Introduction to Information Retrieval & Web Search

Kevin Duh
Johns Hopkins University

June 2019

Acknowledgments

These slides draw heavily from these excellent sources:

- Paul McNamee’s JSALT2018 tutorial:
  - https://www.clsp.jhu.edu/wp-content/uploads/sites/75/2018/06/2018-06-19-McNamee-JSALT-IR-Soup-to-Nuts.pdf
- Doug Oard’s Information Retrieval Systems course at UMD
  - http://users.umiacs.umd.edu/~oard/teaching/734/spring18/
- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge U. Press. 2008.
  - https://nlp.stanford.edu/IR-book/information-retrieval-book.html
- W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson, 2009
  - http://ciir.cs.umass.edu/irbook/

I never waste memory on things that can easily be stored and retrieved from elsewhere.

-- Albert Einstein

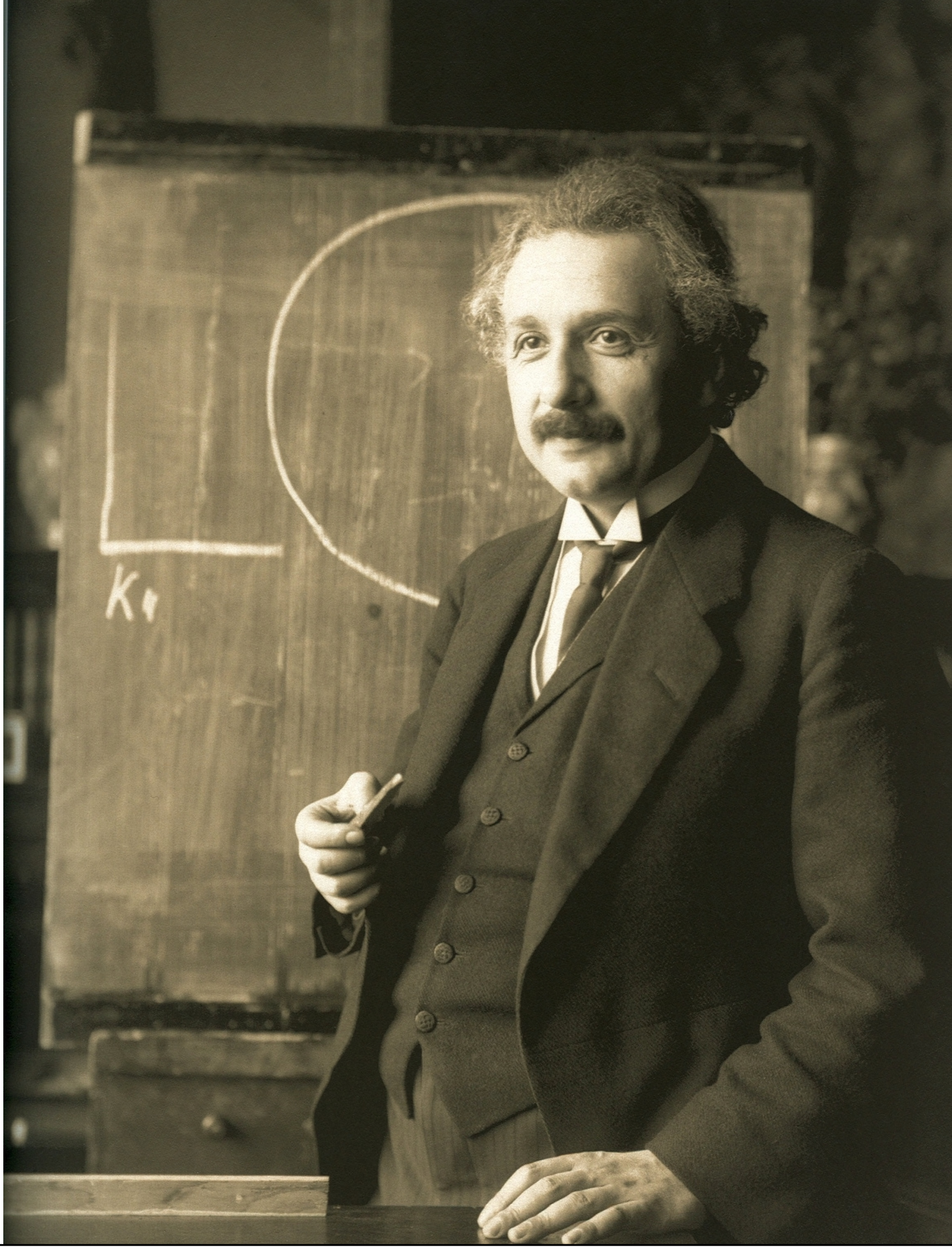


Image source: Einstein 1921 by F Schmutzer

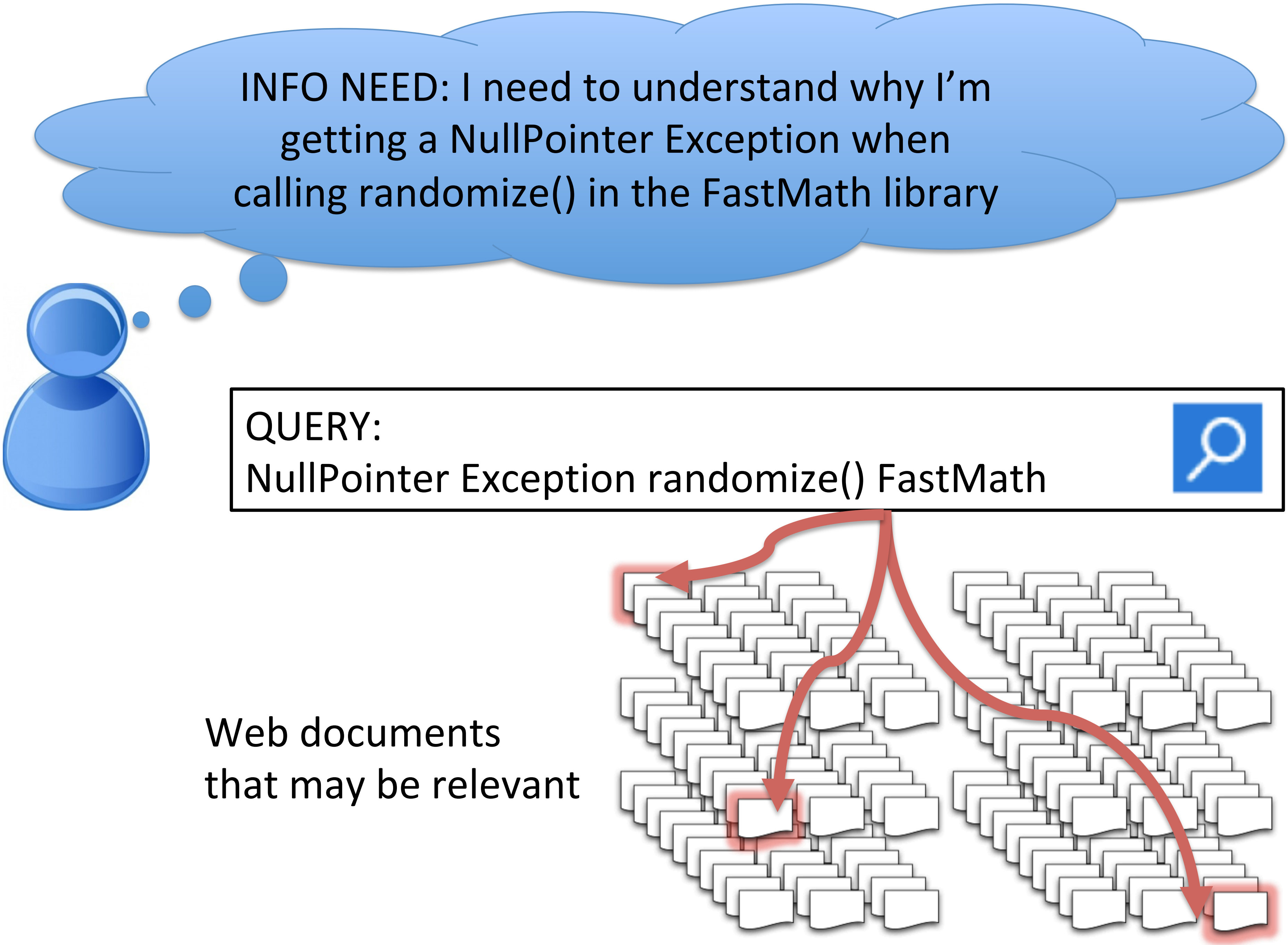
https://en.wikipedia.org/wiki/Albert_Einstein#/media/File:Einstein_1921_by_F_Schmutzer_-_restoration.jpg

What is Information Retrieval (IR)?

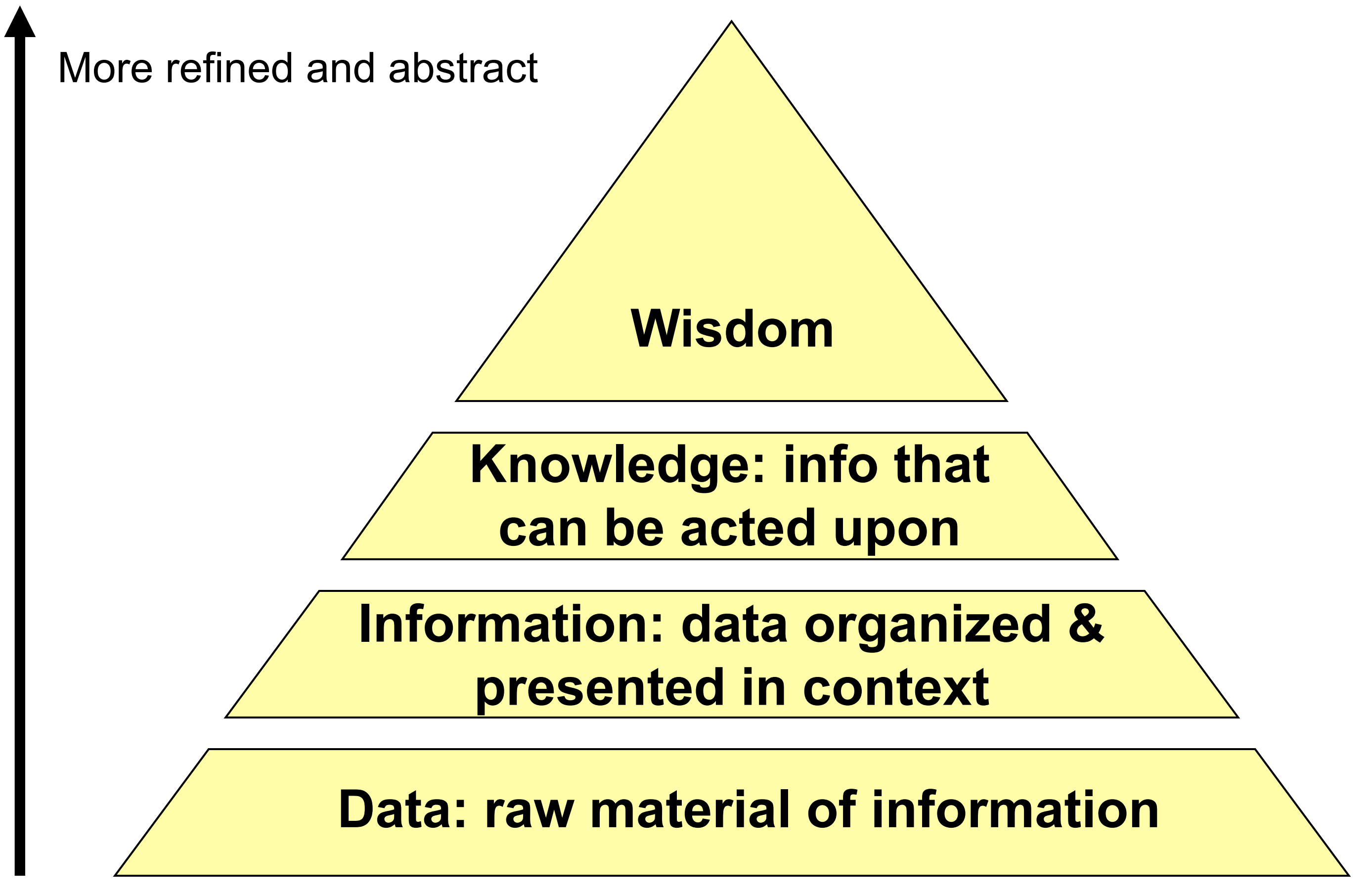
1. Information retrieval is a field concerned with the **structure, analysis, organization, storage, searching, & retrieval of information**.

