11-442 / 11-642 / 11-742:
Search Engines

Document Structure

Jamie Callan
Carnegie Mellon University
callan@cs.cmu.edu

1

Outline

Document structure

- Index support for structure
- Fields
- Multiple representations of meaning
- Hierarchical structure ("XML documents")

33

33

Hierarchical Document Structure

Hierarchical document structure raises some of the same issues that were seen with flat document structures

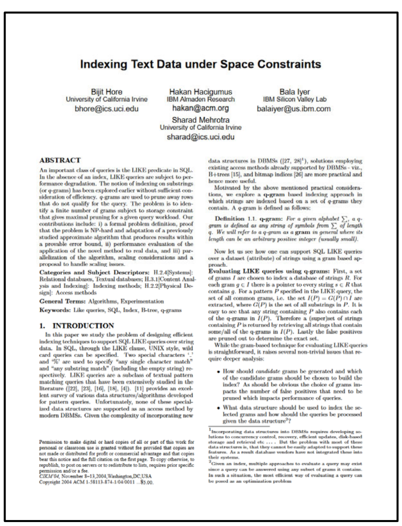
- Manual queries vs. automatic queries
- Independent evidence vs. different perspectives

Implemented using forward indexes or inverted lists that store document structure

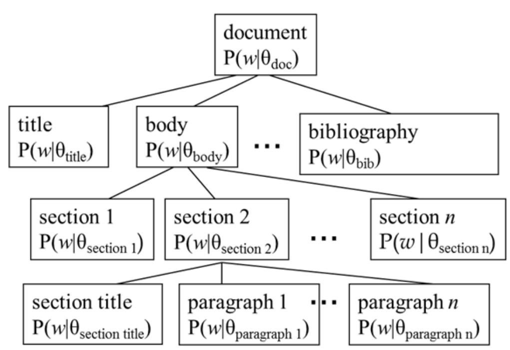
Much of the work on hierarchical document structure has been done in Bayesian inference networks

- Indri

Hierarchical Document Structure

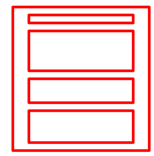

Indexing Text Data under Space Constraints

Bijit Hore
University of California Irvine
bhore@ics.uci.edu
Hakan Hacigumus
IBM Almaden Research
hakan@acm.org
Bala Iyer
IBM Silicon Valley Lab
balaiyer@us.ibm.com
Sharad Mehrotra
University of California Irvine
sharad@ics.uci.edu



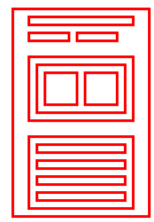
Structured Documents: Hierarchical Structure

Documents with fields and multiple representations are simple uses of document structure



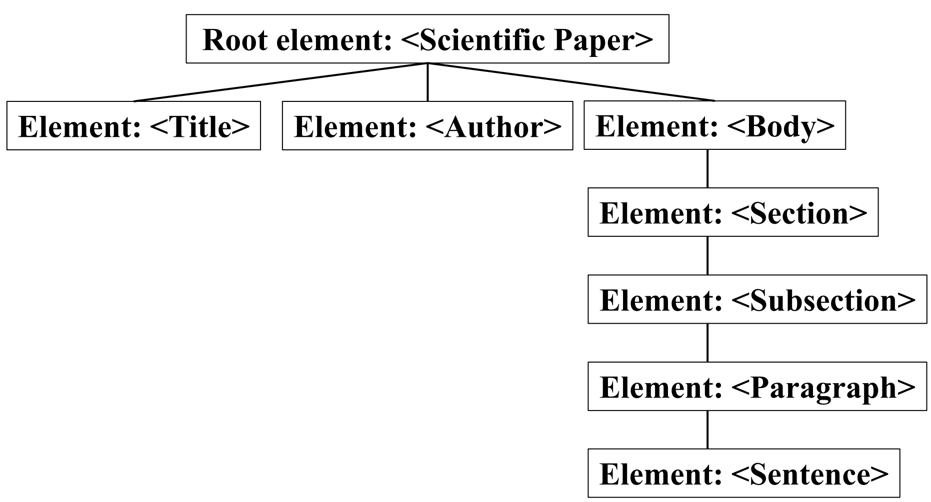
- Each element is independent of other elements
- A term occurs in the element that contains it
- **Retrieval goal:** Retrieve a document

