https://vvtesh.sarahah.com/

Information Retrieval

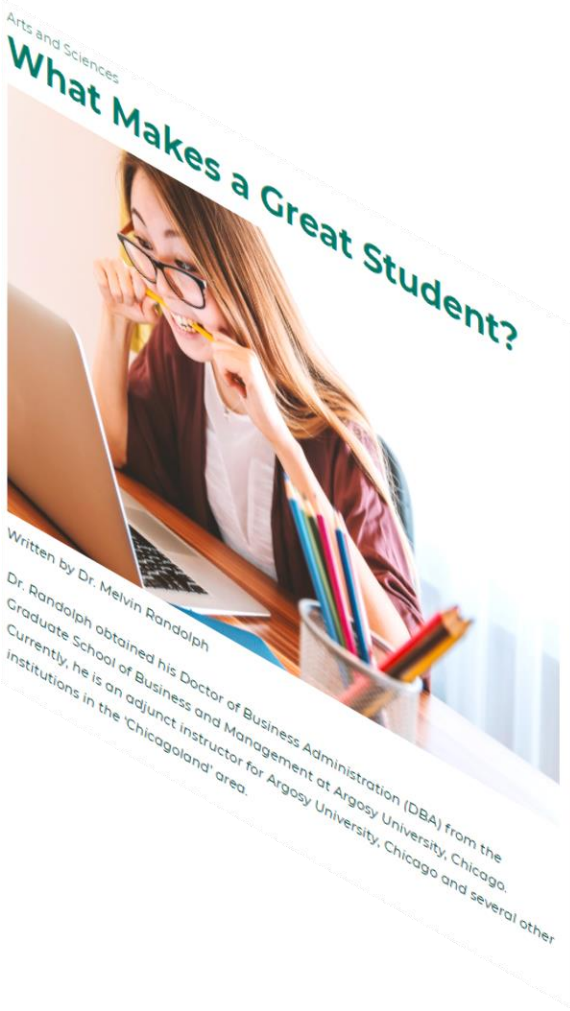
Venkatesh Vinayakarao

Term: Aug – Dec, 2018
Indian Institute of Information Technology, Sri City

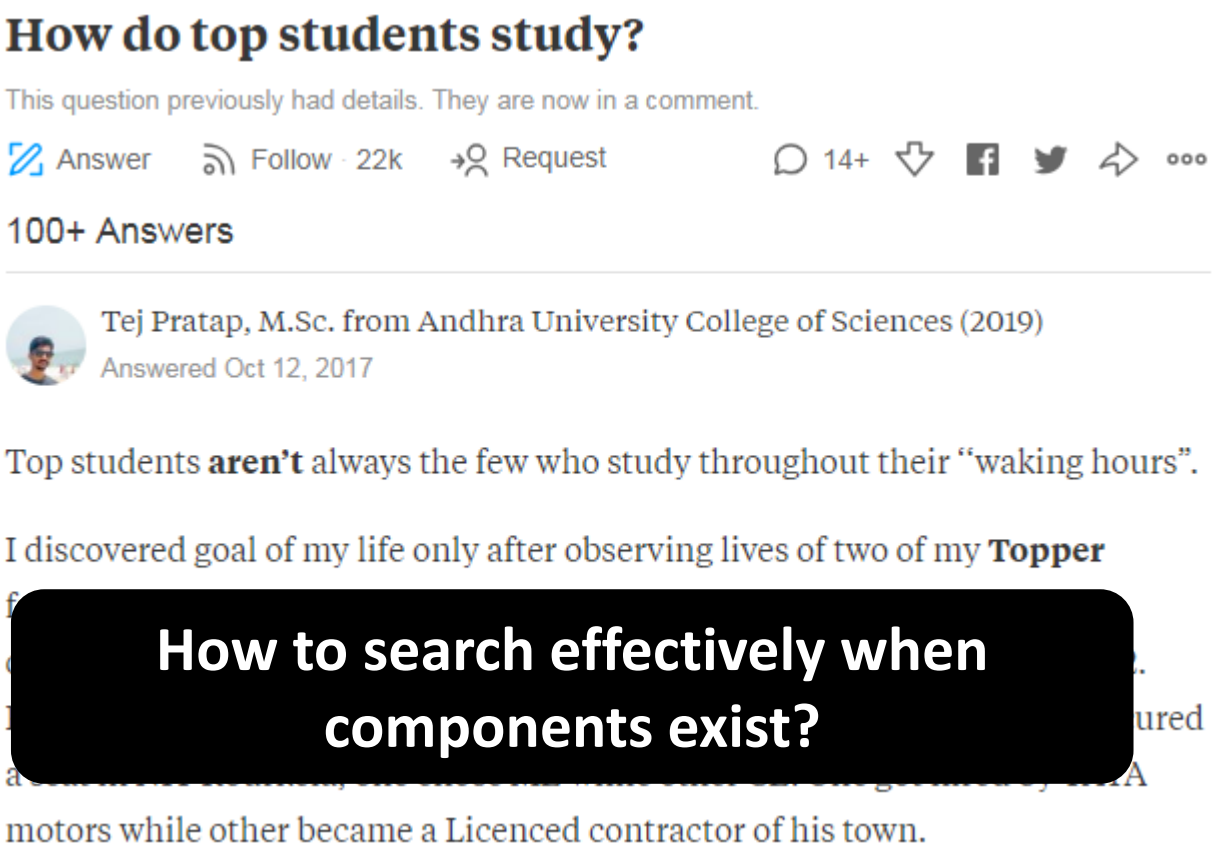
Mission defines strategy, and strategy defines structure.
– Peter Drucker.