   (Gerard Salton, IR pioneer, 1968)

2. Information retrieval focuses on the efficient recall of information that **satisfies a user's information need**.


INFO NEED: I need to understand why I'm getting a NullPointer Exception when calling randomize() in the FastMath library
QUERY:
NullPointer Exception randomize() FastMath
Web documents
that may be relevant

Information Hierarchy



From Doug Oard's slides: http://users.umiacs.umd.edu/~oard/teaching/734/spring18/

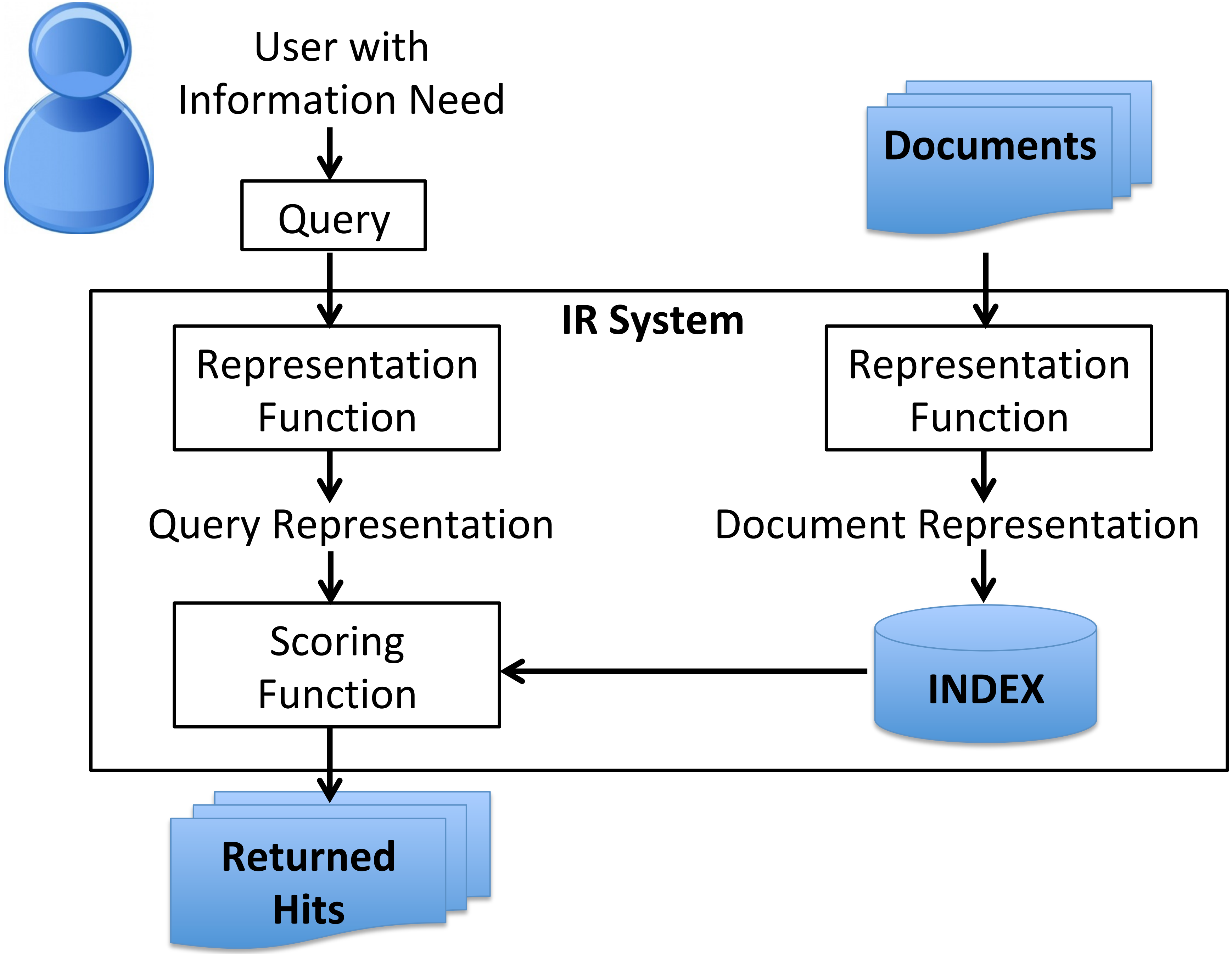
Databases vs. IR

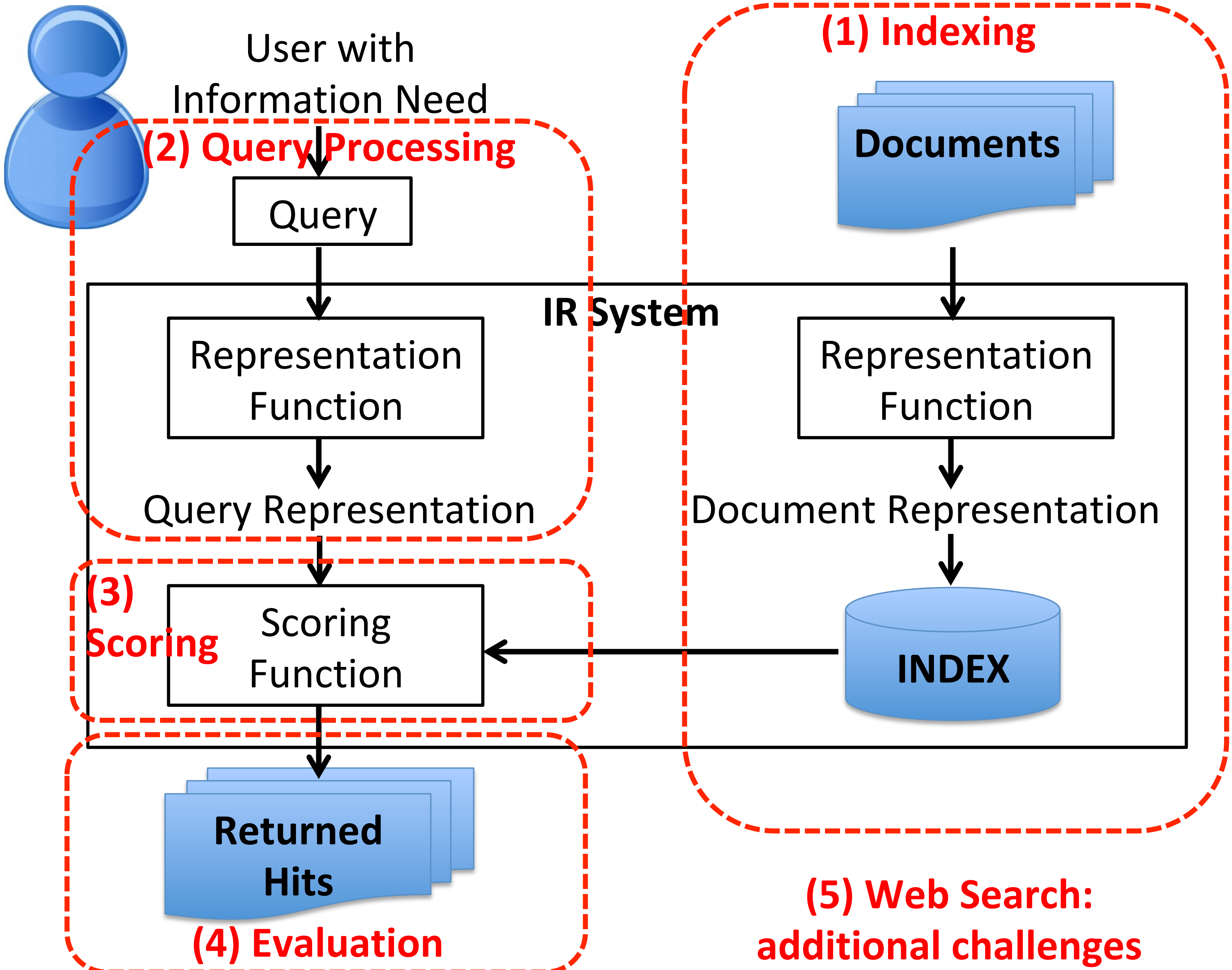
	Database	IR
What we're retrieving	Structured data. Clear semantics based on formal model.	Unstructured data. Free text with metadata. Videos, images, music.
Queries we're posing	Unambiguous formally defined queries.	Vague, imprecise queries
Results we get	Exact. Always correct in a formal sense.	Sometimes relevant sometimes not.

Note: From a user perspective, the distinction may be seamless, e.g. asking Siri a question about nearby restaurants w/ good reviews

From Doug Oard's slides: http://users.umiacs.umd.edu/~oard/teaching/734/spring18/

Structure of IR System & Tutorial Overview


User with
Information Need
Query
Documents
IR System
Representation Function
Representation Function
Query Representation
Document Representation
Scoring Function
INDEX
Returned Hits


User with
Information Need
(1) Indexing
Documents
(2) Query Processing
Query
IR System
Representation
Function
Representation
Function
Query Representation
Document Representation
(3)
Scoring
Scoring
Function
INDEX
Returned
Hits
(4) Evaluation
(5) Web Search:
additional challenges

Index vs Grep

- Say we have collection of Shakespeare plays
- We want to find all plays that contain:

QUERY:
Brutus AND Caesar AND NOT Calpurnia

- Grep: Start at 1st play, read everything and filter if criteria doesn't match (linear scan, 1M words)
- Index (a.k.a. Inverted Index): build index data structure off-line. Quick lookup at query-time.

These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

The Shakespeare collection as Term-Document Incidence Matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

Matrix element (t,d) is:

1 if term t occurs in document d,

0 otherwise

These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

The Shakespeare collection as Term-Document Incidence Matrix

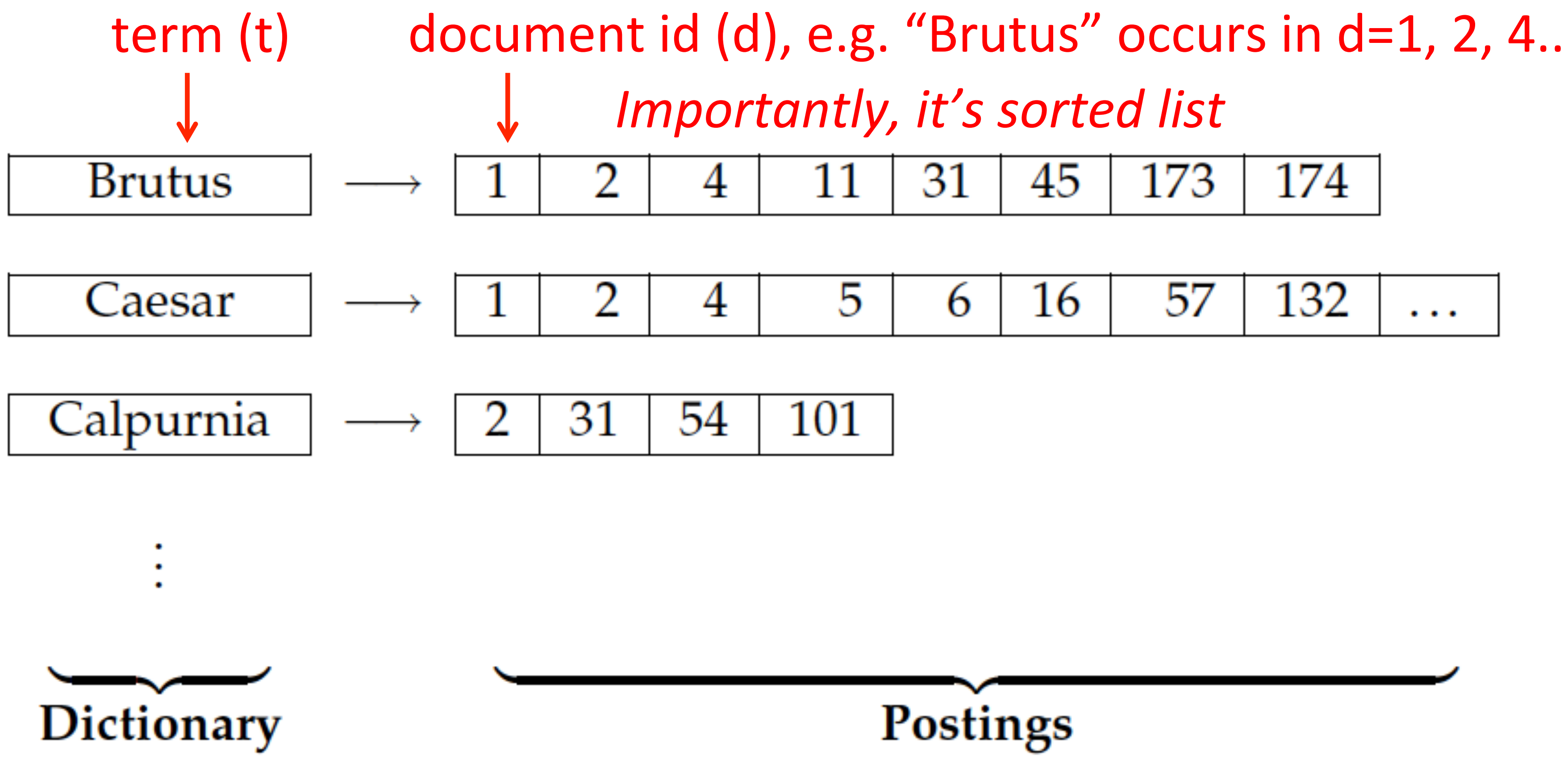
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

