Information Retrieval: An Introduction

Dr. Grace Hui Yang

InfoSense

Department of Computer Science

Georgetown University, USA

huiyang@cs.georgetown.edu

Jan 2019 @ Cape Town

A Quick Introduction

- What do we do at InfoSense
  - Dynamic Search
    - IR and AI
  - Privacy and IR
- Today's lecture is on IR fundamentals
  - Textbooks and some of their slides are referenced and used here
    - Modern Information Retrieval: The Concepts and Technology behind Search. by Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Second condition. 2011.
    - Introduction to Information Retrieval. C.D. Manning, P. Raghavan, H. Schütze. Cambridge UP, 2008.
    - Foundations of Statistical Natural Language Processing. Christopher D. Manning and Hinrich Schütze.
    - Search Engines: Information Retrieval in Practice. W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009.
  - Personal views are also presented here
    - Especially in the Introduction and Summary sections

Outline

- What is Information Retrieval
  - Task, Scope, Relations to other disciplines
- Process
  - Preprocessing, Indexing, Retrieval, Evaluation, Feedback
- Retrieval Approaches
  - Boolean
  - Vector Space Model
  - BM25
  - Language Modeling
- Summary
  - What works
  - State-of-the-art retrieval effectiveness
  - Relation to the learning-based approaches

What is Information Retrieval (IR)?

- Task: To find a few among many
- It is probably motivated by the situation of information overload and acts as a remedy to it
- When defining IR, we need to be aware that there is a broad sense and a narrow sense

Broad Sense of IR

- It is a discipline that finds information that people want
- The motivation behind would include
  - Humans' desire to understand the world and to gain knowledge
  - Acquire sufficient and accurate information/answer to accomplish a task
- Because finding information can be done in so many different ways, IR would involve:
  - Classification (Wednesday lecture by Fraizio Sabastiani and Alejandro Mereo))
  - Clustering
  - Recommendation
  - Social network
  - Interpreting natural languages (Wednesday lecture by Fraizio Sabastiani and Alejandro Mereo))
  - Question answering
  - Knowledge bases
  - Human-computer interaction (Friday lecture by Rishabh Mehrotra)
  - Psychology, Cognitive Science, (Thursday lecture by Joshua Kroll), …
  - Any topic that listed on IR conferences such as SIGIR/ICTIR/CHIIR/CIKM/WWW/WSDM…