Documents with hierarchical fields are a more advanced use of document structure

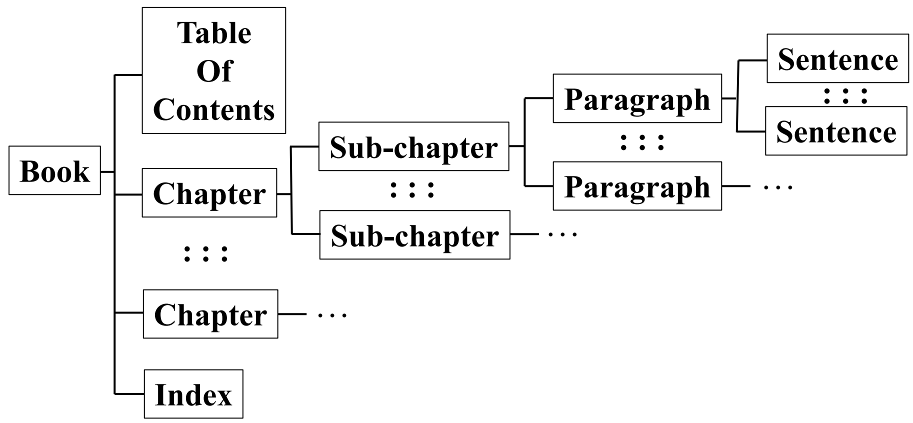


- Elements are related to other elements
- A term occurs in all elements that contain it
- **Retrieval goal:** Retrieve documents or elements

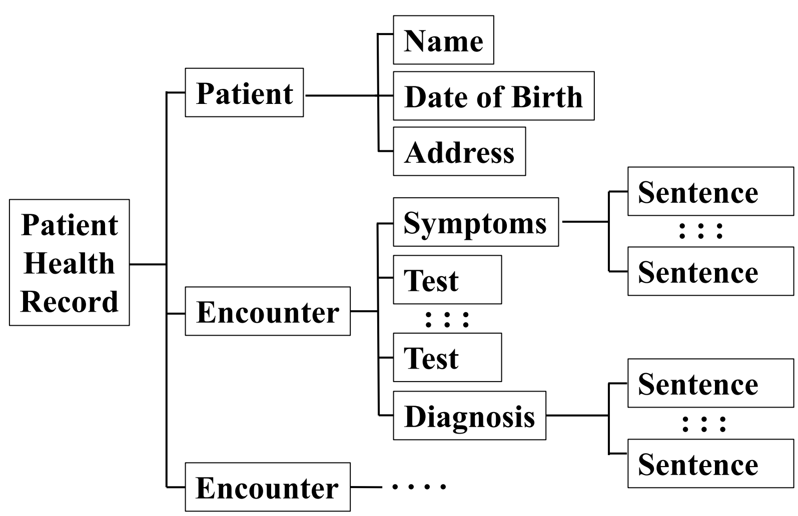
Structured Documents: Example Hierarchical Structures



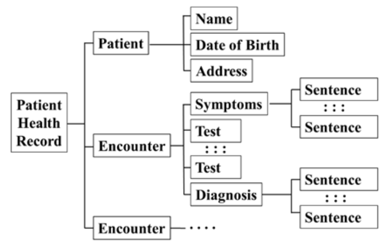
Structured Documents: Example Hierarchical Structures



Structured Documents: Example Hierarchical Structures



Structured Documents: New Issues Raised by Hierarchical Structure



What type of <u>element</u> to retrieve?

- Document? Encounter? Diagnosis?

What corpus statistics to use?

- Document vs. element

How to combine evidence from different elements?

- If the query is "chest pain", should we give higher rank to patients that have <u>several</u> matching sentences or encounters?

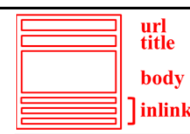
Exact-match vs. best-match document structure?

- Perhaps the query doesn't exactly match the document structure

40

40

Hierarchical Structure: Ranking Elements



Indri allows the <u>ranked element</u> to be specified in the query

- Any belief ("score") operator can rank any type of element
- Syntax: `#OPERATOR[element](argument` …)
- If no element is specified, the default is `Document`

Queries that rank different types of elements

- Documents: #AND(breast cancer treatment)
- Paragraphs: #WAND[paragraph](3 breast 4 cancer 2 treatment)
- Sentences: #AND[sentence](breast cancer treatment)

The result is a list of <u>elements</u> and <u>scores</u>

- I.e., a score list (element_i, score_i)

41

41

Hierarchical Structure: Ranking Elements

How is element e ranked?

- I.e., what is $p(q_i \mid e)$?