QUERY:
Brutus AND Caesar AND NOT Calpurnia

Answer: "Antony and Cleopatra"(d=1), "Hamlet"(d=4)

These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

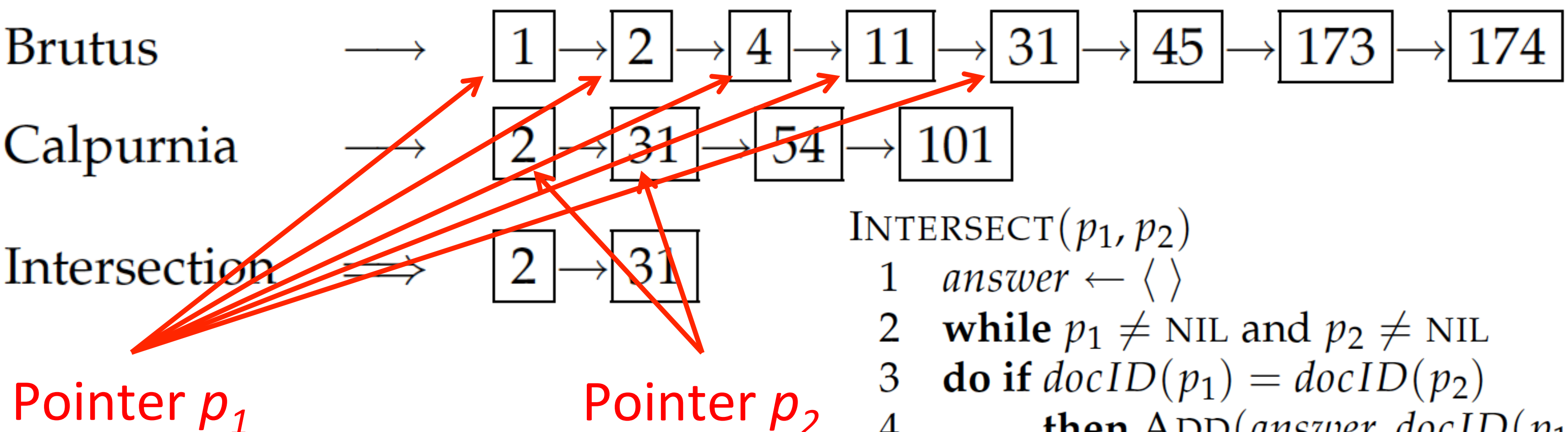
Inverted Index Data Structure



These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

Efficient algorithm for List Intersection (for Boolean conjunctive "AND" operators)

QUERY:
Brutus AND Calpurnia



INTERSECT(p_1, p_2)

```
answer ← ⟨ ⟩
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
     then ADD(answer, docID(p1))
          p1 ← next(p1)
          p2 ← next(p2)
     else if docID(p1) < docID(p2)
            then p1 ← next(p1)
            else p2 ← next(p2)
return answer
```

These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

Time and Space Tradeoffs

- Time complexity at query-time:
  - Linear scan over postings
  - $O(L_1 + L_2)$ where L_t is length of posting for term t
  - vs. grep through all documents $O(N)$, $L << N$
- Time complexity at index-time:
  - $O(N)$ for one pass through collection
  - Additional issue: efficient adding/deleting documents
- Space complexity (example setup):
  - Dictionary: Hash/Trie in RAM
  - Postings: Array on disk

Quiz: How would you process these queries?

QUERY:
Brutus AND Caesar AND Calpurnia

QUERY:
Brutus AND (Caesar OR Calpurnia)

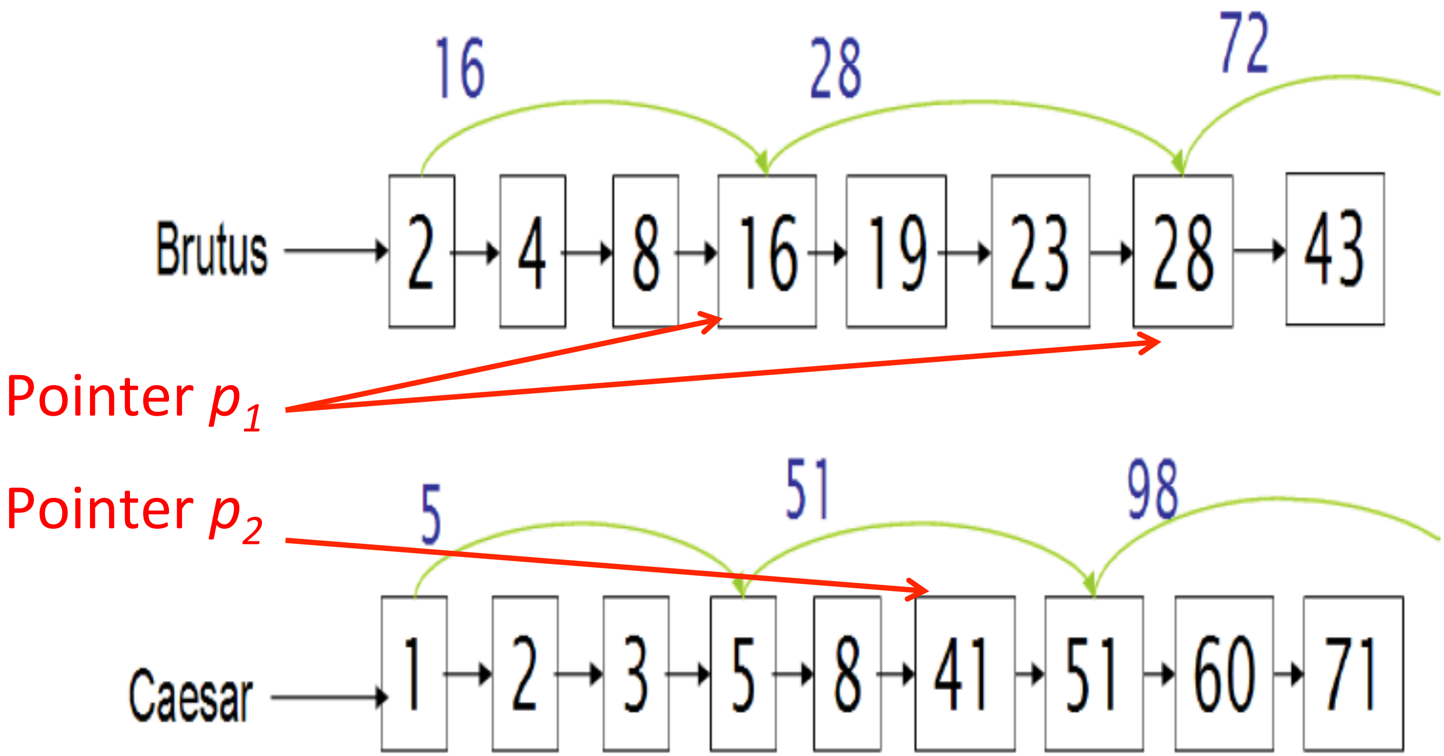
QUERY:
Brutus AND Caesar AND NOT Calpurnia

Brutus	⟶	1	2	4	11	31	45	173	174	
Caesar	⟶	1	2	4	5	6	16	57	132	…
Calpurnia	⟶	2	31	54	101					

Think: What terms to process first? How to handle OR, NOT?

Optional meta-data in inverted index

- Skip pointers: For faster intersection, but extra space



These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

Optional meta-data in inverted index

- Position of term in document: Enables phrasal queries

QUERY:
"to be or not to be"

$$\text{to}, 993427: \langle\, 1, 6: \langle 7, 18, 33, 72, 86, 231\rangle; 2, 5: \langle 1, 17, 74, 222, 255\rangle; 4, 5: \langle 8, 16, 190, 429, 433\rangle; 5, 2: \langle 363, 367\rangle; 7, 3: \langle 13, 23, 191\rangle; \ldots \rangle$$

$$\text{be}, 178239: \langle\, 1, 2: \langle 17, 25\rangle; 4, 5: \langle 17, 191, 291, 430, 434\rangle; 5, 3: \langle 14, 19, 101\rangle; \ldots \rangle$$

term (t) → be

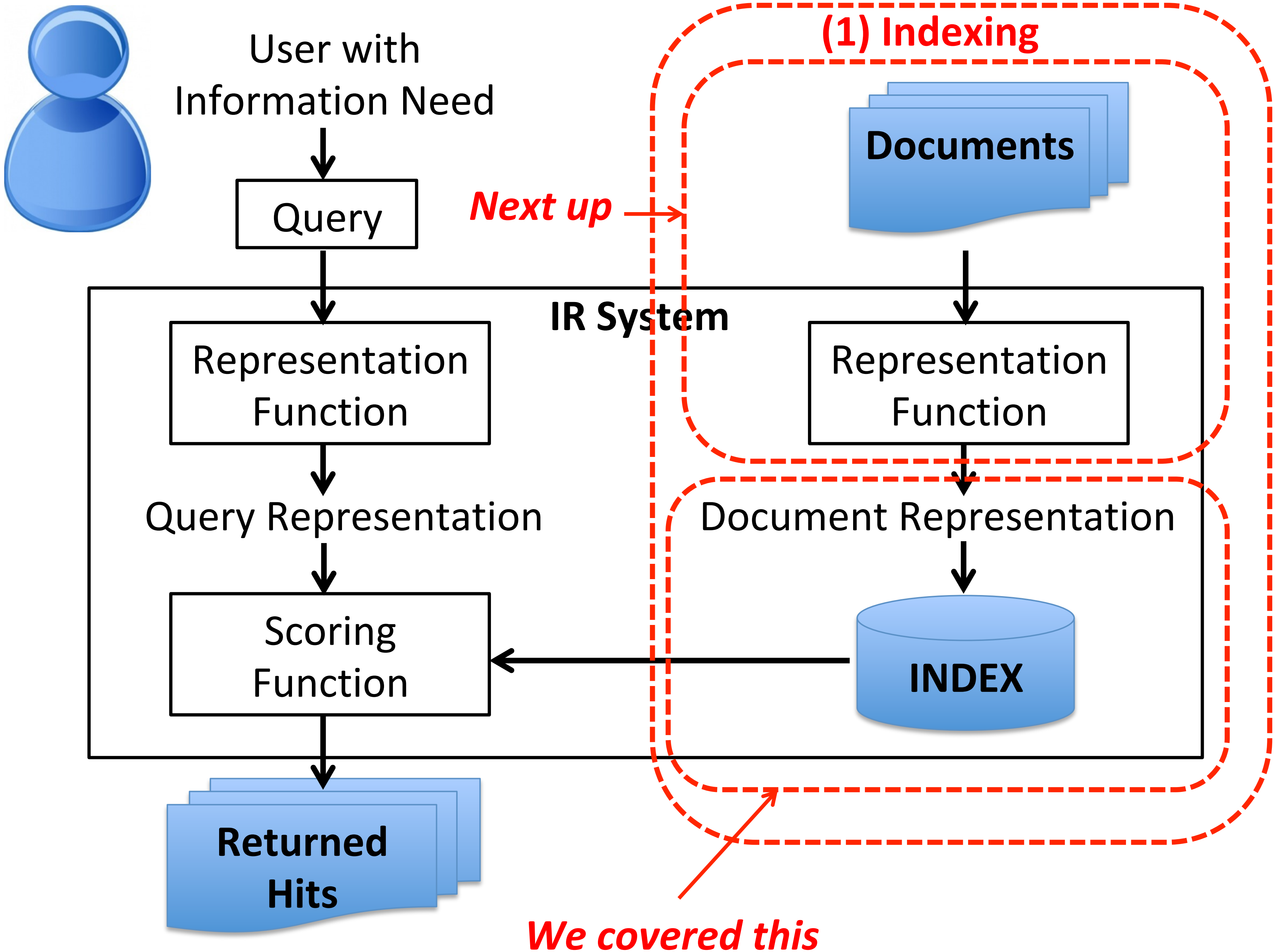
document frequency → 178239

term occurs in document d=4 with term frequency of 5, at positions 17, 191, 291, 430, 434 → 4, 5: ⟨17, 191, 291, 430, 434⟩