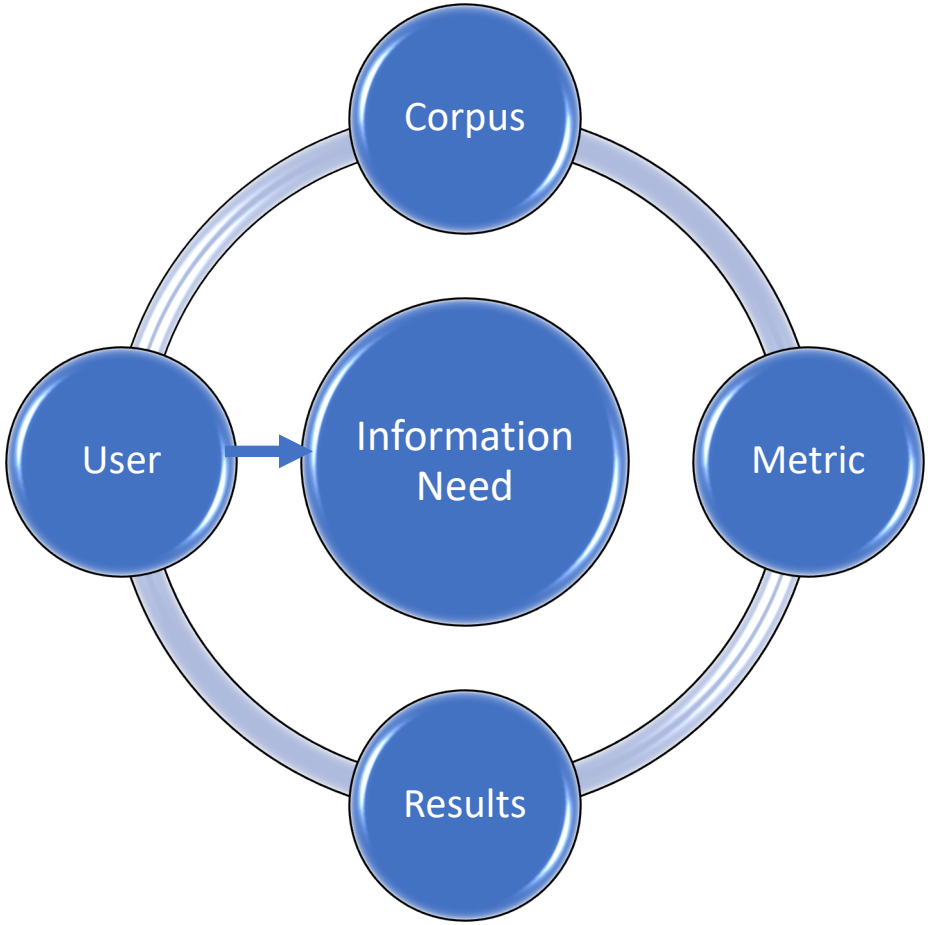
Narrow Sense of IR

- It is ‘search’
  - Mostly searching for documents
- It is a computer science discipline that designs and implements algorithms and tools to help people find information that they want
  - from one or multiple large collections of materials (text or multimedia, structured or unstructured, with or without hyperlinks, with or without metadata, in a foreign language or not – Monday Lecture Multilingual IR by Doug Oard),
  - where people can be a single user or a group
  - who initiate the search process by an information need,
  - and, the resulting information should be relevant to the information need (based on the judgement by the person who starts the search)