One option: Use Jelinek-Mercer smoothing

$$p(q_i \mid e) = (1-\lambda) p_{MLE}(q_i \mid e) + \lambda \, p_{MLE}(q_i \mid C)$$

Typically Dirichlet smoothing isn't used for element ranking

- Probably more a historical accident than a justified choice

Hierarchical Structure: Ranking Elements

Common problems in ranking elements

- The document structure is wrong (or broken)
  - Common with web documents
- The query is too strict
  - #AND[title](iphone) doesn't match "Apple Cuts Phone Price"
  - #AND[paragraph](solutions to poverty)
    "poverty" and "solutions" may appear in different paragraphs

Additional smoothing reduces the effect of these problems

$$p(q_i \mid e) = \underbrace{\lambda_1 p_{MLE}(q_i \mid e)}_{\text{Element}} + \underbrace{\lambda_2 p_{MLE}(q_i \mid d)}_{\text{Document}} + \underbrace{\lambda_3 p_{MLE}(q_i \mid C)}_{\text{Collection}} \quad \lambda_1+\lambda_2+\lambda_3=1$$

Hierarchical Structure: Documents with Multiple Elements

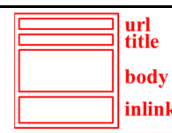
Often the goal is to use evidence from one type of element (e_1) to rank another type of element (e_2)

- E.g., Find web pages that have 'Jamie Callan' as inlink text

How is evidence from different types of elements combined?

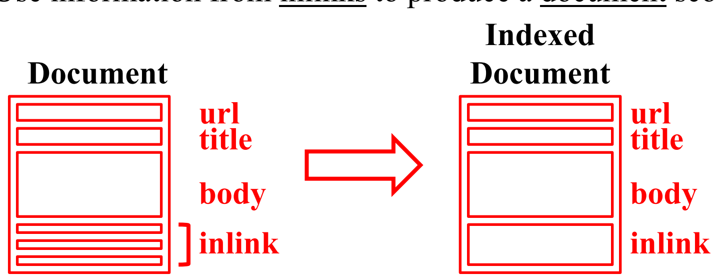
1. Aggregation
   - Can be done during indexing (e.g., Lucene)
   - Can be done during query evaluation (e.g., Indri)
2. Combination
   - Can be done during query evaluation (e.g., Indri)

Hierarchical Structure: Aggregation During Indexing



Goal: Find web pages that have 'apple ipad' as inlink text

- Use information from inlinks to produce a document score



- Combine ("aggregate") multiple instances of an element type (e.g., inlink) into one bag of words
- Okapi BM25F and Lucene do this

Hierarchical Structure: Aggregation During Querying

Goal: Find <u>web pages</u> that have ‘apple ipad’ as <u>inlink</u> text

- Use information from <u>inlinks</u> to produce a <u>document</u> score

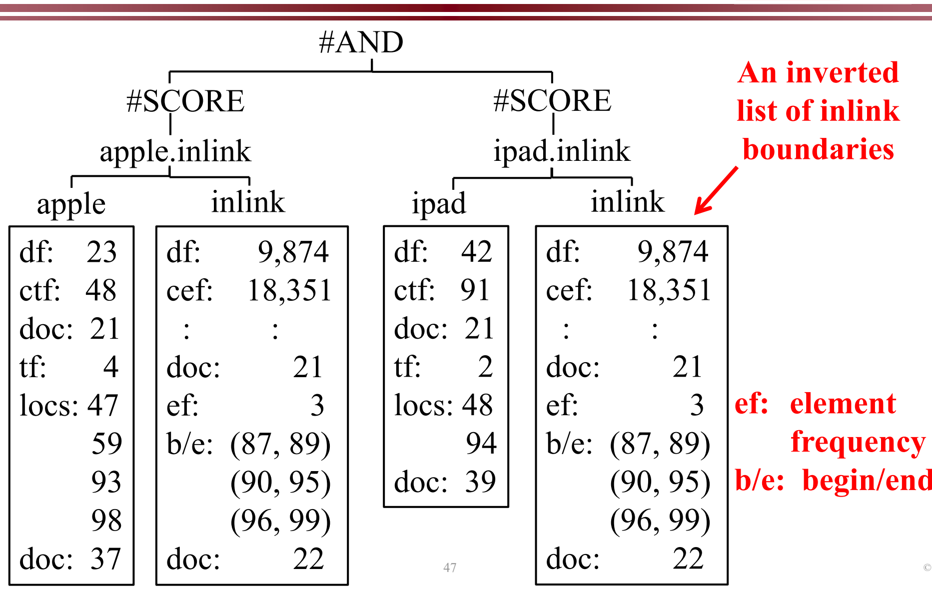
Indri queries can aggregate <u>element</u> information into <u>document</u> information during query evaluation