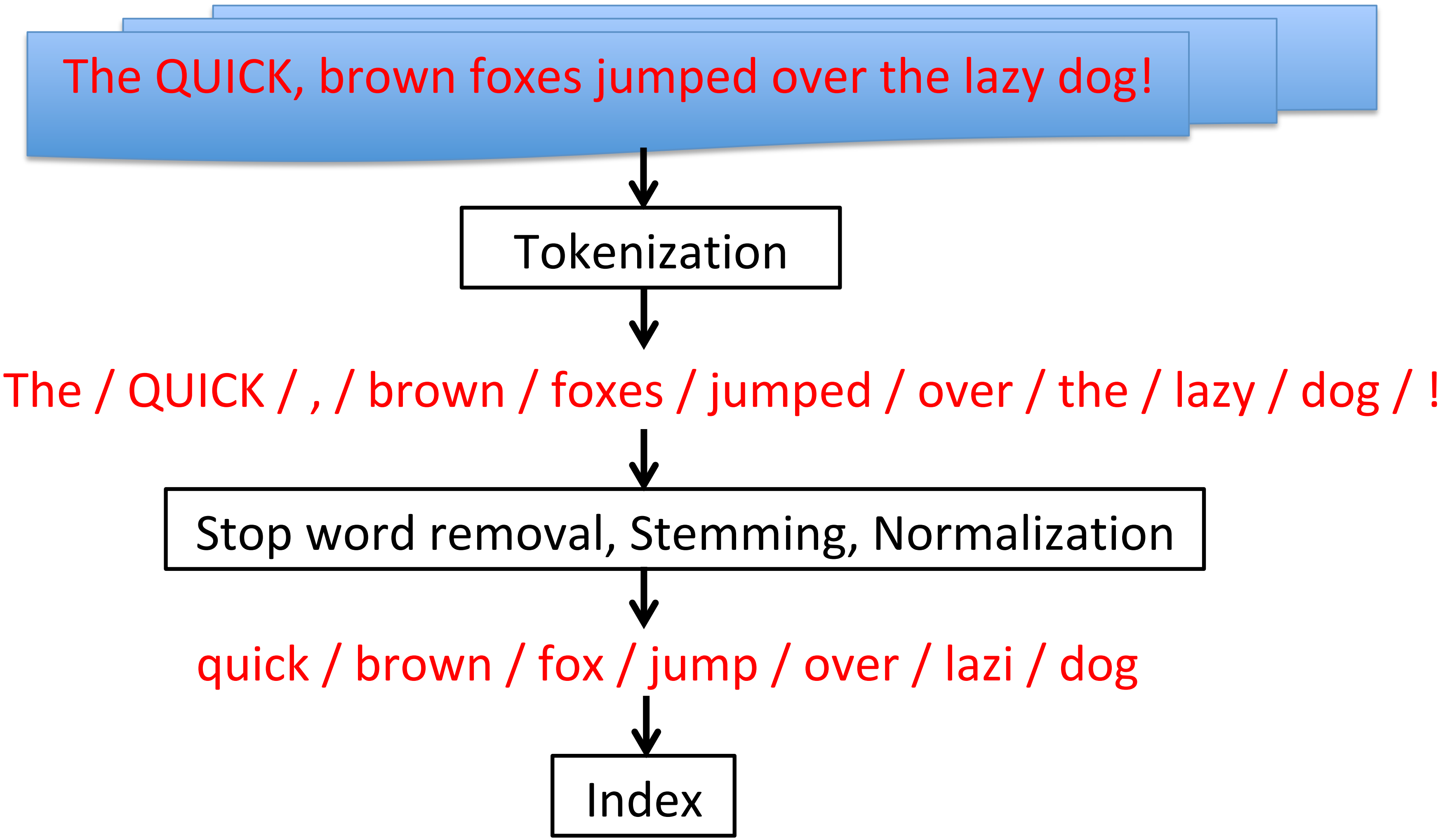
Index construction and management

- Dynamic index
  - Searching Twitter vs. static document collection
- Distributed solutions
  - MapReduce, Hadoop, etc.
  - Fault tolerance
- Pre-computing components for score function

→ Many interesting technical challenges!


User with
Information Need
Query
Next up
(1) Indexing
Documents
IR System
Representation
Function
Representation
Function
Query Representation
Document Representation
Scoring
Function
INDEX
Returned
Hits
We covered this

Representing a Document as a Bag-of-words (but what words?)



Issues in Document Representation

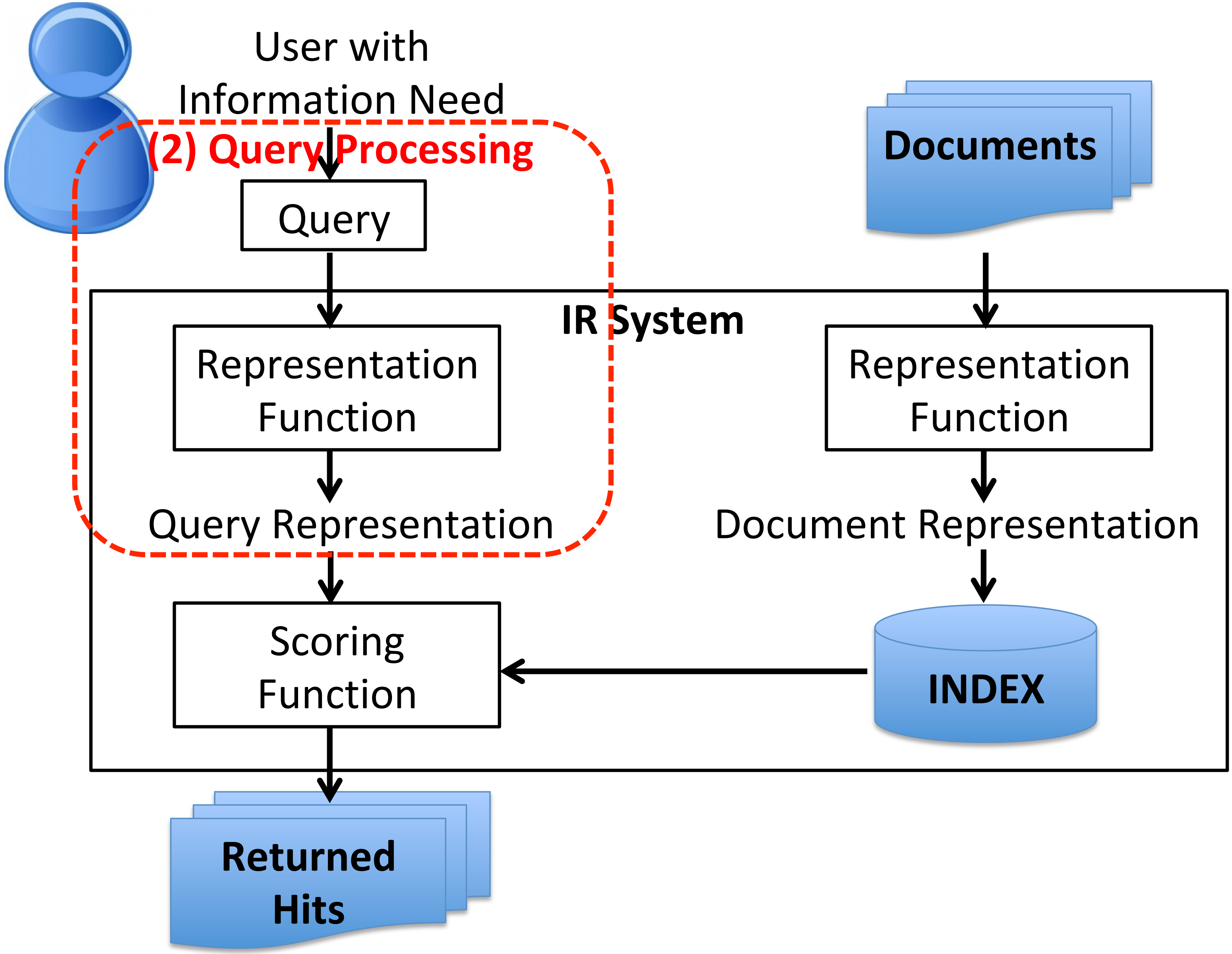
- Language-specific challenges
- Polysemy & Synonyms:
  - – “bank” in multiple senses, represented the same?
  - – “jet” and “airplane” should be same?
- Acronyms, Numbers, Document structure
- Morphology

aghnaaguq

aghnagh-	-$\sim$:(ng)u-	-$\sim_f$(g/t)u-	-q
woman-	-to.be.N-	-INTR.IND-	-3SG

‘*She is a woman*’

Central Siberian Yupik morphology example from E. Chen & L. Schartz, LREC 2018:
http://dowobeha.github.io/papers/lrec18.pdf


User with
Information Need
(2) Query Processing
Query
Documents
IR System
Representation Function
Representation Function
Query Representation
Document Representation
Scoring Function
INDEX
Returned Hits

Query Representation

- Of course, the query string must go through the same tokenization, stop word removal and normalization process like the documents

- But we can do more, esp. for free-text queries
  - – to guess user's intent & information need

Keyword search vs. Conceptual search

- Keyword search / Boolean retrieval:

  BOOLEAN QUERY:
  Brutus AND Caesar AND NOT Calpurnia

  – Answer is exact, must satisfy these terms

- Conceptual search (or just "search" like Google)

  FREE-TEXT QUERY:
  Brutus assassinate Caesar reasons

  – Answer may not need to exactly match these terms
  – *Note this naming may not be standard*

Query Expansion for "conceptual" search

- Add terms to the query representation
  - Exploit knowledge base, WordNet, user query logs

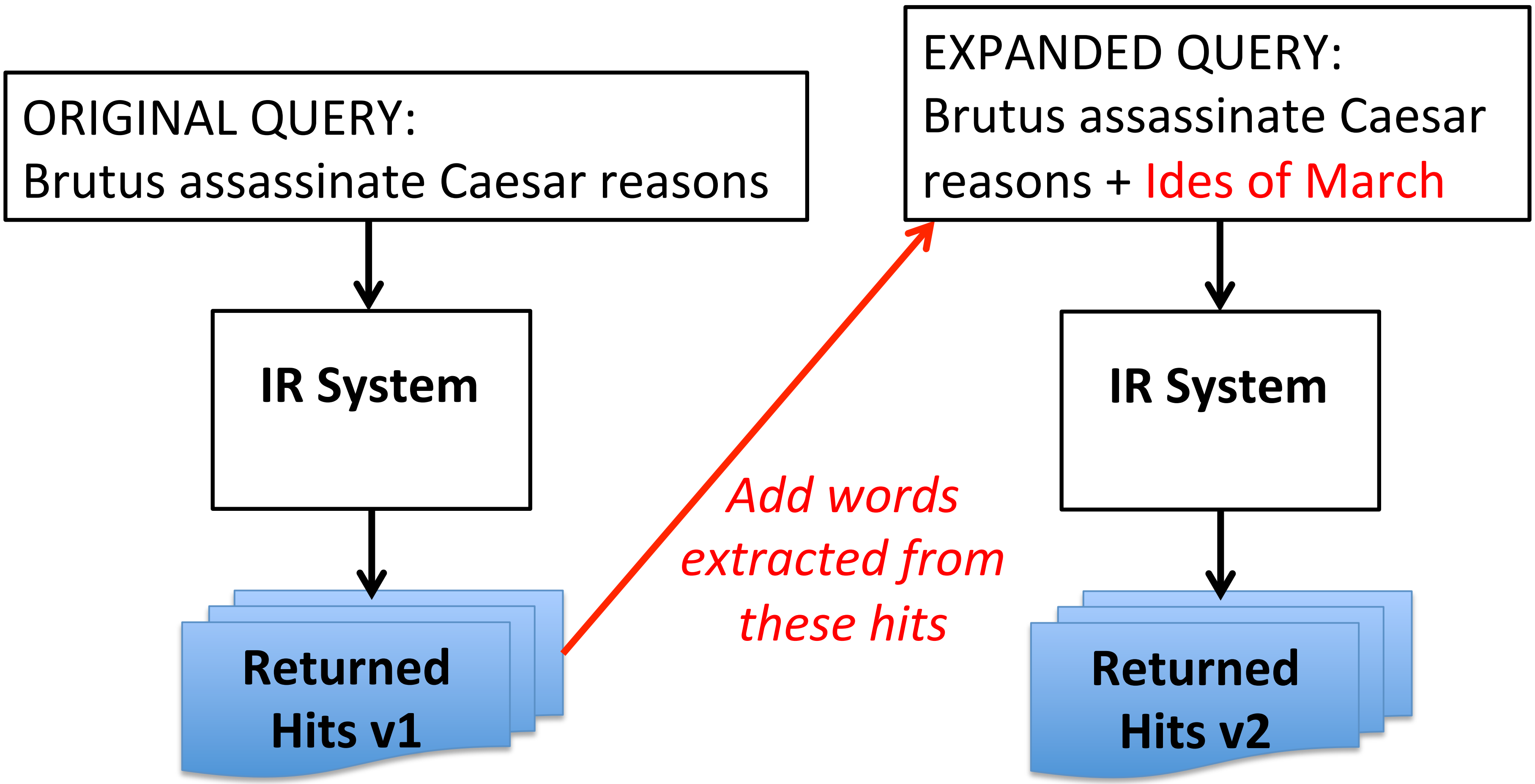
ORIGINAL FREE-TEXT QUERY:
Brutus assassinate Caesar reasons