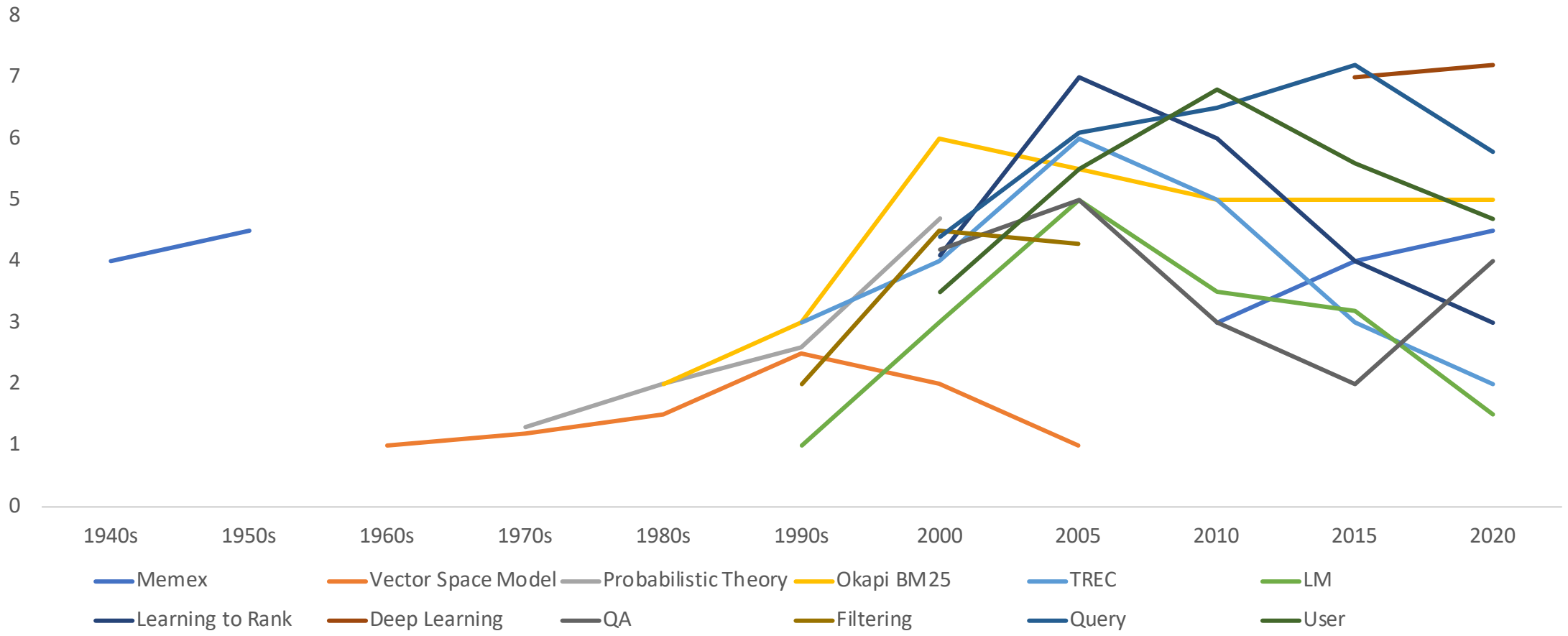
Narrowest Sense of IR

- It helps people find relevant documents
  - from one large collection of material (which is the Web or a TREC collection),
  - where there is a single user,
  - who initiates the search process by a query driven by an information need,
  - and, the resulting documents should be ranked (from the most relevant to the least) and returned in a list