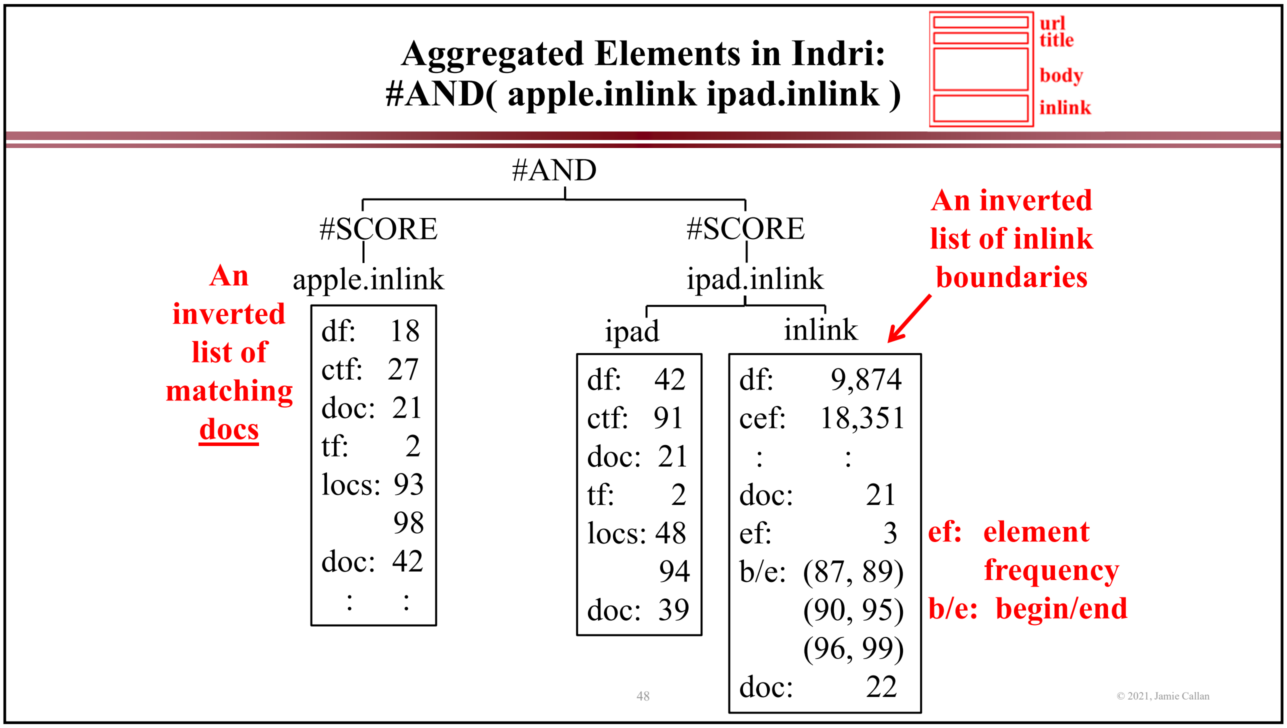
- <u>apple.inlink</u> aggregates all occurrences of apple in inlink elements
  - Implemented as a QryIop operator
  - Result is one <u>inverted list</u> for apple as an inlink term

