero.
- We would give a single term in the query "veto power".
- For example, for query *michael jackson top hits* a document about *michael jackson top songs* (but not using the word *hits*) would have $P(q|M_d) = 0$. That's undesirable!
- We need to smooth the estimates to avoid zeros.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Smoothing

- Key intuition: A non-occurring term is possible (even though it didn't occur).
- But no more likely than would be expected by chance in the collection.
- Notation:
  - M_c: the collection model.
  - cf_t: the number of occurrences of t in the collection.
  - $T = \sum_t \mathrm{cf}_t$: the total number of tokens in the collection.

$$\hat{P}(t|M_c) = \frac{\mathrm{cf}_t}{|T|}.$$

- We will use $\hat{P}(t|M_c)$ to *smooth* $P(t|d)$ away from zero.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Jelinek-Mercer Smoothing

- Jelinek-Mercer Smoothing:

$$P(t|d) = \lambda \cdot P(t|M_d) + (1 - \lambda) \cdot P(t|M_c)$$

- Mixes the probability from the document with the general collection frequency of the word.
- High value of λ: conjunctive-like search – tends to retrieve documents containing all query words.
- Low value of λ: more disjunctive, suitable for long queries.
- Tuning λ is important for good performance.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Jelinek-Mercer Smoothing

- Jelinek-Mercer Smoothing:

$$P(q|d) \propto \prod_{1 \leq k \leq |q|} \lambda \cdot P(t_k|M_d) + (1 - \lambda) \cdot P(t_k|M_c)$$

- What we model: the user has a document in mind and generates the query from this document.
- $P(q|d)$ is the probability that the document that the user had in mind was in fact this one.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Jelinek-Mercer Smoothing: Example

- Jelinek-Mercer Smoothing:

$$P(q|d) \propto \prod_{1 \leq k \leq |q|} \lambda \cdot P(t_k|M_d) + (1 - \lambda) \cdot P(t_k|M_c)$$

- Collection: d_1 and d_2.
- d_1: *jackson was one of the most talented entertainers of all time*.
- d_2: *michael jackson anointed himself king of pop*.
- q: *michael jackson*.
- Use mixture model with $\lambda = 1/2$.
- $P(q|d_1) = [(0/11 + 1/18)/2] \cdot [(1/11 + 2/18)/2] \approx 0.003$.
- $P(q|d_2) = [(1/7 + 1/18)/2] \cdot [(1/7 + 2/18)/2] \approx 0.013$.
- Ranking: $d_2 > d_1$.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Dirichlet Smoothing

- Dirichlet Smoothing:

$$P(t|d) = \frac{\mathrm{tf}_{d,t} + \mu \cdot P(t|M_c)}{|d| + \mu}$$

- The background distribution $P(t|M_c)$ is the prior for $P(t|d)$.
- Intuition: before having seen any part of the document we start with the background distribution as our estimate.
- As we read the document and count terms we update the background distribution.
- The weighting factor μ determines how strong an effect the prior has.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Jelinek-Mercer or Dirichlet Smoothing?

- Dirichlet performs better for keyword queries, Jelinek-Mercer performs better for verbose queries.
- Both models are sensitive to the smoothing parameters — you should not use these models without parameter tuning.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Summary of Language Models

- View the document as a generative model that generates the query.
- Define the precise generative model we want to use.
- Estimate parameters (different parameters for each document's model).
- Smooth to avoid zeros.
- Apply to query and find document most likely to have generated the query.
- Present most likely document(s) to user.

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.

Discussion

- BM25/LM: based on probability theory.
- Vector space: based on similarity, a geometric/linear algebra notion.
- Term frequency is directly used in all three models.
  - LMs: raw term frequency.
  - BM25/Vector space: more complex.
- Length normalization.
  - Vector space: in-built into Cosine similarity.
  - LMs: probabilities are inherently length normalized.
  - BM25: tuning parameters for optimizing length normalization
- Inverse document frequency.
  - BM25/vector space: use it directly.
  - LMs: Mixing term and collection frequencies has an effect similar to idf.
  - Terms rare in the general collection, but common in some documents will have a greater influence on the ranking.
  - Collection frequency (LMs) vs. document frequency (BM25, vector space).

All from: Manning et al., "Introduction to Information Retrieval," Cambridge University Press.