Players in Information Retrieval

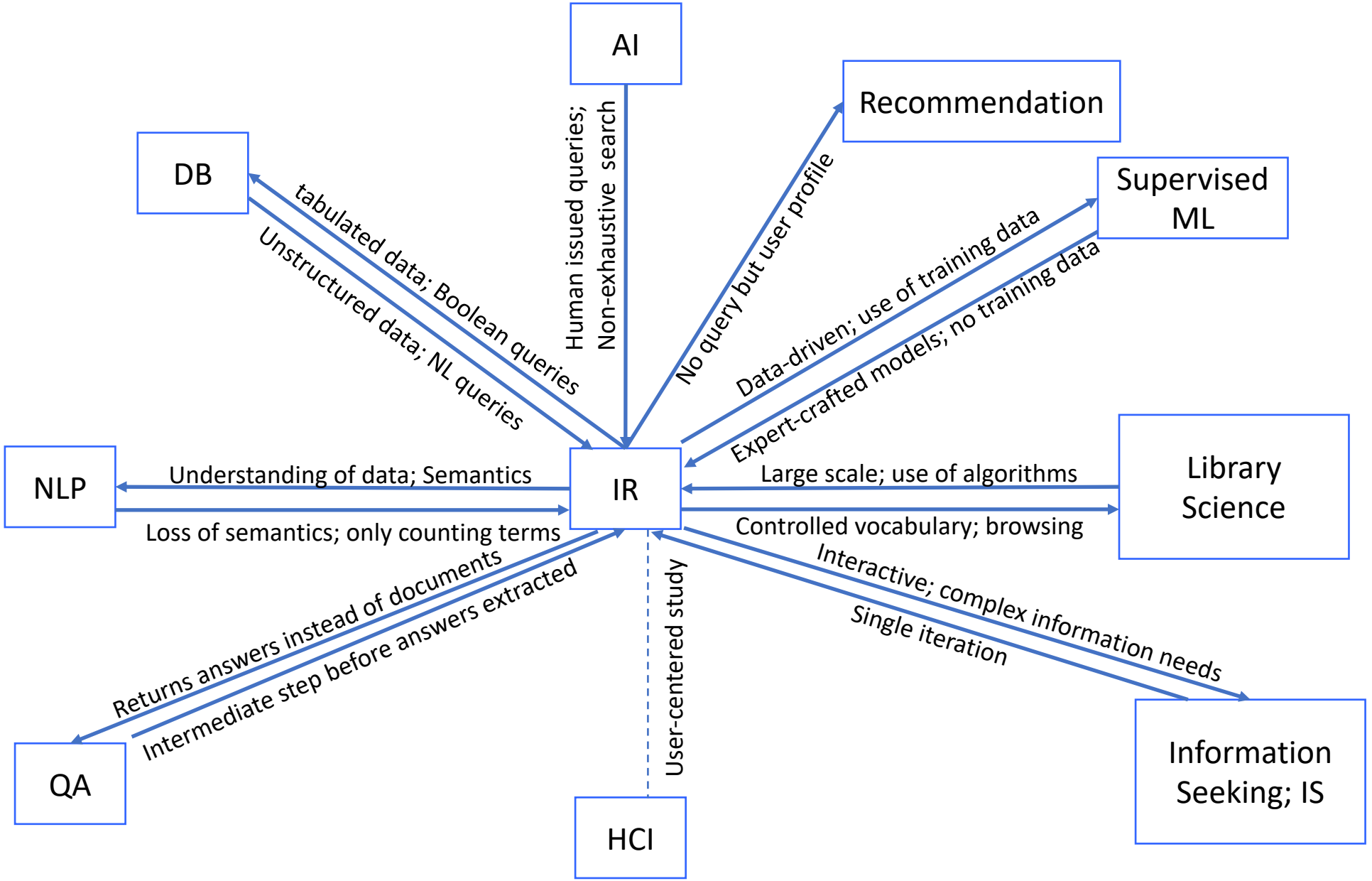


A Brief Historical Line of Information Retrieval



Relationships to Sister Disciplines

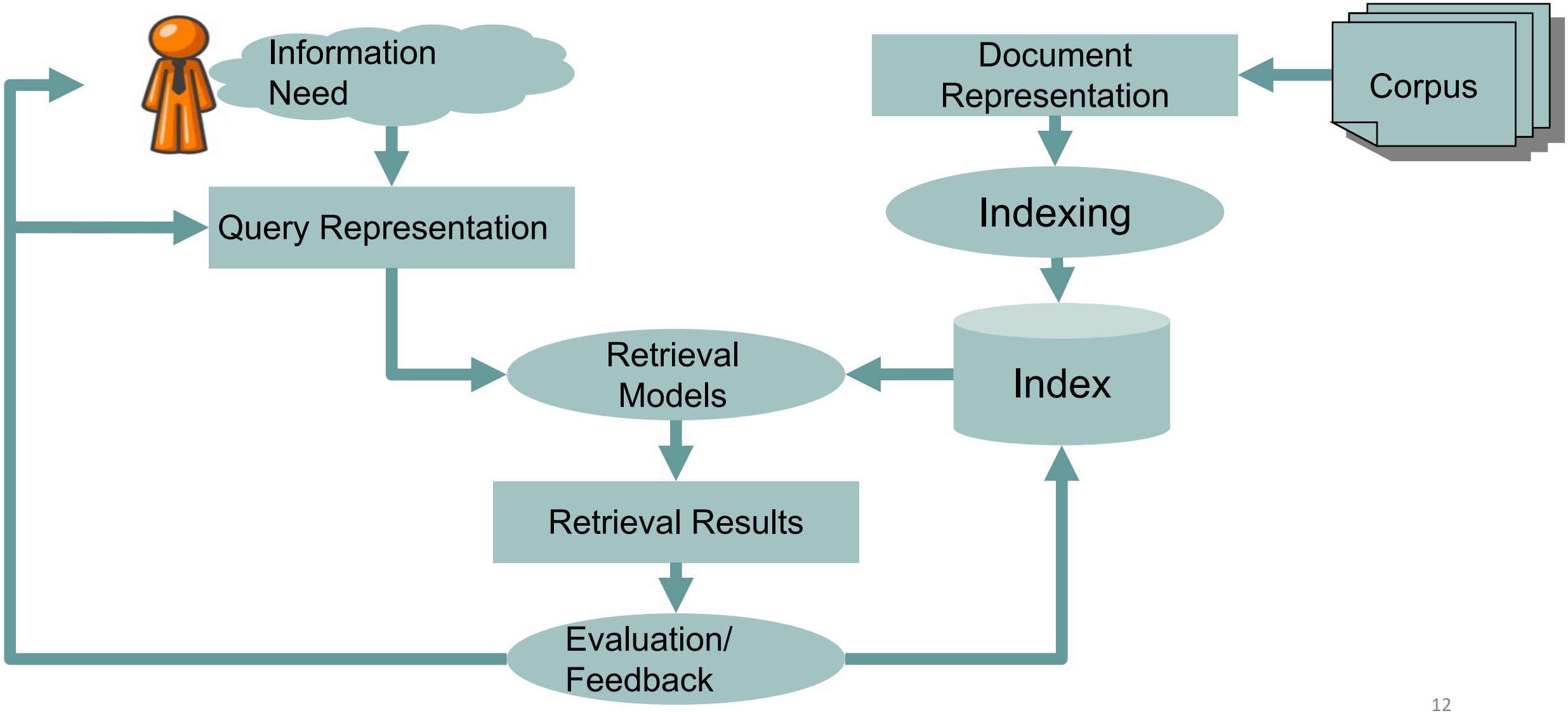
Solid line: transformations or special cases
Dashed line: overlap with