Example

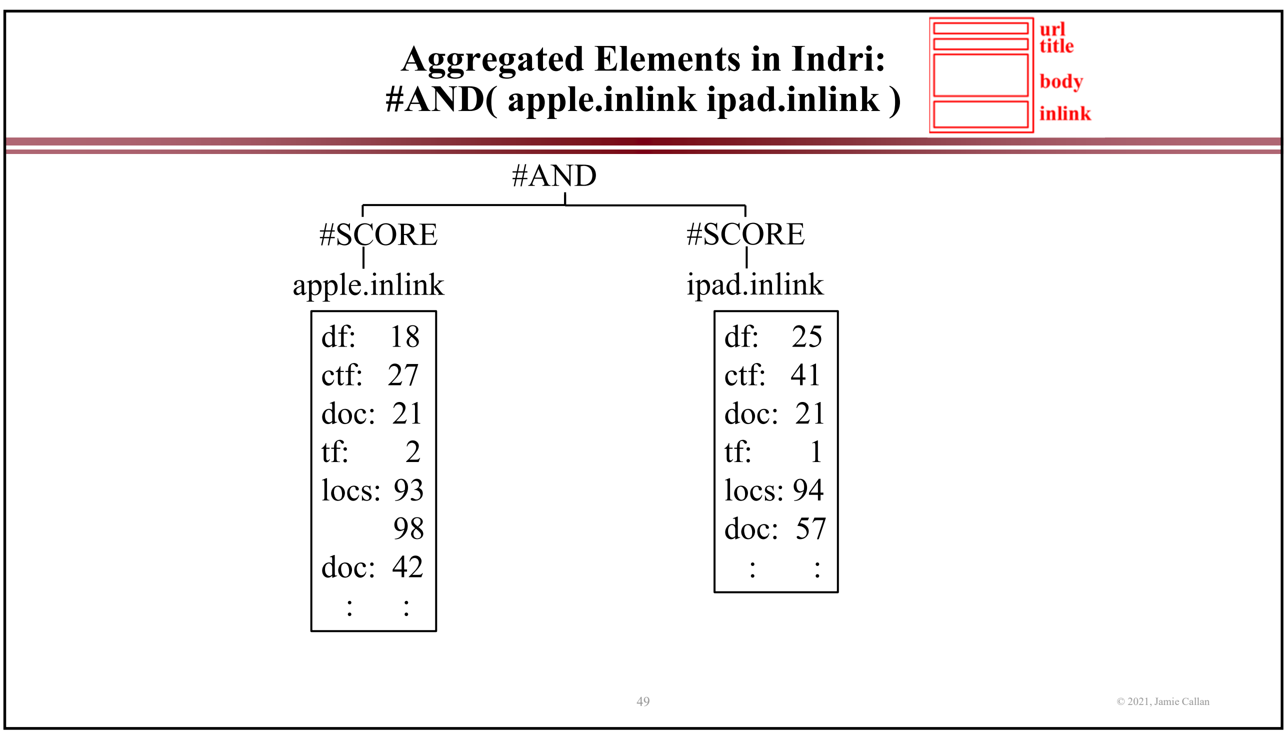
#WAND(#WSUM (0.3 apple.title 0.2 apple.inlink 0.5 apple.body)
#WSUM (0.3 ipad.title 0.2 ipad.inlink 0.5 ipad.body))

Aggregated Elements in Indri: #AND(apple.inlink ipad.inlink)




Aggregated Elements in Indri:
#AND(apple.inlink ipad.inlink)
url
title
body
inlink
#AND
#SCORE
apple.inlink
#SCORE
ipad.inlink
ipad
inlink
An inverted list of matching docs
df: 18
ctf: 27
doc: 21
tf: 2
locs: 93
98
doc: 42
: :
df: 42
ctf: 91
doc: 21
tf: 2
locs: 48
94
doc: 39
df: 9,874
cef: 18,351
: :
doc: 21
ef: 3
b/e: (87, 89)
(90, 95)
(96, 99)
doc: 22
An inverted list of inlink boundaries
ef: element frequency
b/e: begin/end
48
© 2021, Jamie Callan

48


Aggregated Elements in Indri:
#AND(apple.inlink ipad.inlink)
url
title
body
inlink
#AND
#SCORE
apple.inlink
#SCORE
ipad.inlink
df: 18
ctf: 27
doc: 21
tf: 2
locs: 93
98
doc: 42
: :
df: 25
ctf: 41
doc: 21
tf: 1
locs: 94
doc: 57
: :
49
© 2021, Jamie Callan

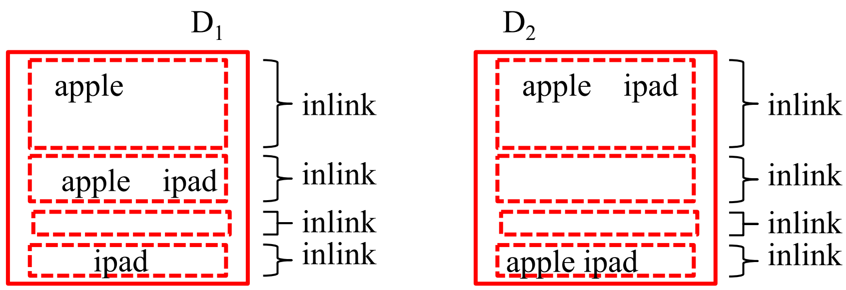
49

Hierarchical Structure: Aggregated Elements

Goal: Rank documents that have inlinks containing apple ipad

#AND(apple.inlink ipad.inlink)

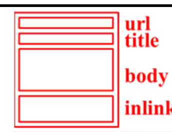
- Aggregation makes these documents look the same to the query
  – Probably okay, because inlinks are short and typically used to rank documents