Documents with Structure

- So far, we discussed unstructured text.
- Many documents in reality have a structure.
  - They are composed of Zones and Fields.



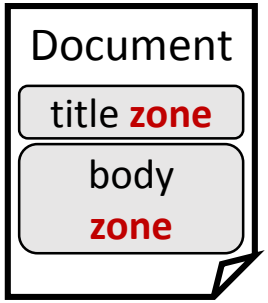
Identify Some Key Components



How to search effectively when components exist?

Zones and Fields

- A document is associated with *metadata* which are useful in search.

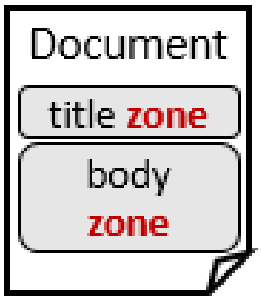


METADATA Fields
filename
author
title
date of creation

- **Sample Query**: find documents authored by William Shakesphere in 1601 containing the phrase “you brutus”
- **Sample Query**: find documents with merchant in the title and william in the author list and the phrase gentle rain in the body

Zones and Fields

- Zones are arbitrary free text.
- Fields may take relatively small set of values.
  - Fields may call for *range query* (year between 1600 and 1700) support

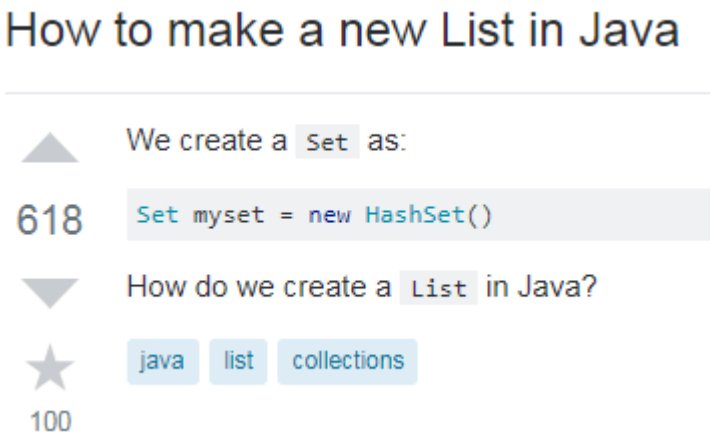


METADATA Fields
filename
author
title
date of creation

How to index zones and fields?

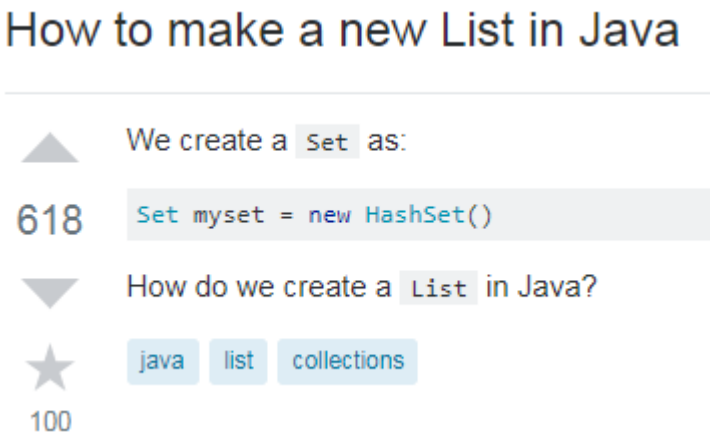
Quiz

- Assume we are indexing stackoverflow data. Which of the following are zones?
  - question
  - answer
  - number of answers
  - comments
  - number of comments
  - code blocks