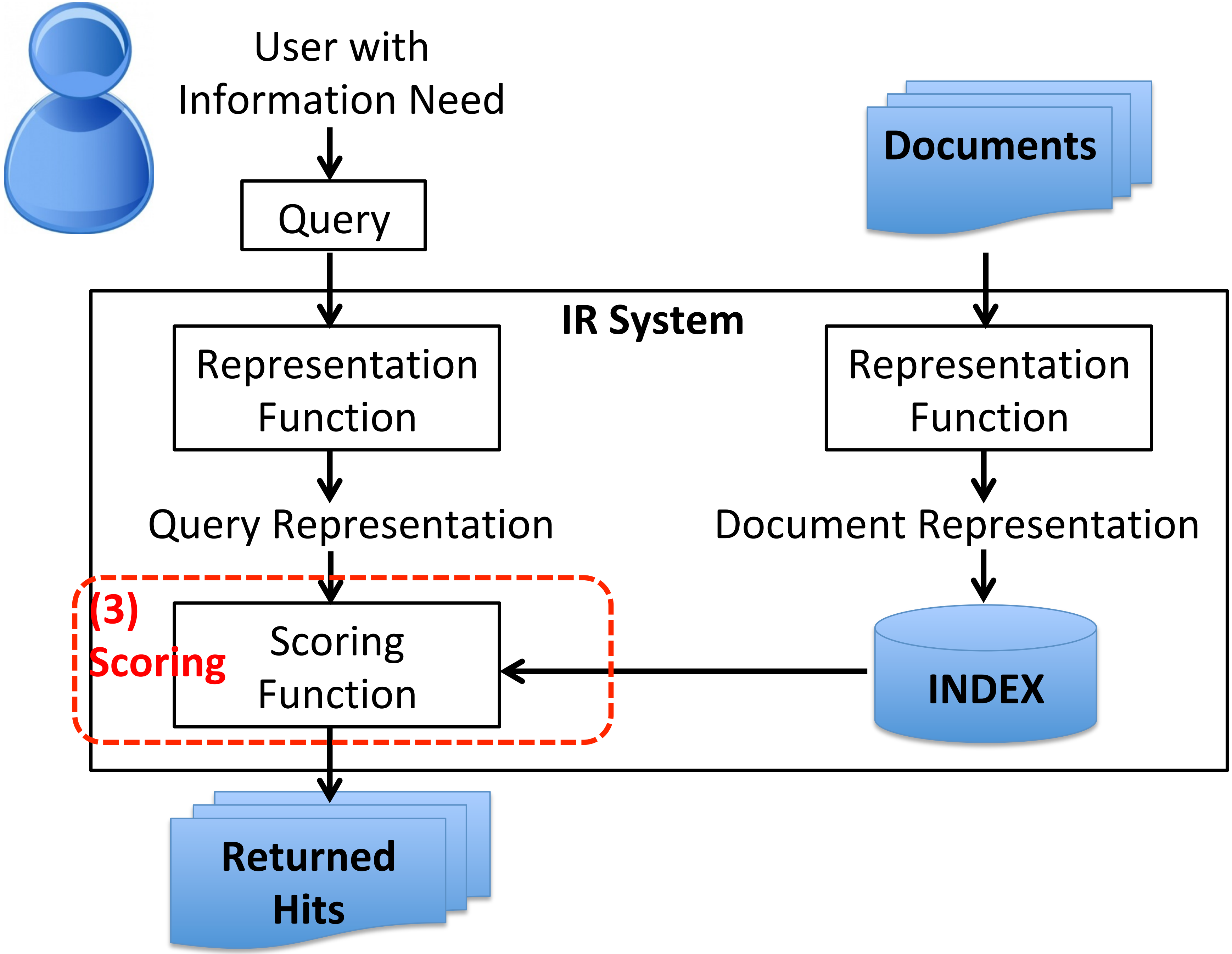
↓

EXPANDED QUERY:
Brutus assassinate kill Caesar reasons why

Pseudo-Relevance Feedback

- Query expansion by iterative search




User with
Information Need
Query
Documents
IR System
Representation
Function
Representation
Function
Query Representation
Document Representation
(3)
Scoring
Scoring
Function
INDEX
Returned
Hits

Motivation for scoring documents

- For keyword search, all documents returned should satisfy query, and are equally relevant
- For conceptual search:
  - May have too many returned documents
  - Relevance is a gradation

  → *Score* documents and return a *ranked list*

TF-IDF Scoring Function

- Given query *q* and document *d*

$$\text{TF-IDF}(q,d) = \sum_{t \in q} \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

terms t in q

Term frequency (raw count) of t in d

Inverse document frequency

$$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$$

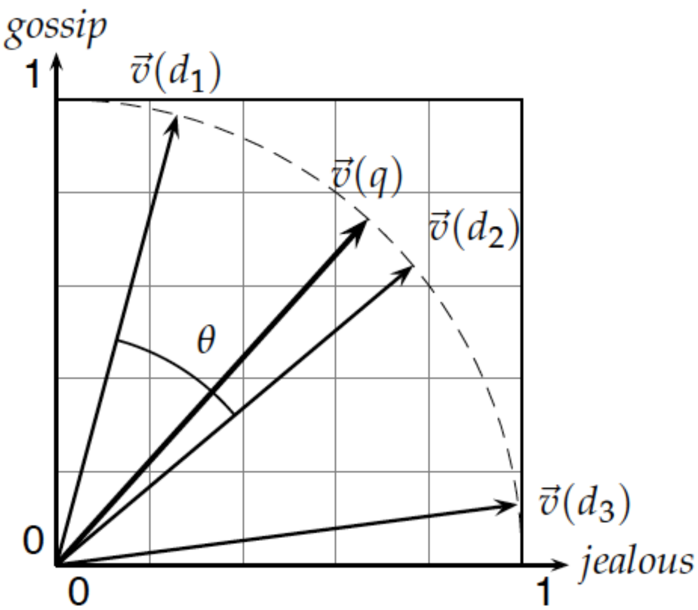
Total number of documents

Number of documents
with >=1 occurrence of t

Vector-Space Model View

- View documents (d) & queries (q) each as vectors,
  - Each vector element represents a term
  - whose value is the TF-IDF of that term in d or q
- Score function can be viewed as e.g. Cosine Similarity between vectors

$$\text{score}(q,d) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}$$



These examples/figures are from: Manning, Raghavan, Schütze, Intro to Information Retrieval, CUP, 2008

Alternative Scoring Functions: BM25

$$score(q,d) = \sum_{t \in q} \mathrm{idf_t} \times \frac{\mathrm{tf_{t,d}} \cdot (k_1 + 1)}{\mathrm{tf_{t,d}} + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

Query

Document

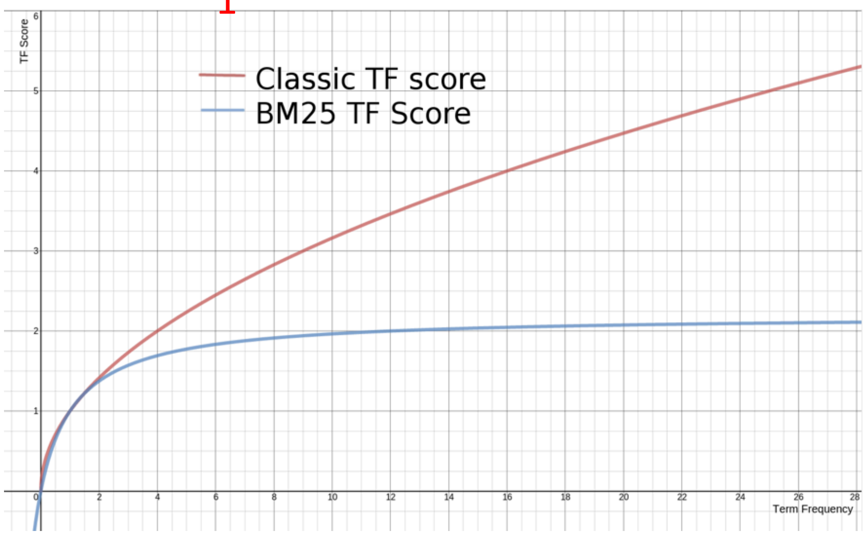
Inverse Document Frequency of query term