50

50

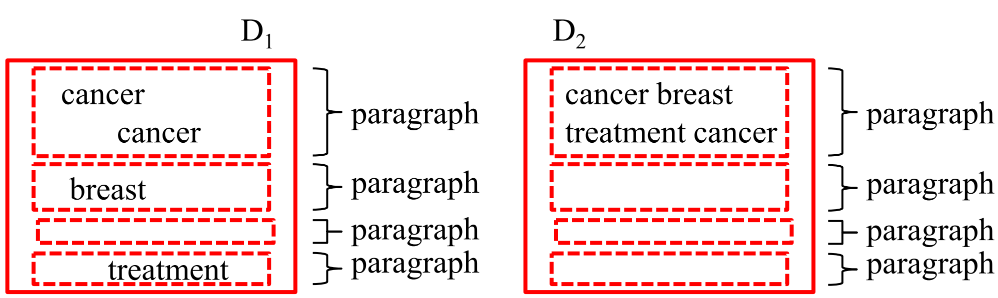
Hierarchical Structure: Aggregated Elements



Goal: Rank documents that have paragraphs that discuss breast cancer treatment

#AND(breast.paragraph cancer.paragraph treatment.paragraph)

- Aggregation makes these documents look the same to the query
  – Probably not okay, e.g., if returning paragraphs to the user



51

51

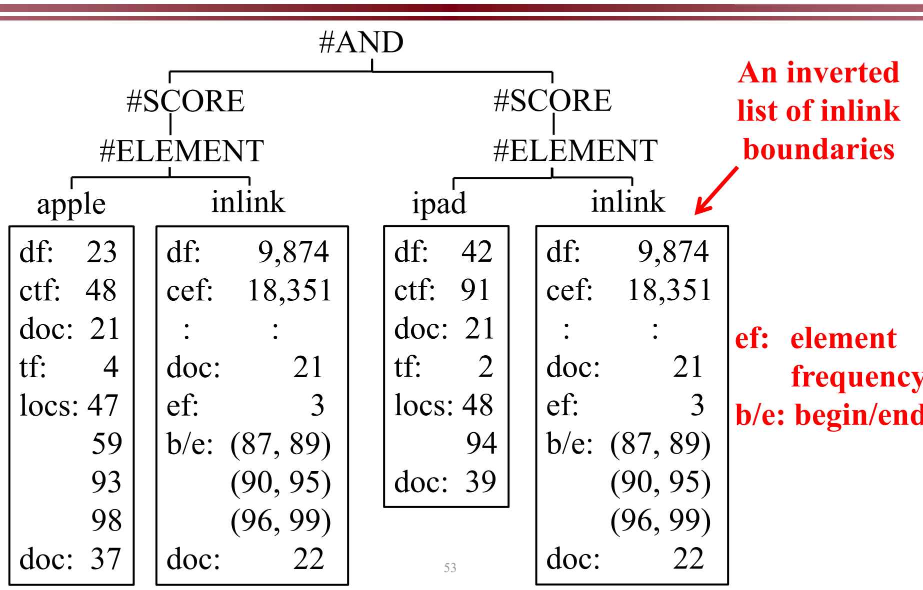
Hierarchical Structure: Combining Evidence

The alternative to aggregation is combining evidence from different elements of the same type

- **Aggregation:** Combine inverted lists
  - Use QryIop operators
- **Combination:** Combine scores
  - Use both QryIop and QrySop operators

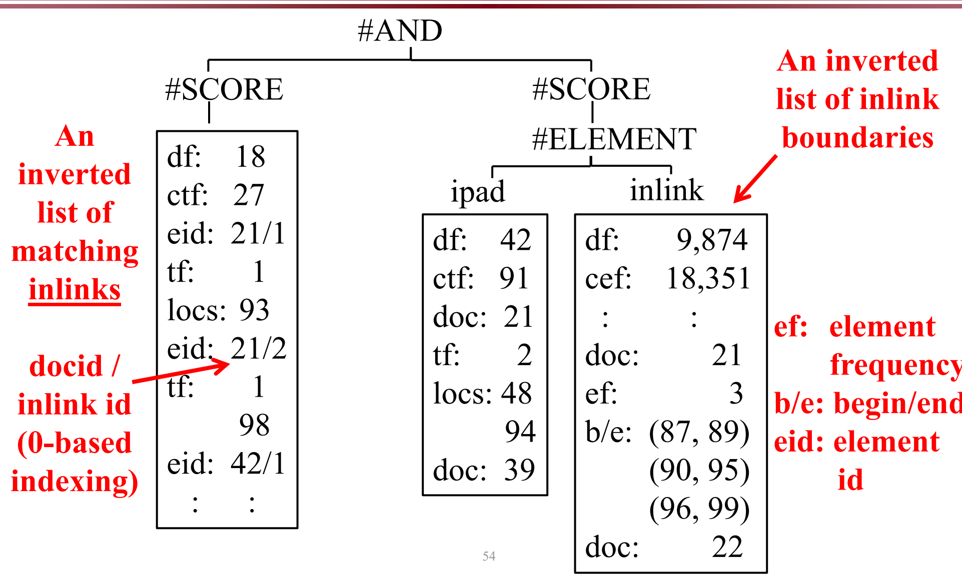
52

Combining Elements in Indri: #AND[inlink](apple ipad)