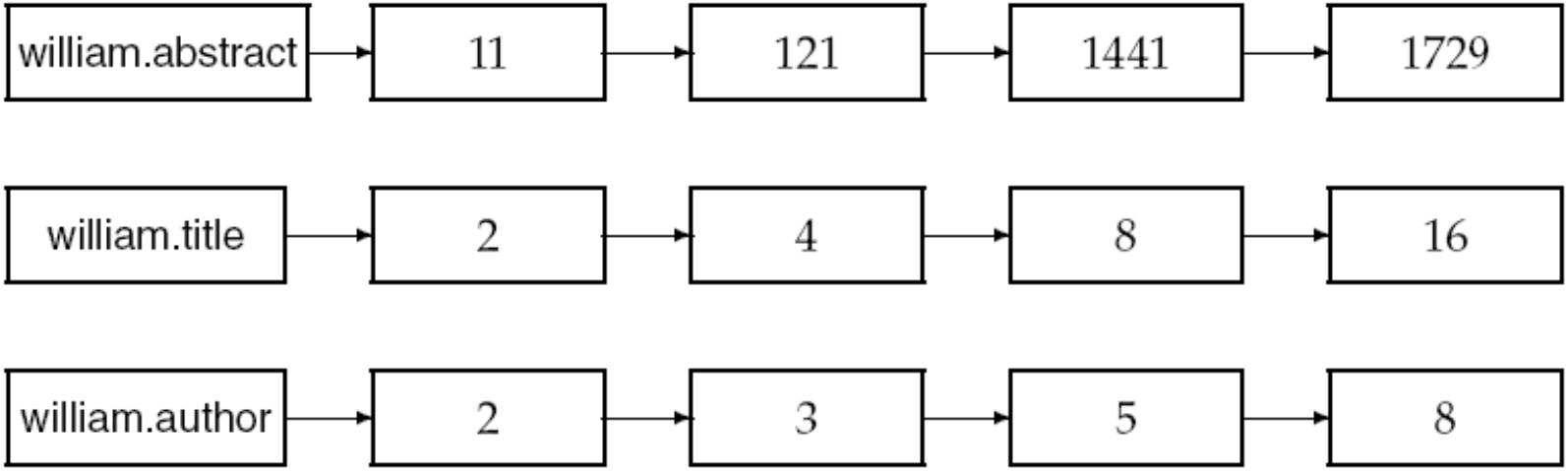
Quiz

- Assume we are indexing stackoverflow data. Which of the following are zones?
  - **question**
  - **answer**
  - number of answers
  - **comments**
  - number of comments
  - **code blocks**



Indexing Zones and Fields

- Create separate Index for each field and each zone.
  - Standard Inverted Index +
  - Parametric Indexes (one for each field) +
  - Zone Indexes (one for each zone)