Frequency of query term in document

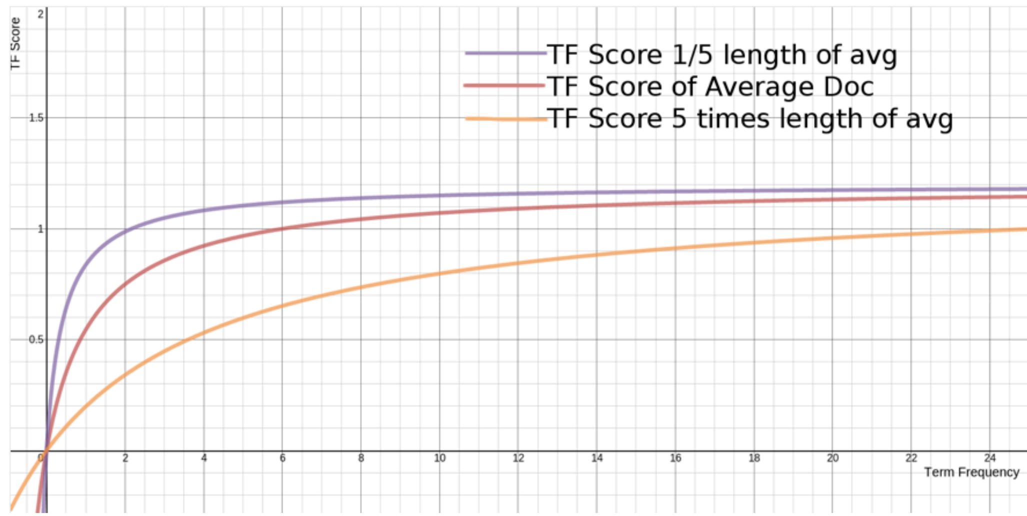
Document length ratio

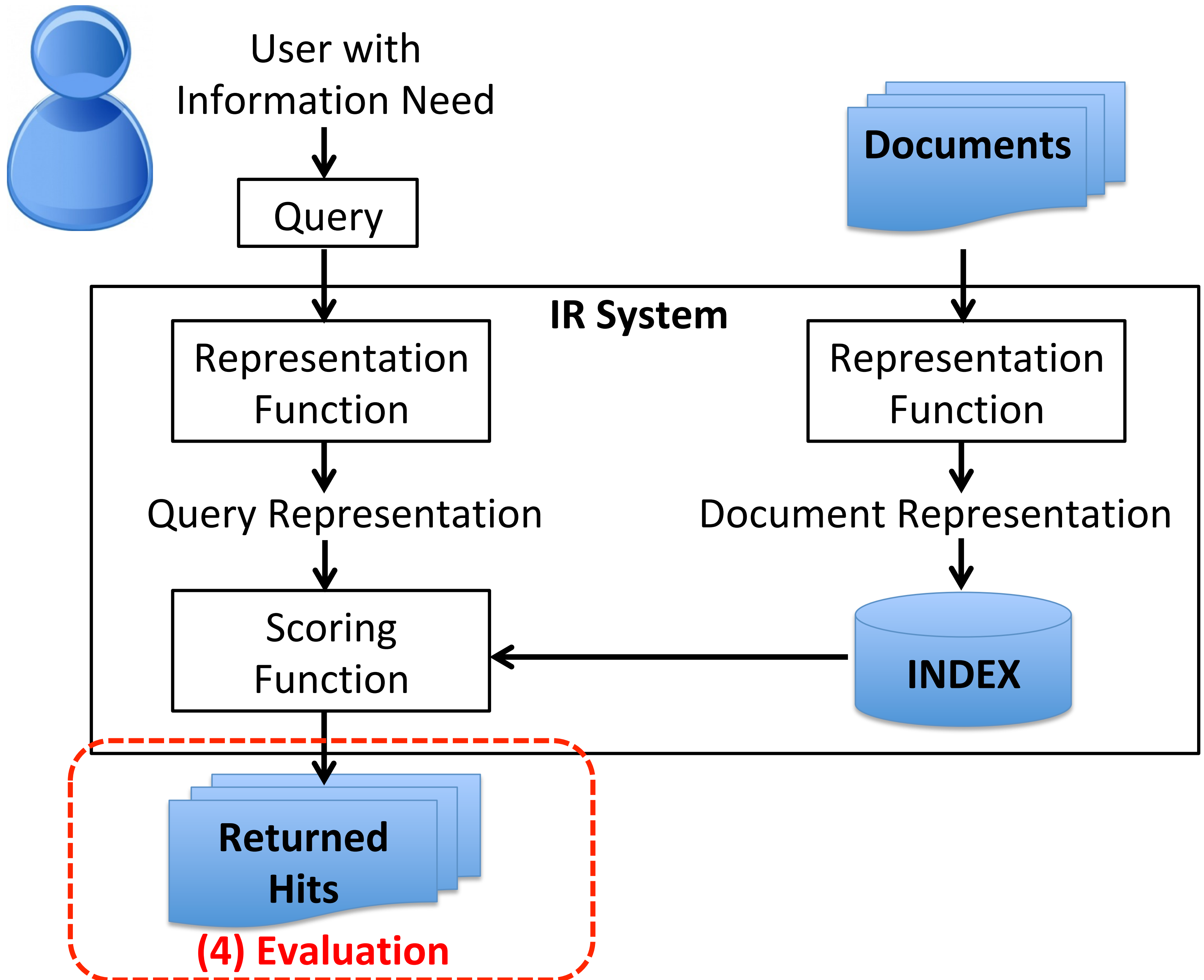
Tunable Hyperparameters

k_1: Saturation for tf



b: Document length bias




User with
Information Need
Query
Documents
IR System
Representation
Function
Representation
Function
Query Representation
Document Representation
Scoring
Function
INDEX
Returned
Hits
(4) Evaluation

Evaluation: How good/bad is my IR?

- Evaluation is important:
  - Compare two IR systems
  - Decide whether our IR is ready for deployment
  - Identify research challenges
- Two Ingredients for a trustworthy evaluation:
  - Answer Key
  - A Meaningful Metric: given query q, returned ranked list, and answer key, computes a number

Precision and Recall

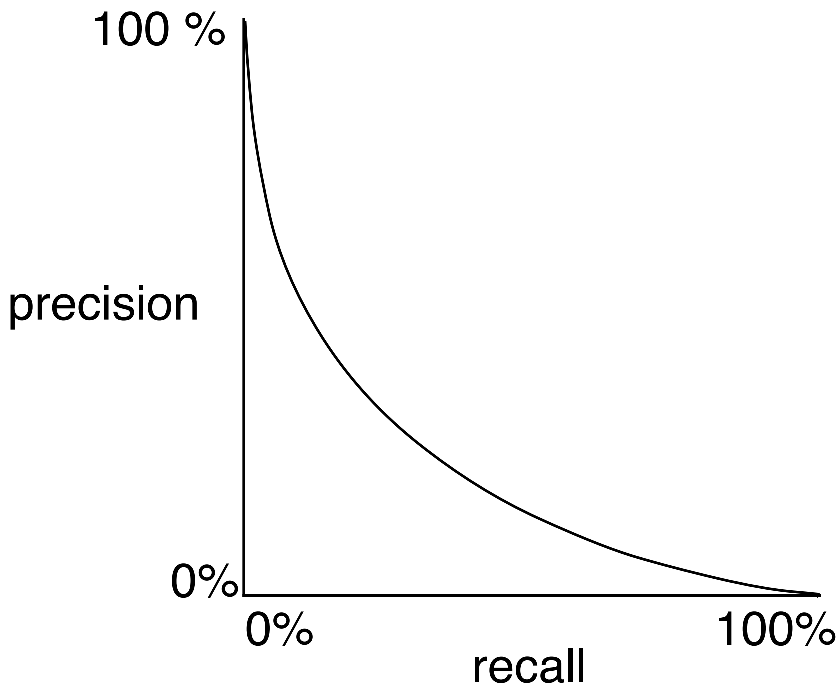
	relevant	not relevant
retrieved	A	B
not retrieved	C	D

"Type one errors" "Errors of commission" "False positives"

"Type two errors"
"Errors of omission"
"False negatives"

$$\text{precision} = \frac{A}{A + B}$$

$$\text{recall} = \frac{A}{A + C}$$



average precision = area under curve

From Paul McNamee's JSALT 2018 tutorial slides

Issues with Precision and Recall

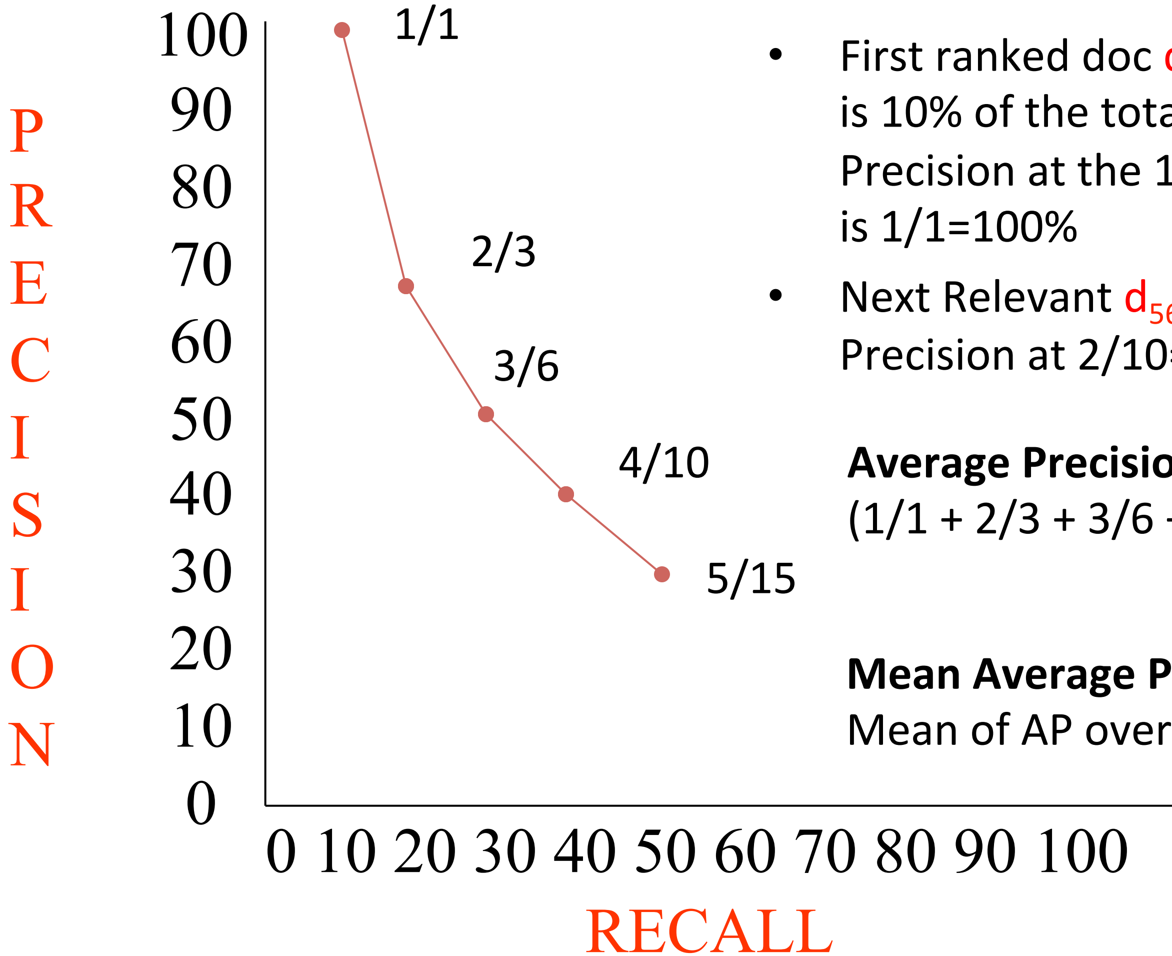
- We often don’t know true recall value
  - For large collection, impossible to have annotator read all documents to assess relevance of a query
- Focused on evaluating sets, rather than ranked lists

We’ll introduce Mean Average Precision (MAP) here. Note that IR evaluation is a deep field, worth another lecture by itself!

Example for 1 query: precision & recall at different positions in ranked list

10 relevant: $R_q=\{d_3,d_5,d_9,d_{25},d_{39},d_{44},d_{56},d_{71},d_{89},d_{123}\}$

Ranked List: d_{123}, d_{84} , d_{56}, d_6, d_8, d_9, d_{511}, d_{129}, d_{187}, d_{25}, d_{38m}, d_{48}, d_{250}, d_{113}, d_3

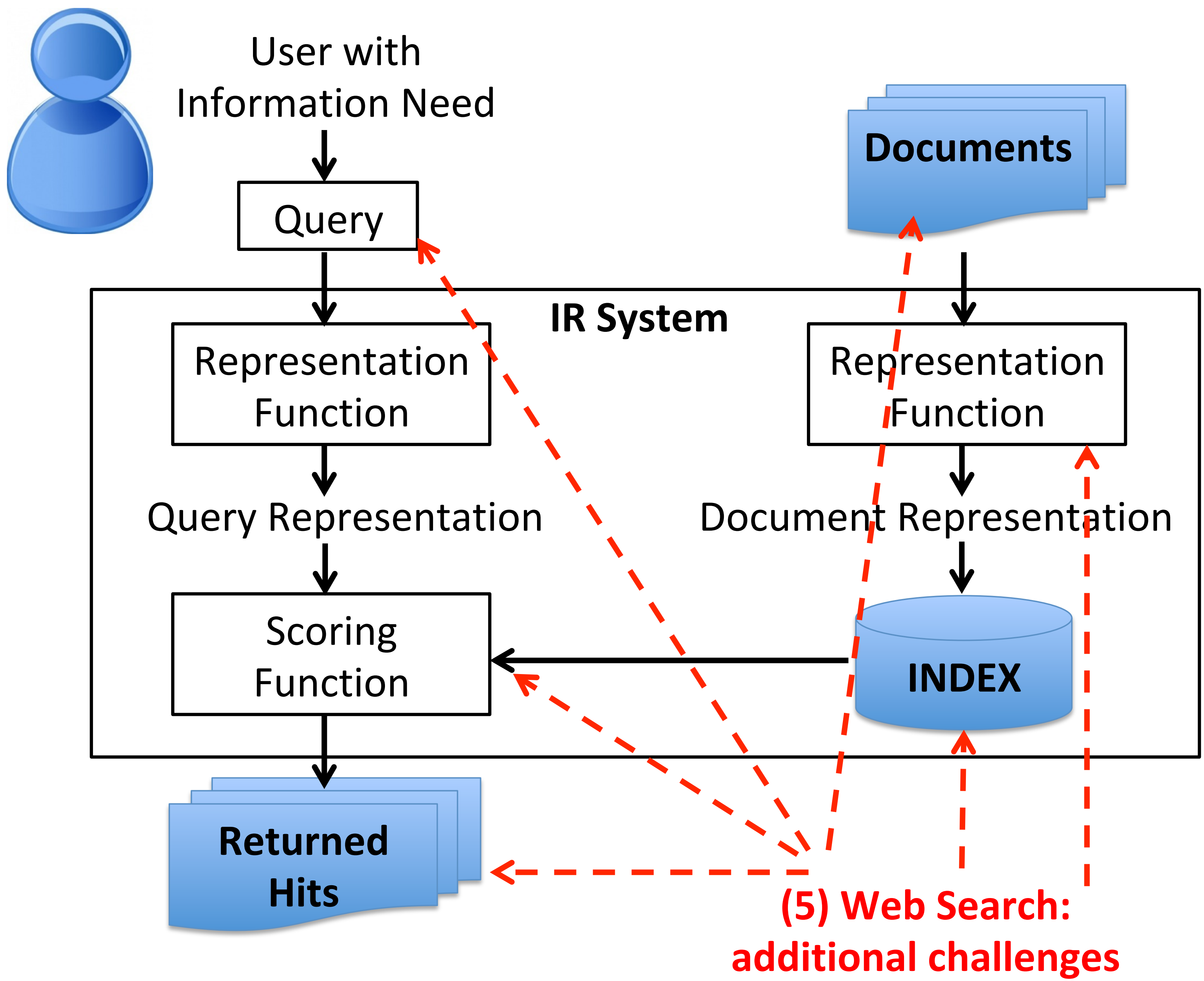


- First ranked doc d_{123} is relevant, which is 10% of the total relevant. Therefore Precision at the 1/10=10% Recall level is 1/1=100%
- Next Relevant d_{56} gives us 2/3=66% Precision at 2/10=20% recall level

Average Precision (AP):
(1/1 + 2/3 + 3/6 + 4/10 + 5/15) / 5 = 0.58

Mean Average Precision (MAP):
Mean of AP over multiple queries

From Paul McNamee's JSALT 2018 tutorial slides


User with
Information Need
Query
Documents
IR System
Representation
Function
Representation
Function
Query Representation
Document Representation
Scoring
Function
INDEX
Returned
Hits
(5) Web Search:
additional challenges

A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.