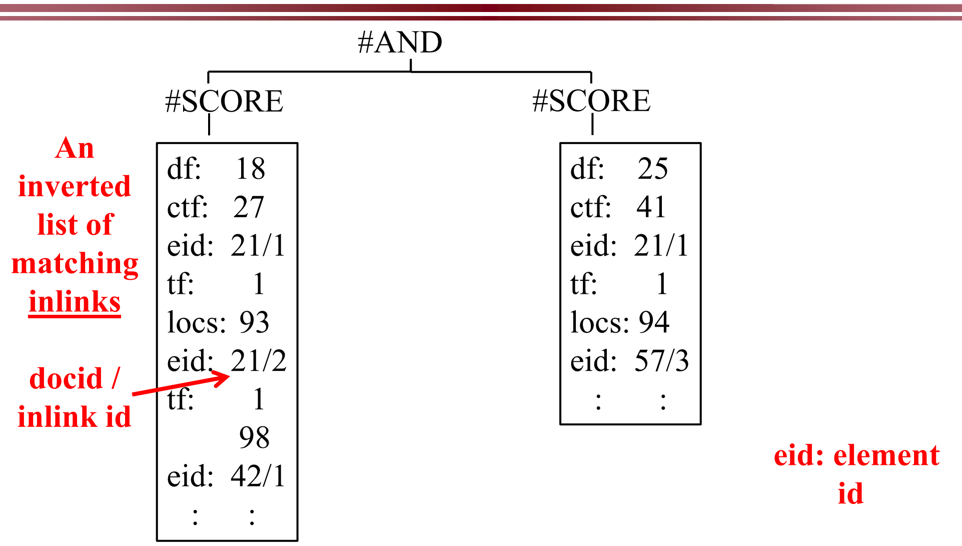


53

Combining Elements in Indri: #AND[inlink](apple ipad)



Combining Elements in Indri: #AND[inlink](apple ipad)



Combining Elements in Indri:
#AND[inlink](apple ipad)

#AND

df: 18
ctf: 27
21/1, 0.453
21/2, 0.467
42/1, 0.421
: :

A score list of matching <u>inlinks</u>

docid / inlink id **score**

#SCORE

df: 25
ctf: 41
eid: 21/1
tf: 1
locs: 94
eid: 57/3
: :

eid: element id

Combining Elements in Indri:
#AND[inlink](apple ipad)

#AND

df: 18
ctf: 27
21/1, 0.453
21/2, 0.467
42/1, 0.421
: :

A score list of matching <u>inlinks</u>

docid / inlink id **score**

df: 25
ctf: 41
21/1, 0.673
57/3, 0.597
: :

Combining Elements in Indri: #AND[inlink](apple ipad)

A score list of matching <u>inlinks</u>

df: 25
ctf: 41
21/1, 0.305
21/2, 0.006
42/1, 0.005
57/3, 0.014
: :

docid / inlink id **score**

If the goal is to retrieve a <u>ranked list of inlinks</u>, sort the list by score and display it

- Done!

Combining Elements in Indri

The previous example is a little unusual

- Retrieving individual inlinks is an unusual task
- Purpose: To provide a clear contrast with aggregation

The <u>technique</u> in the previous example is general and useful

- E.g., retrieving <u>sentences</u> or <u>passages</u> for Alexa
- E.g., retrieving <u>sections</u> of a long scientific paper

Hierarchical Structure: Combining Element Scores

Suppose the goal is to use inlink scores to rank documents

#SUM[document] (#AND[inlink](apple ipad))

#SUM-ELEMENTS
(to document)

#AND

#SCORE — #ELEMENT — apple, inlink

#SCORE — #ELEMENT — ipad, inlink

apple	inlink	ipad	inlink
df: 23	df: 9,874	df: 42	df: 9,874
ctf: 48	cef: 18,351	ctf: 91	cef: 18,351
doc: 21	: :	doc: 21	: :
tf: 4	doc: 21	tf: 2	doc: 21

Hierarchical Structure: Combining Element Scores

Suppose the goal is to use inlink scores to rank documents

#SUM[document] (#AND[inlink](apple ipad))

#SUM-ELEMENTS
(to document)

df: 25
ctf: 41
21/1, 0.305
21/2, 0.006
42/1, 0.005
57/3, 0.014
: :