Outline

- What is Information Retrieval
  - Task, Scope, Relations to other disciplines
- Process
  - Preprocessing, Indexing, Retrieval, Evaluation, Feedback
- Retrieval Approaches
  - Boolean
  - Vector Space Model
  - BM25
  - Language Modeling
- Summary
  - What works
  - State-of-the-art retrieval effectiveness
  - Relations to the learning-based approaches

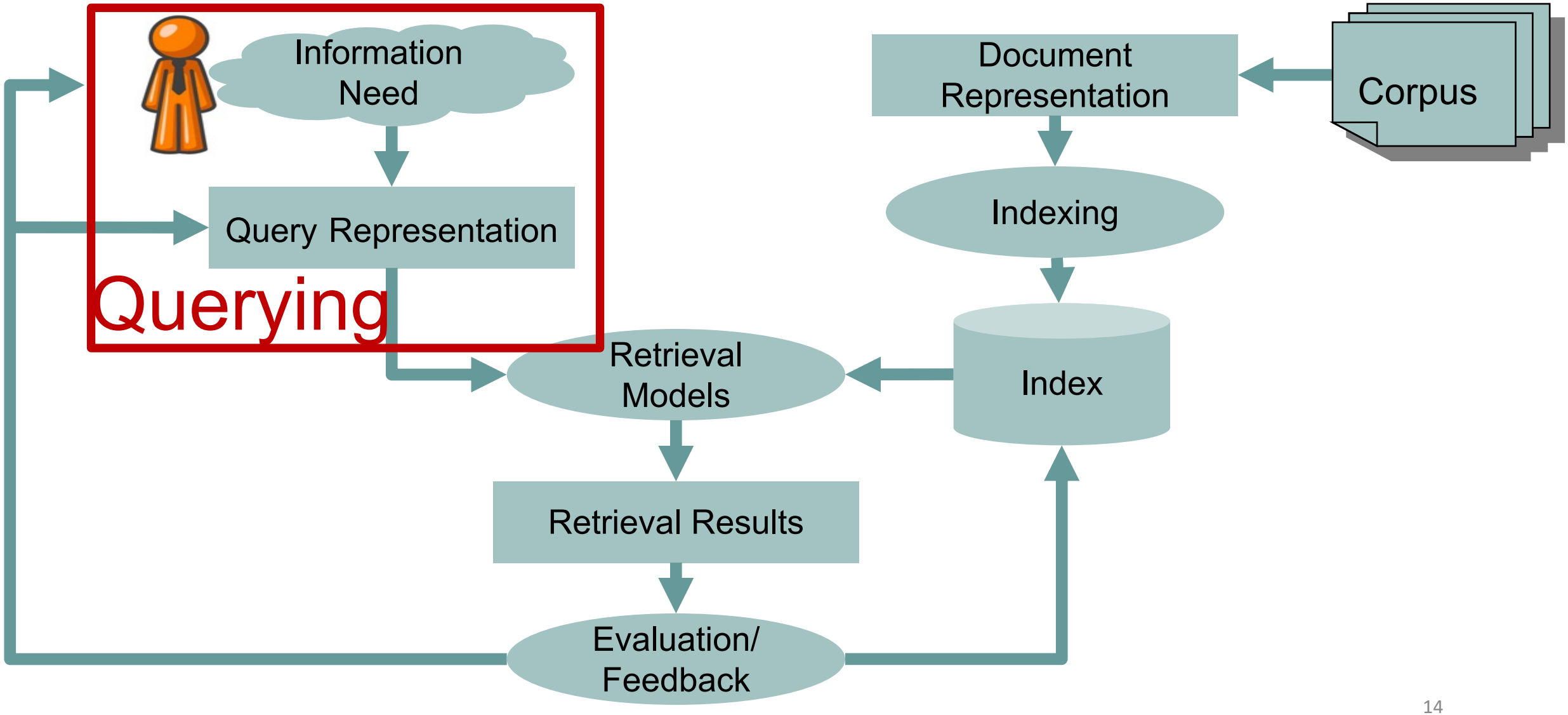
Process of Information Retrieval



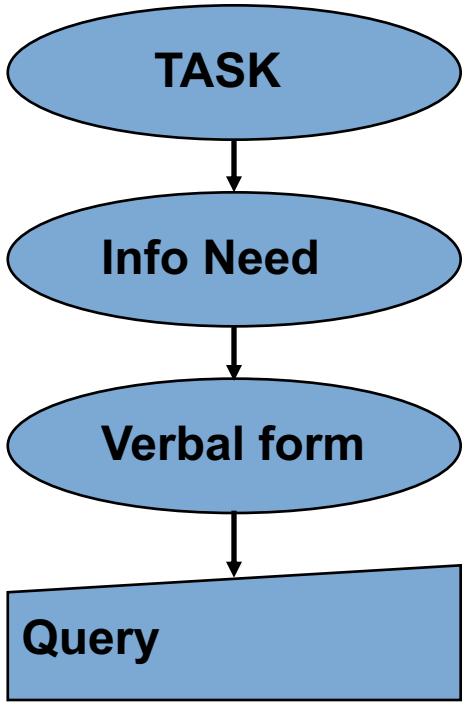
Terminology