-- Vannevar Bush (1945)

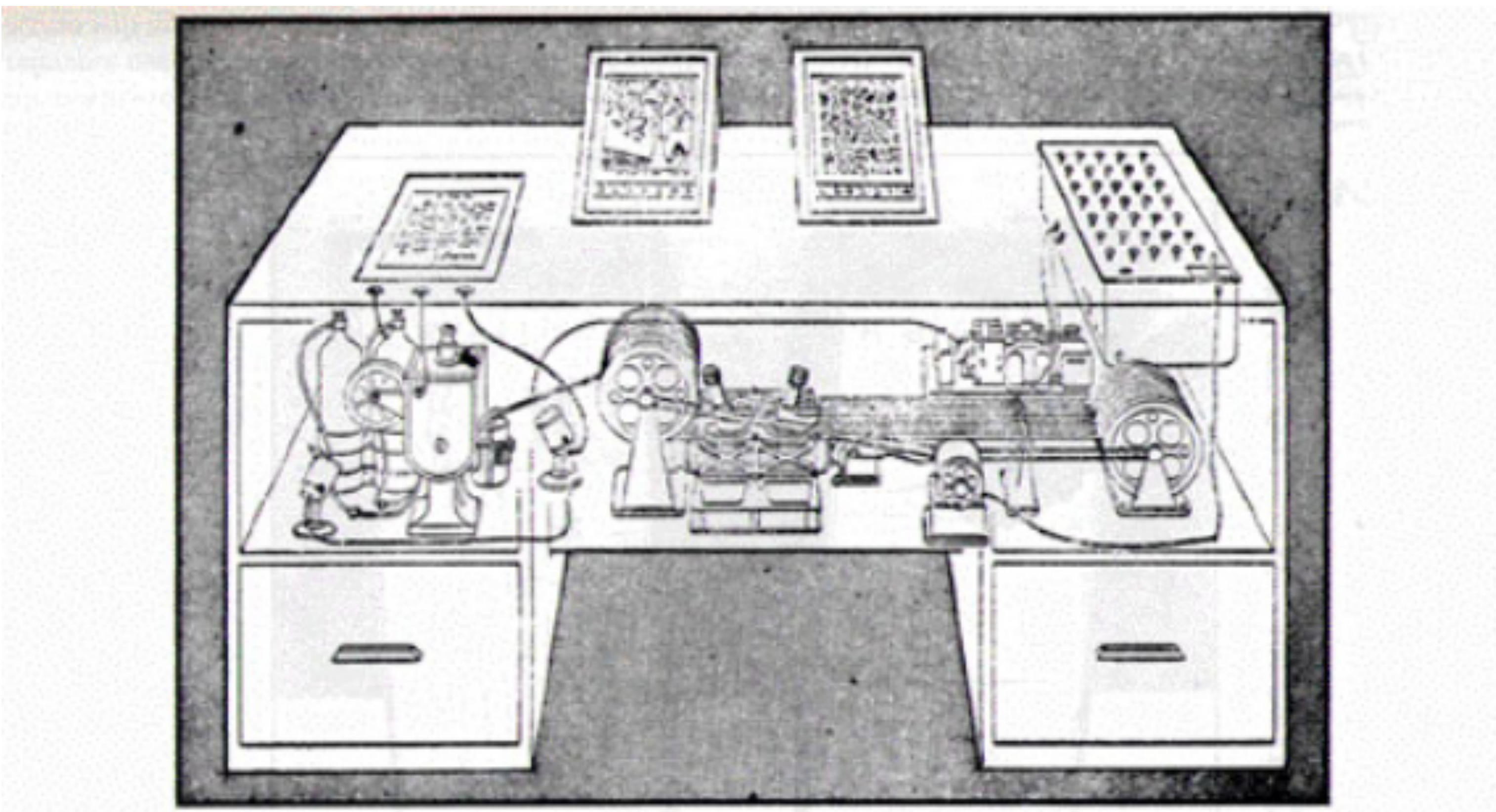


Image Source: Original illustration of the Memex from the Life reprint of "As We May Think"
https://history-computer.com/Internet/Dreamers/Bush.html


Google!
.com

Some history

- 1945: Vannevar Bush writes about MEMEX
- 1975: Microsoft founded
- 1981: IBM PC
- 1989: Tim Berners-Lee invents WWW
- 1992: 1M internet hosts, but only 50 web sites
- 1994: Yahoo founded, builds online directory
- 1995: AltaVista indexes 15M web pages
- 1998: Google founded
- 2004: Google IPO

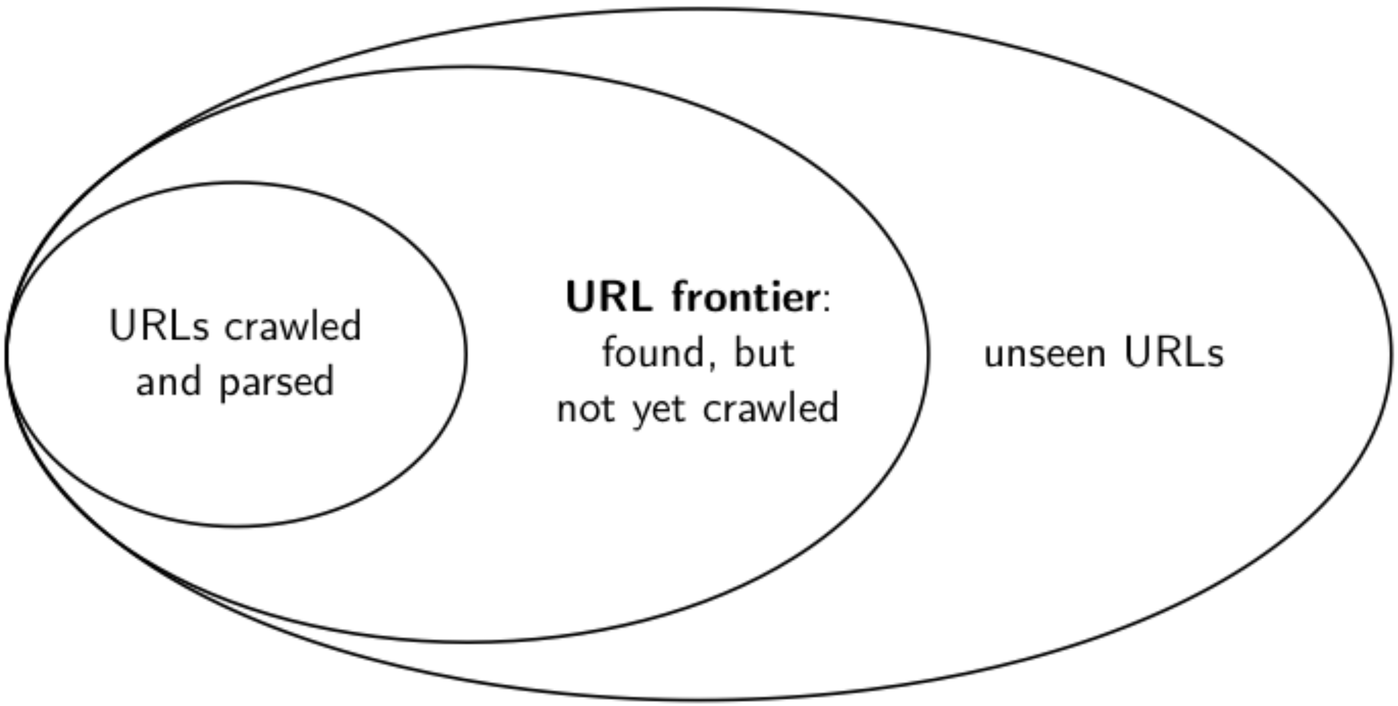
From Paul McNamee’s JSALT 2018 tutorial slides

Web Search: a sample of challenges & opportunities

- Crawling
  - Infrastructure to handle scale
  - Where to crawl, how often: Freshness, Deep Web
- Web document characteristics:
  - Hypertext structure, HTML tags
  - Diverse types of information
  - Dealing with Search Engine Optimization (SEO)
- Large User base
  - Long-tail of queries
  - Exploiting query logs and click logs
  - User interface research (including voice search)
- Advertising ecosystem, etc.

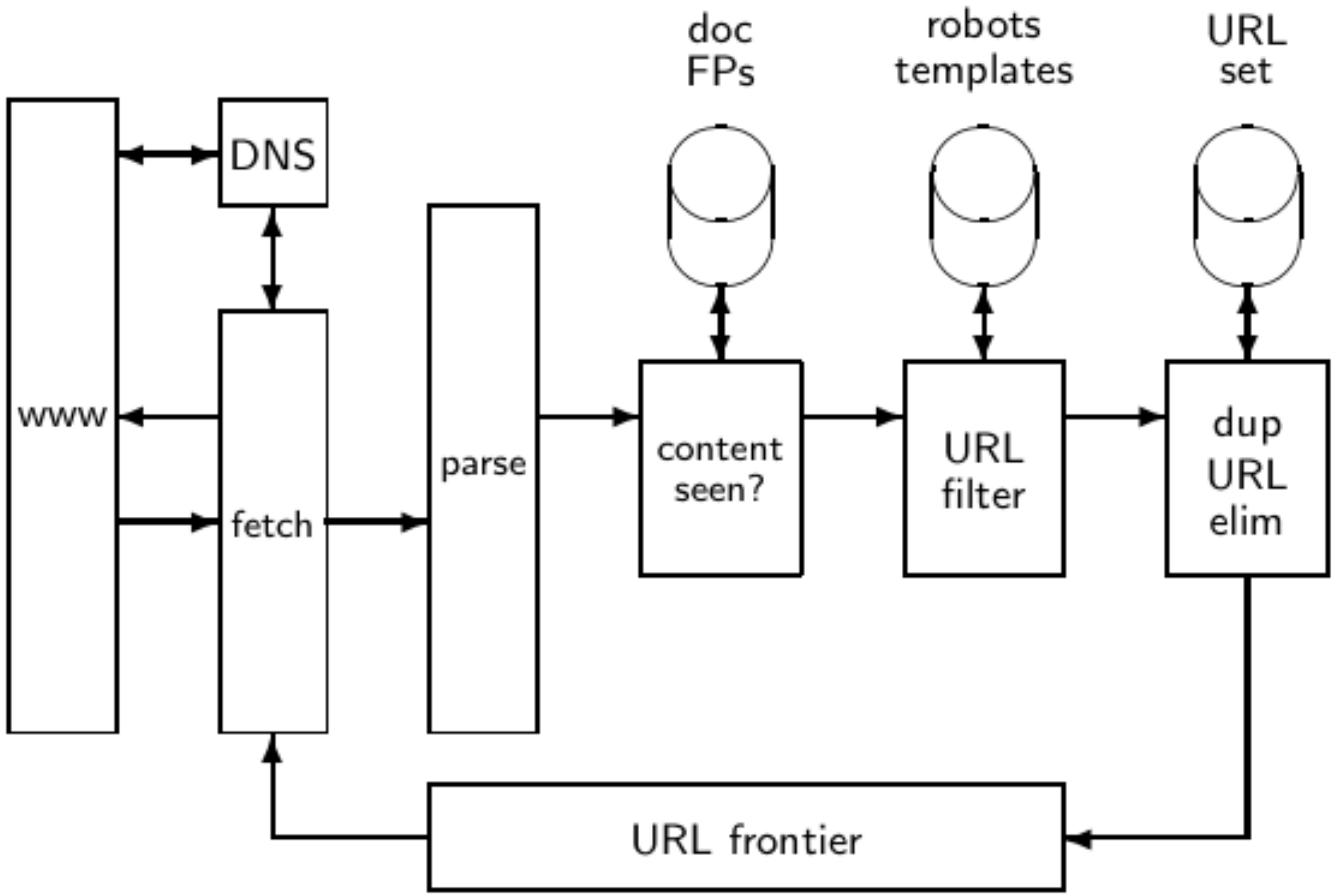
Crawling: Basic algorithm

- Start with a set of known pages in the queue
- Repeat: (1) pop queue, (2) download & parse page, (3) push discovered URL on queue