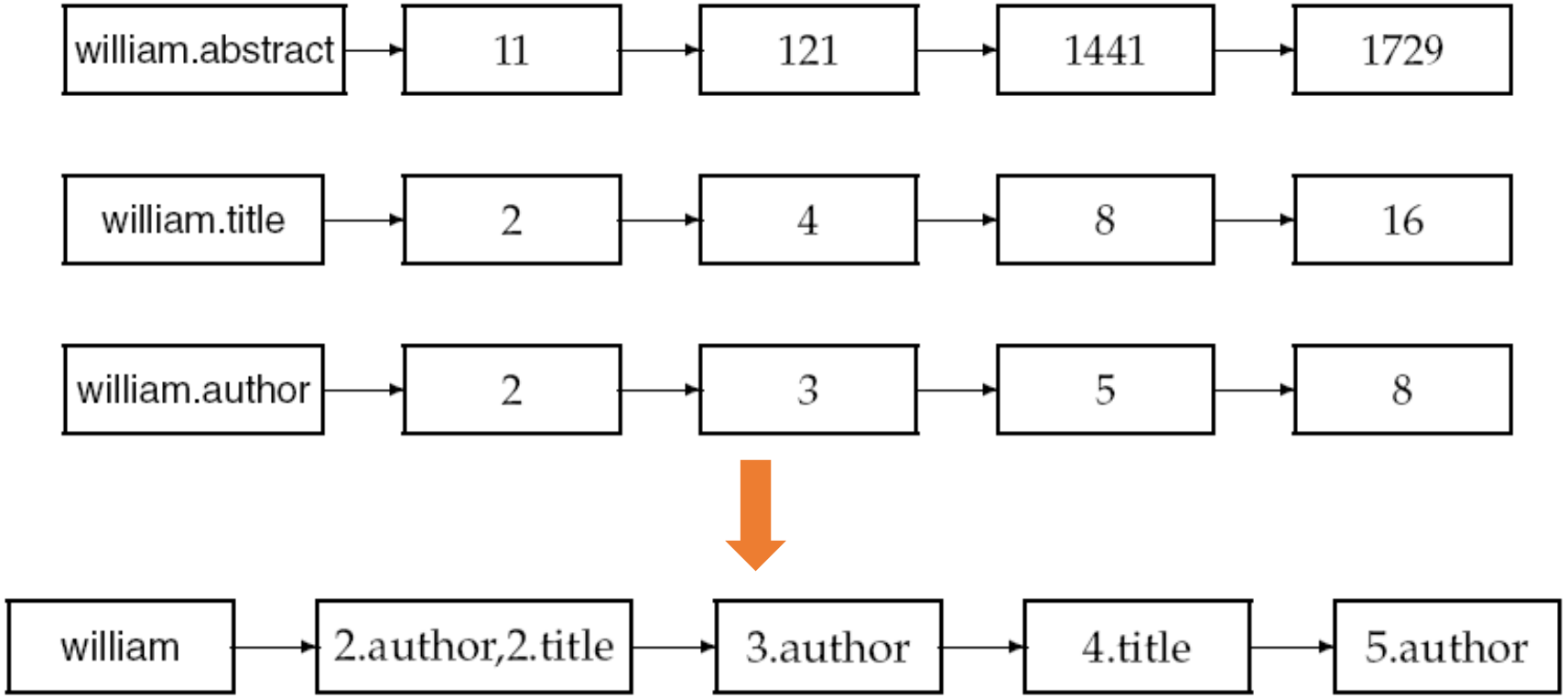


Zone Index Example

Is there a better way to index zones and fields?

We can do better…

- Encode zones in postings.



Weighted Zones

- Not all zones are equally important!
- Consider a collection where documents have three zones (l = 3):
  - author (least important)
  - title (more important)
  - body (most important)
- We can associate a weight, g_i to each zone
  - author (g_1 = 0.2)
  - title (g_2 = 0.3)
  - body (g_3 = 0.5)

$$\sum_{i=1}^{l} g_i = 1$$

$$g_i \in [0,1]$$

Weighted Zone Scoring

- If all query terms appear in ith zone,
  - we say s_i = 1.
- Then, we score the document as