- Query: text to represent an information need
- Document: a returned item in the index
- Term/token: a word, a phrase, an index unit
- Vocabulary: set of the unique tokens
- Corpus/Text collection
- Index/database: index built for a corpus
- Relevance feedback: judgment from human
- Evaluation Metrics: how good is a search system?
  - Precision, Recall, F1

Document Retrieval Process



From Information Need to Query



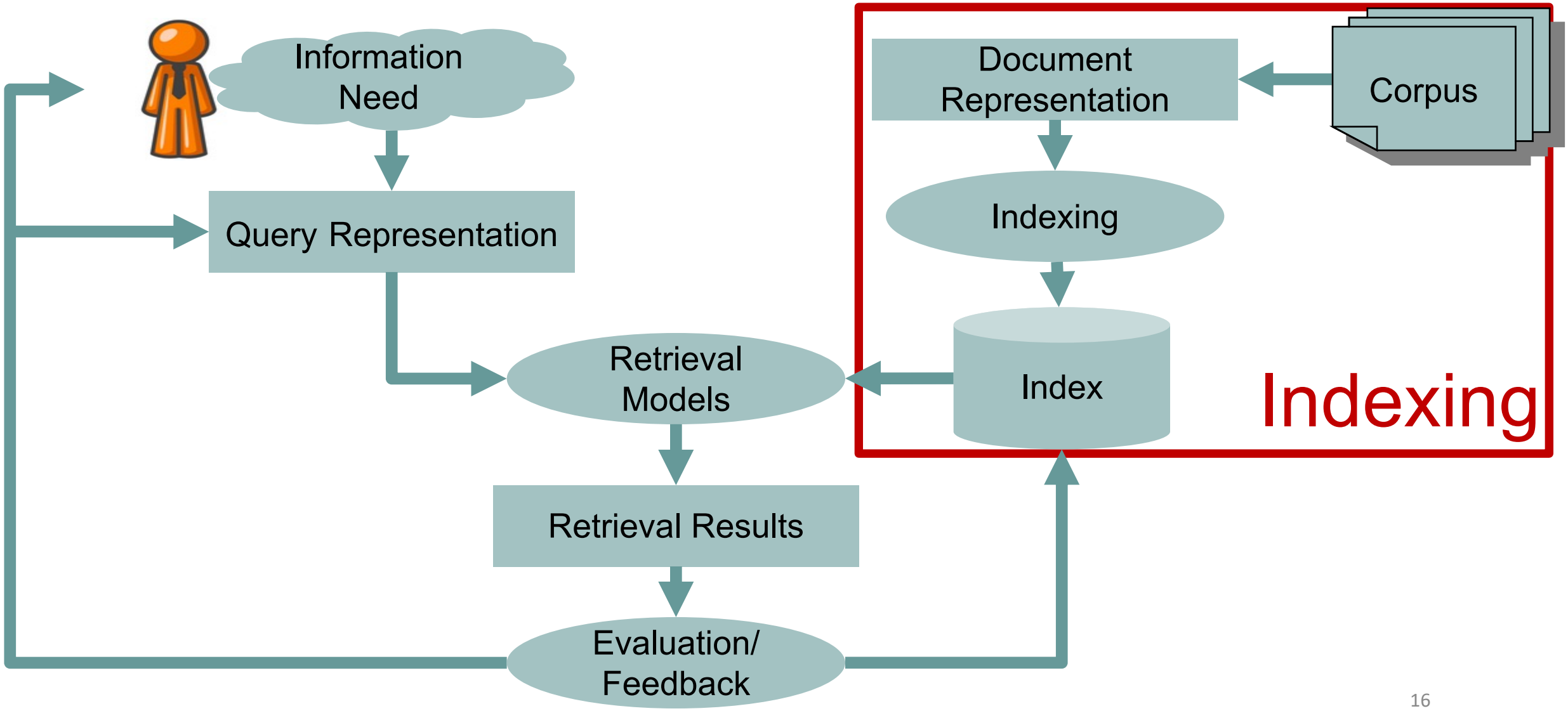
Get rid of mice in a politically correct way