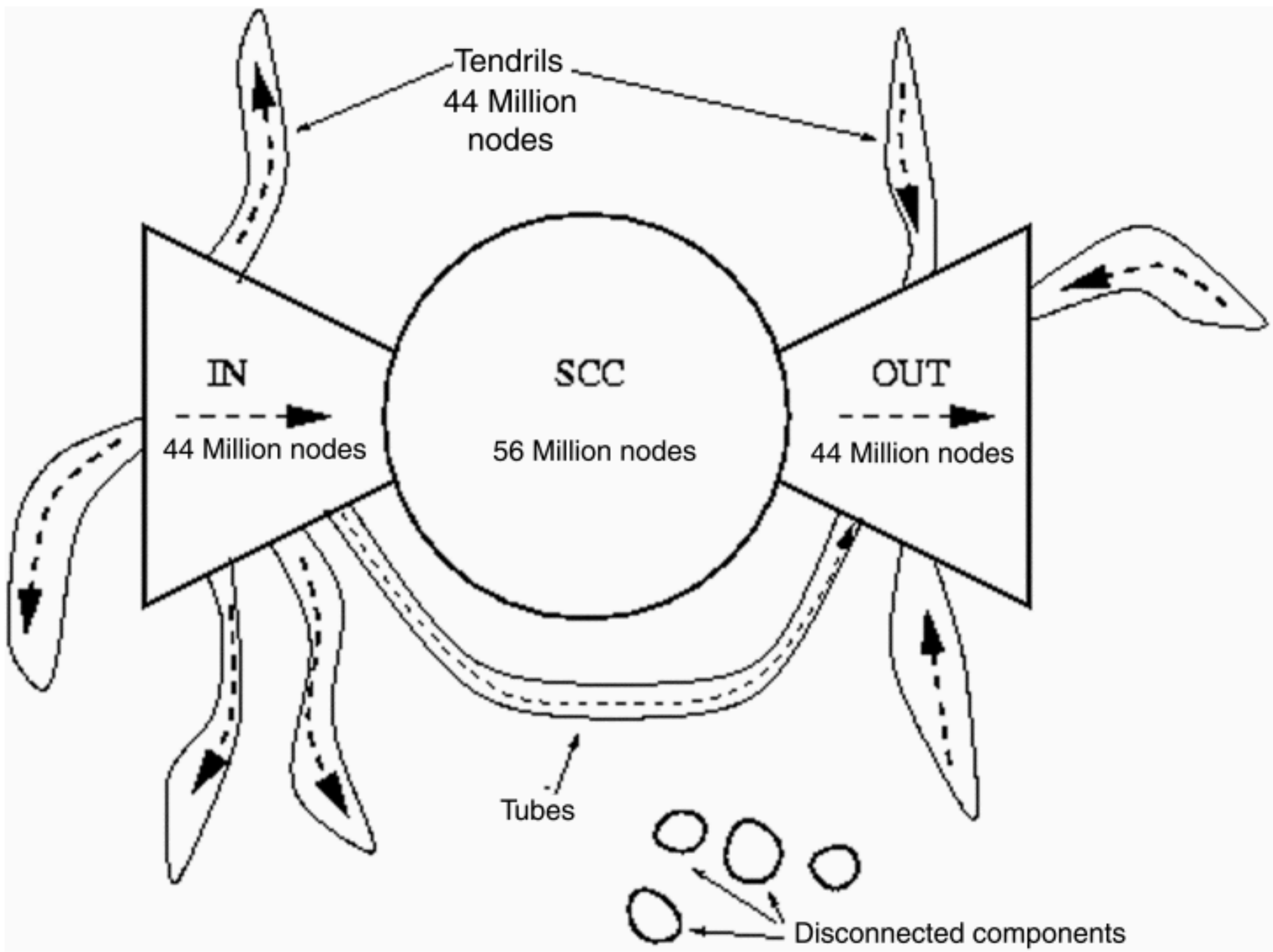
From Doug Oard's slides: http://users.umiacs.umd.edu/~oard/teaching/734/spring18/

Crawling: Basic algorithm



Bowtie link structure of the Web, circa 2000

A. Broder et al. / Computer Networks 33 (2000) 309–320



Exploiting link structure: PageRank

- Pages with more in-links have more authority

- "Prior" document score

- Can be viewed as probability of a random surfer landing on a page

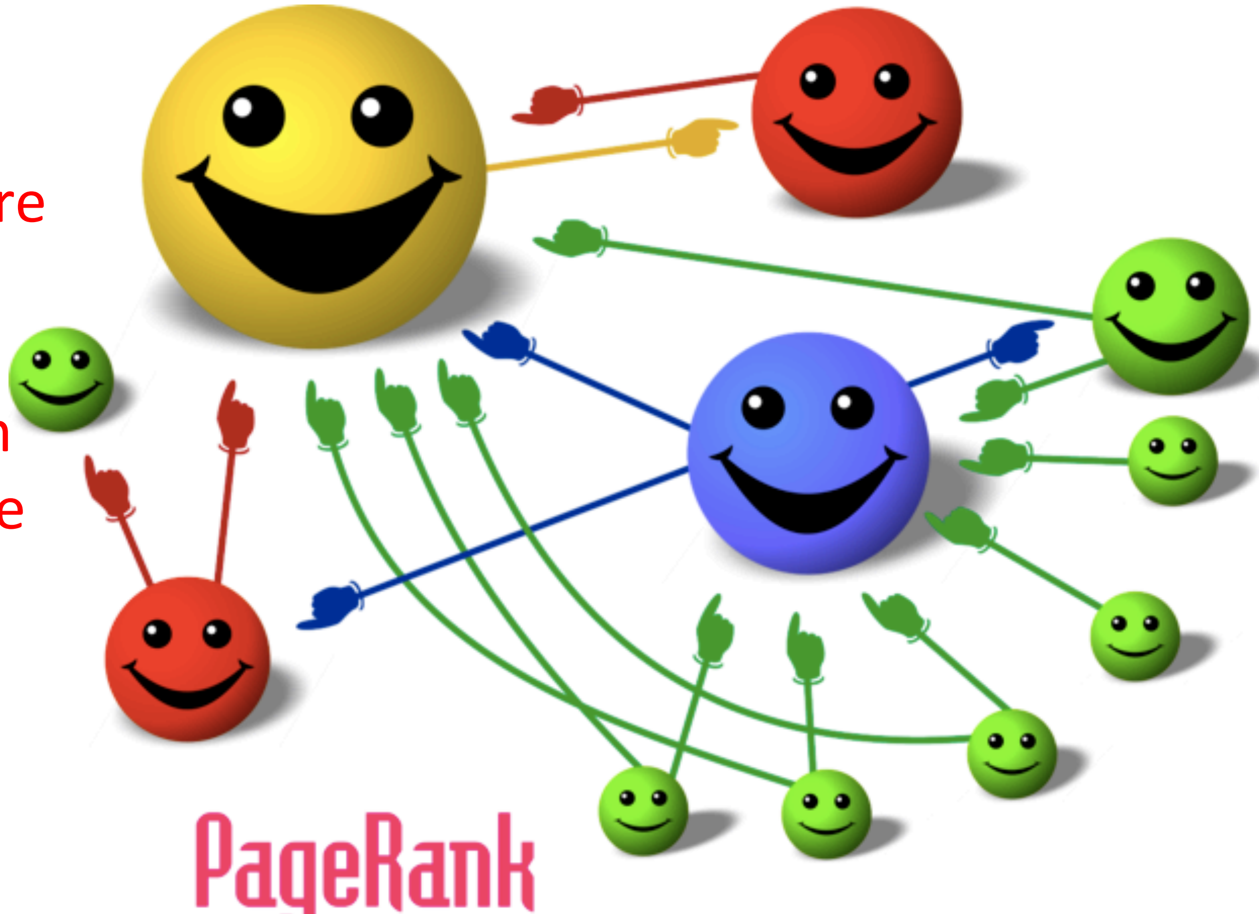
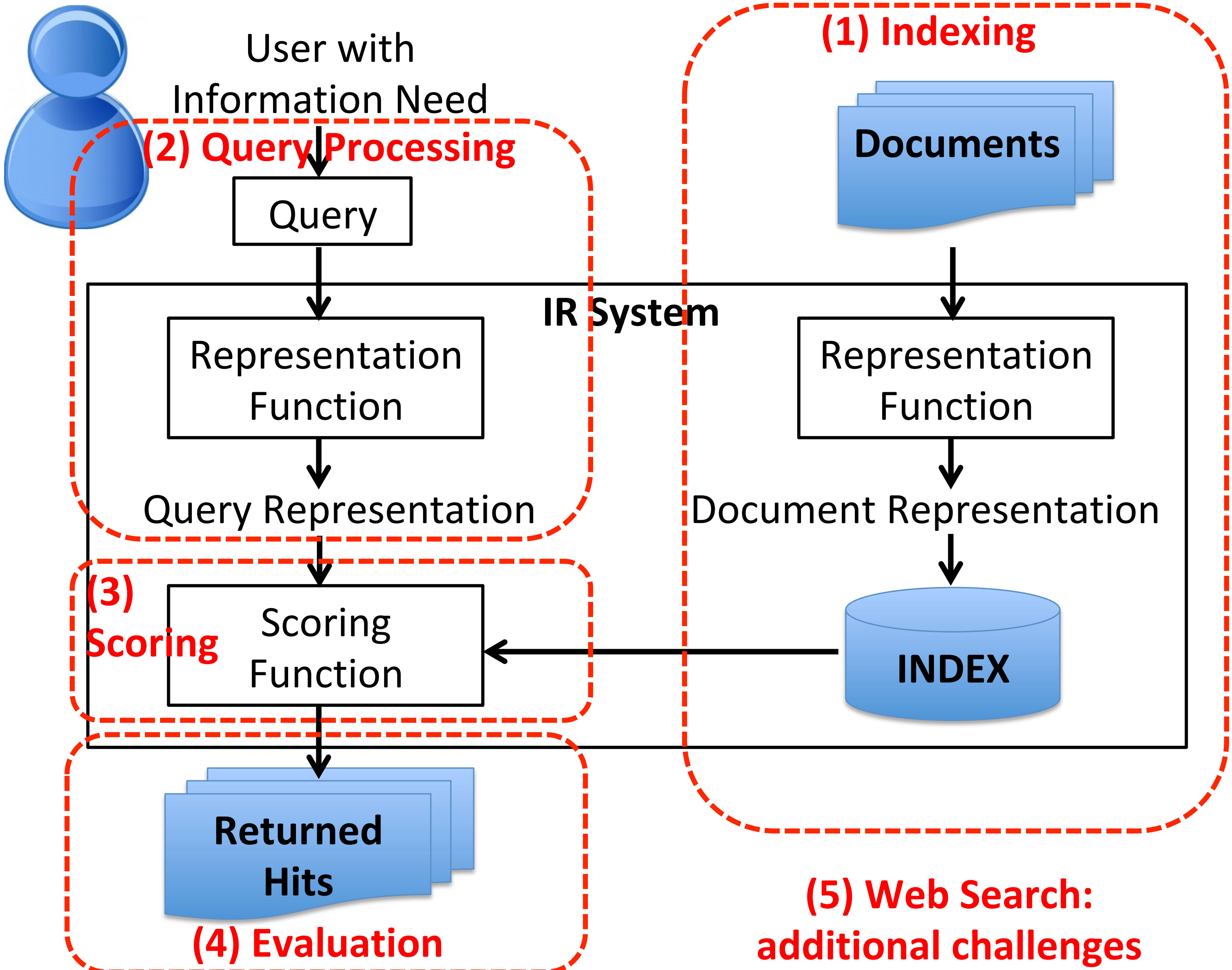


Image source: Illustration of PageRank by Felipe Micaroni Lalli
https://en.wikipedia.org/wiki/File:PageRank-hi-res.png

Diversity of user queries

- “20-25% of the queries we will see today, we have never seen before”
    - Udi Manber (Google VP, May 2007)
- A. Broder in *A taxonomy of Web search (2002)* classifies user queries as:
    - Informational
    - Navigational
    - Transactional

To Sum Up



User with Information Need
(2) Query Processing
Query
IR System
Representation Function
Query Representation
(3) Scoring
Scoring Function
Returned Hits
(4) Evaluation
(1) Indexing
Documents
Representation Function
Document Representation
INDEX
(5) Web Search: additional challenges