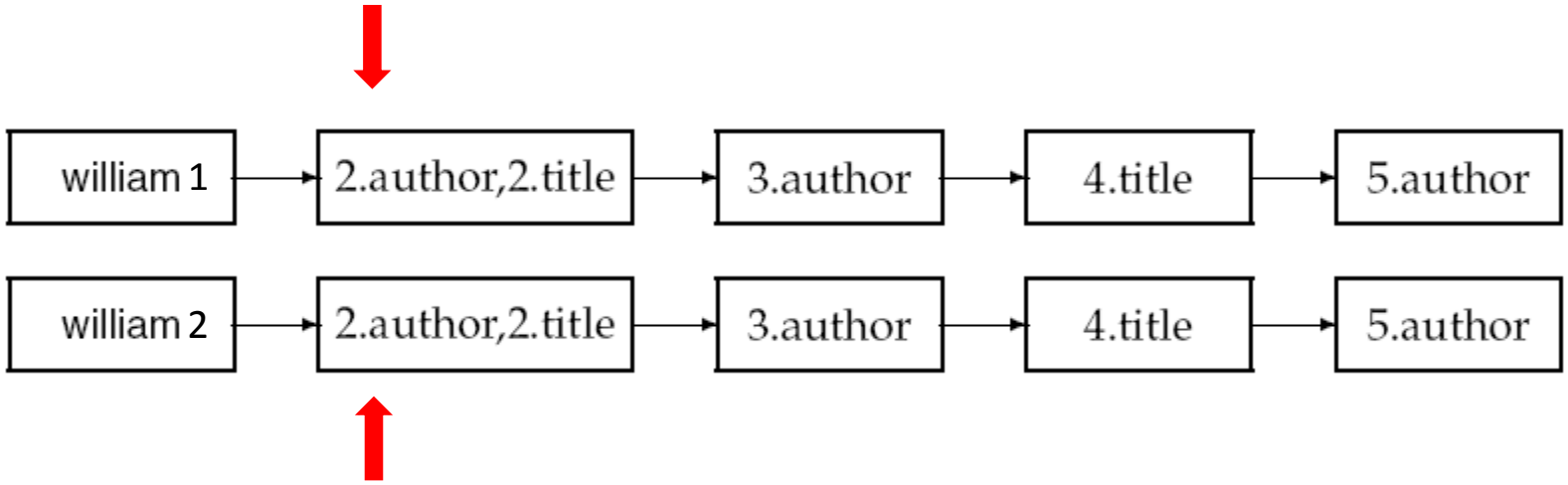
$$\sum_{i=1}^{l} g_i \, s_i$$

Quiz

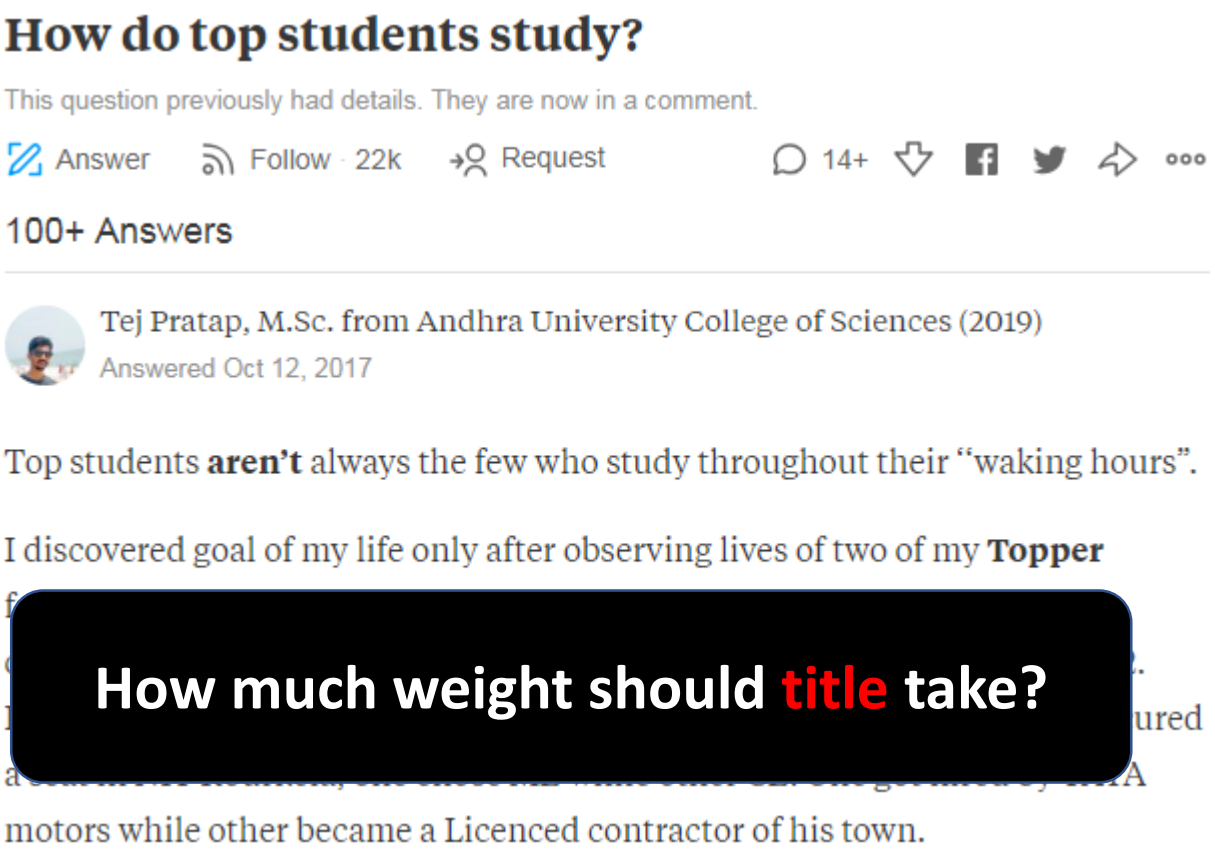
- Consider a collection with the following zone weights
  - author ($g_1 = 0.2$)
  - title ($g_2 = 0.3$)
  - body ($g_3 = 0.5$)
- If the term *Shakespeare* were to appear in the title and body zones but not in author zone, the score of the document would be **0.8** .