Info about removing mice without killing them

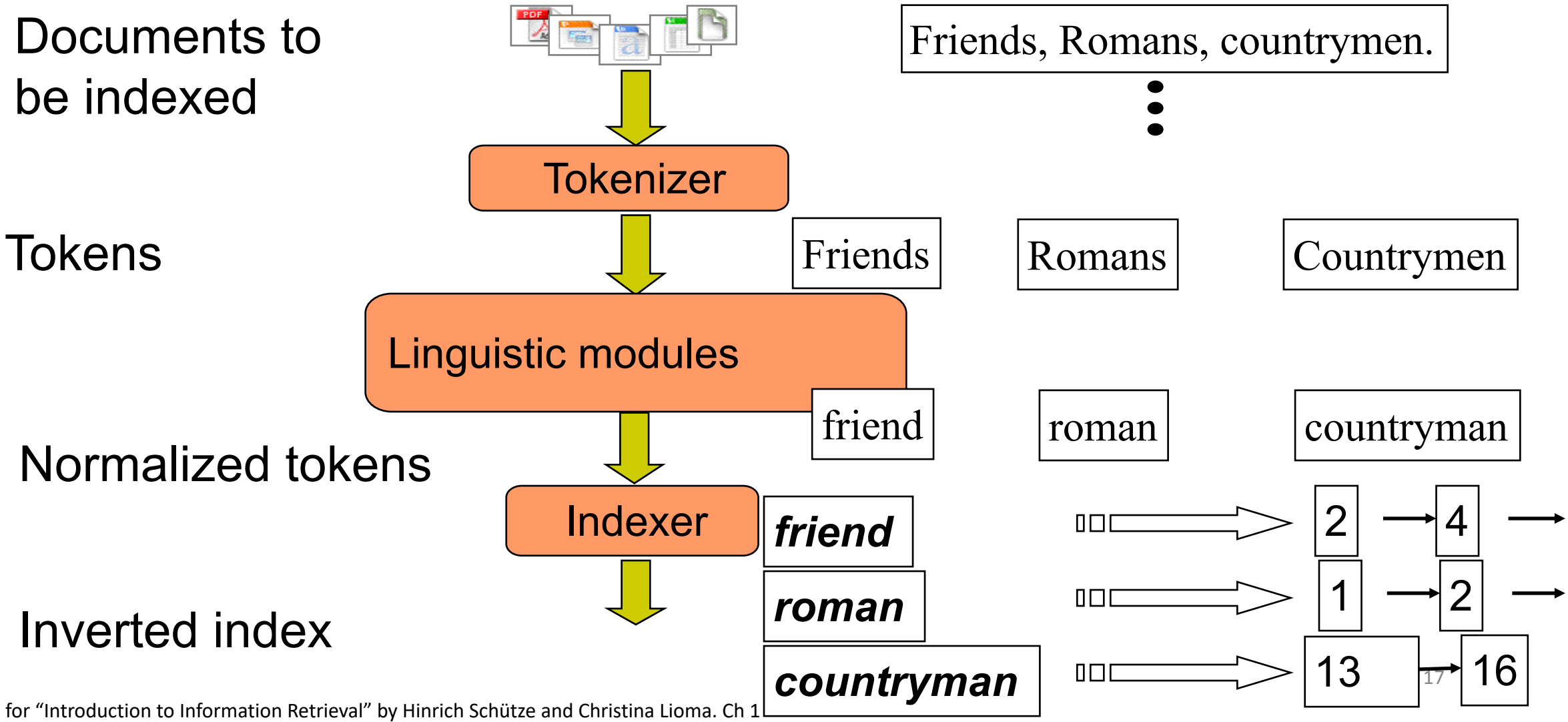
How do I trap mice alive?