The result of #AND[inlink](apple ipad) (shown on earlier slides)

Hierarchical Structure: Combining Element Scores

Suppose the goal is to use inlink scores to rank documents

#SUM[document] (#AND[inlink](apple ipad))

#SUM-ELEMENTS
(to document)

df: 25
ctf: 41
21/1, 0.305
21/2, 0.006
42/1, 0.005
57/3, 0.014
: :

The result of
#AND[inlink](apple ipad)
(shown on earlier slides)

#SUM-ELEMENTS
sums child scores (inlink)
to produce parent scores (document)

62

62

Hierarchical Structure: Combining Element Scores

Suppose the goal is to use inlink scores to rank documents

#SUM[document] (#AND[inlink](apple ipad))

#SUM-ELEMENTS
(to document)

df: 25
ctf: 41
21/1, 0.305
21/2, 0.006
42/1, 0.005
57/3, 0.014
: :

Two inlinks for docid 21 (21/1, 21/2)

The result of
#AND[inlink](apple ipad)
(shown on earlier slides)

#SUM-ELEMENTS
sums child scores (inlink)
to produce parent scores (document)

63

63

Hierarchical Structure: Combining Element Scores

Suppose the goal is to use inlink scores to rank documents

#SUM[document] (#AND[inlink](apple ipad))

	df: 25
	ctf: 32
Docid 21	21, 0.311
	42, 0.005
	57, 0.014
	: :

Hierarchical Structure: Combining Element Scores

Indri queries must specify the combination method

- The right method is problem-specific

Return documents that have many matching sentences

- #MAX[document] (#AND[sentence](breast cancer treatment))
  - Only the best sentence is considered
- #SUM[document] (#AND[sentence](breast cancer treatment))
  - Poorly matching sentences reduce the score
- #OR[document] (#AND[sentence](breast cancer treatment))
  - Prefer documents with more matching sentences

Hierarchical Structure: Combining Element Scores

What does this query mean?

#SUM[document] (#AND[sentence] (breast cancer treatment))

Your software uses depth-first evaluation

1. Apply #AND (breast cancer treatment) to every <u>sentence</u>
   – The result is a list of <sentence, score>
2. Apply #SUM to the <u>sentence scores</u> produced in step 1.
   – Note: <u>Different</u> #SUM operator semantics than HW2 #SUM
   – The score for a document is the sum of its sentence scores
   – The result is <document, score>

The query returns a list of <document, score>

66 © 2021, Jamie Callan

66

Hierarchical Structure: Combining Element Scores

Consider 4 <u>chapters</u> that have <u>sentences</u> matching the query

…which is the best match?

- #MAX considers them equal
- #AND prefers C_4
- #OR prefers C_1
- #AVERAGE prefers C_4

C_1	C_2	C_3	C_4
0.010	0.010	0.010	0.010
0.009	0.009	0.009	
0.001	0.001		
0.001			

It may seem "obvious" that #OR is the best choice

…but, the best choice is problem-dependent

67 © 2021, Jamie Callan

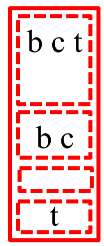
67

Hierarchical Structure: Elements in Indri

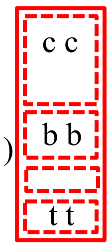
url
title
body
inlink

Suppose the goal is to retrieve a <u>document</u> that has <u>paragraphs</u> discussing breast cancer treatment

#OR[document] (
    #AND[paragraph] (
        breast
        cancer
        treatment))



#AND[document] (
    breast.paragaph
    cancer.paragraph
    treatment.paragraph)



Combination: Partial credit for paragraphs that partially match

Aggregation: The terms might not be in the <u>same</u> paragraph

Outline

Document structure

- Index support for structure
- Fields
- Multiple representations of meaning
- Hierarchical structure (“XML documents”)

Summary

Fields

- Used for two very different purposes
  - – Independent evidence (author, title, journal, …)
  - – Multiple (related) representations (url, title, body, …)
- Know the difference
- Know how these are supported by each retrieval model
- Know how to these would be used differently in queries

70

70

Summary

Hierarchical structure

- How to combine evidence when several elements match
- Exact-match vs. best-match document structure
- Know how these are supported by Indri
- Know how to these would be used differently in queries

71

71

For Additional Information

- S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4). 2009.
- G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley. 1989.
- S. Walker, S.E. Robertson, M. Boughanem, G.J.F. Jones, K. Sparck Jones. "Okapi at TREC-6: Automatic ad hoc, VLC, routing, filtering, and QSDR. TREC-6 Proceedings. 1997.