Weighted Zone Scoring on Inverted Index

A Match Found? Add g_i to an array **score[docID]**.
This array is often called **Accumulator**.



How to assign zone weights?



Machine Learned Relevance

- Use human annotated training examples
- Consider:
  - Only two zones exist: title and body.
  - $S_T(d,q)$= 1 if query term exists in title.
  - $S_B(d,q)$= 1 if query term exists in body.

Example	DocID	Query	s_T	s_B	Judgment
Φ_1	37	linux	1	1	Relevant
Φ_2	37	penguin	0	1	Non-relevant
Φ_3	238	system	0	1	Relevant
Φ_4	238	penguin	0	0	Non-relevant
Φ_5	1741	kernel	1	1	Relevant
Φ_6	2094	driver	0	1	Relevant
Φ_7	3191	driver	1	0	Non-relevant

Learning Weights (without ML)

Query	DocID	User Judgment
linux	37	1
system	238	1
kernel	1741	1
driver	2094	1

Judgment of 1 implies the document is relevant.

Query	DocID	In Title? (S_T)	In Body? (S_B)	Our Score
linux	37	1	1	?
system	238	0	1	?
kernel	1741	1	1	?
driver	2094	0	1	?

Assume zone weights: title ($g_1 = 0.3$) and body ($g_2 = 1 - 0.3 = 0.7$)

Learning Weights

Query	DocID	User Judgment
linux	37	1
system	238	1
kernel	1741	1
driver	2094	1

Query	DocID	In Title? (S_T)	In Body? (S_B)	Our Score
linux	37	1	1	1
system	238	0	1	0.7
kernel	1741	1	1	1
driver	2094	0	1	0.7

Quiz

- If g is the title weight, how to quantify the error?

Query	DocID	User Judgment
linux	37	1
system	238	1
kernel	1741	1
driver	2094	1

s_T	s_B	Score
0	0	0
0	1	1 - g
1	0	g
1	1	1

Query	DocID	In Title? (S_T)	In Body? (S_B)	Our Score
linux	37	1	1	1
system	238	0	1	1 – g
kernel	1741	1	1	1
driver	2094	0	1	1 - g

Learning Weights

$$score = \text{g}.sT + (1 - \text{g}).\text{sB}$$

Then the error,

$$\epsilon = (\text{relevance} - score)^2$$

Our objective is to find g such that we minimize the total error,

$$\sum_{all\ documents} \epsilon$$

For the 01 Case…

- How to quantify the error?

Query	DocID	User Judgment
linux	37	1
system	238	1
kernel	1741	1
driver	2094	1

s_T	s_B	Score
0	0	0
0	1	1 - g
1	0	g
1	1	1

Query	DocID	In Title? (S_T)	In Body? (S_B)	Our Score
linux	37	1	1	1
system	238	0	1	1 – g
kernel	1741	1	1	1
driver	2094	0	1	1 - g

*What if we have non-relevant judgments also?

For 01 case…

Error = $[1-(1-g)]^2 n_{01F} + [0-(1-g)]^2 n_{01R}$.

Total Error

$$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1-g)^2 + n_{00r} + n_{11n}.$$

Can you guess the optimal value of g for which the total error is minimum?

Total Error

$$(n_{01r} + n_{10n})g^2 + (n_{10r} + n_{01n})(1-g)^2 + n_{00r} + n_{11n}.$$

Differentiate w.r.t g and equate to zero

$$\frac{n_{10r} + n_{01n}}{n_{10r} + n_{10n} + n_{01r} + n_{01n}}$$

Thank You!