Find this: mouse trap any language Search

Document Retrieval Process



Inverted index construction



An Index

- Sequence of (Normalized token, Document ID) pairs.

Doc 1

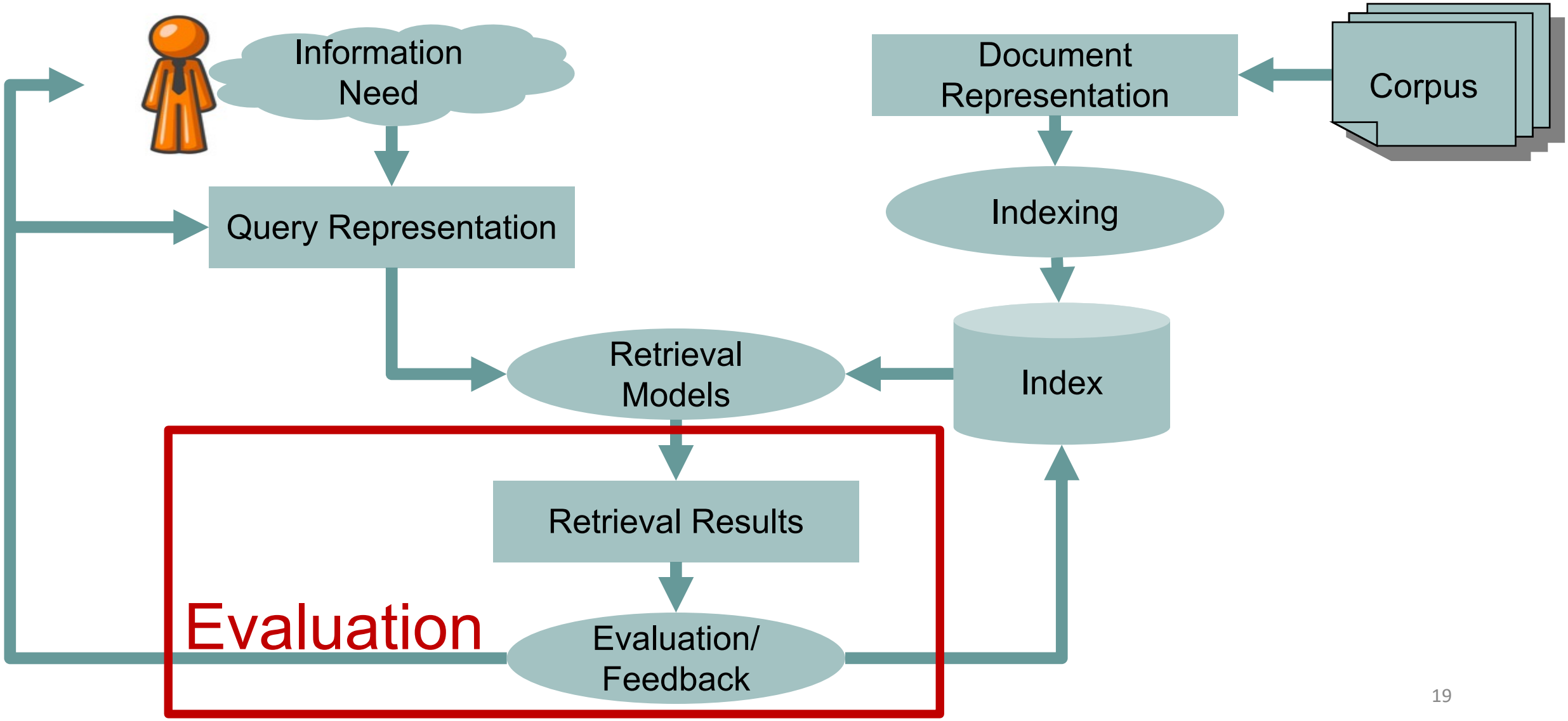
I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

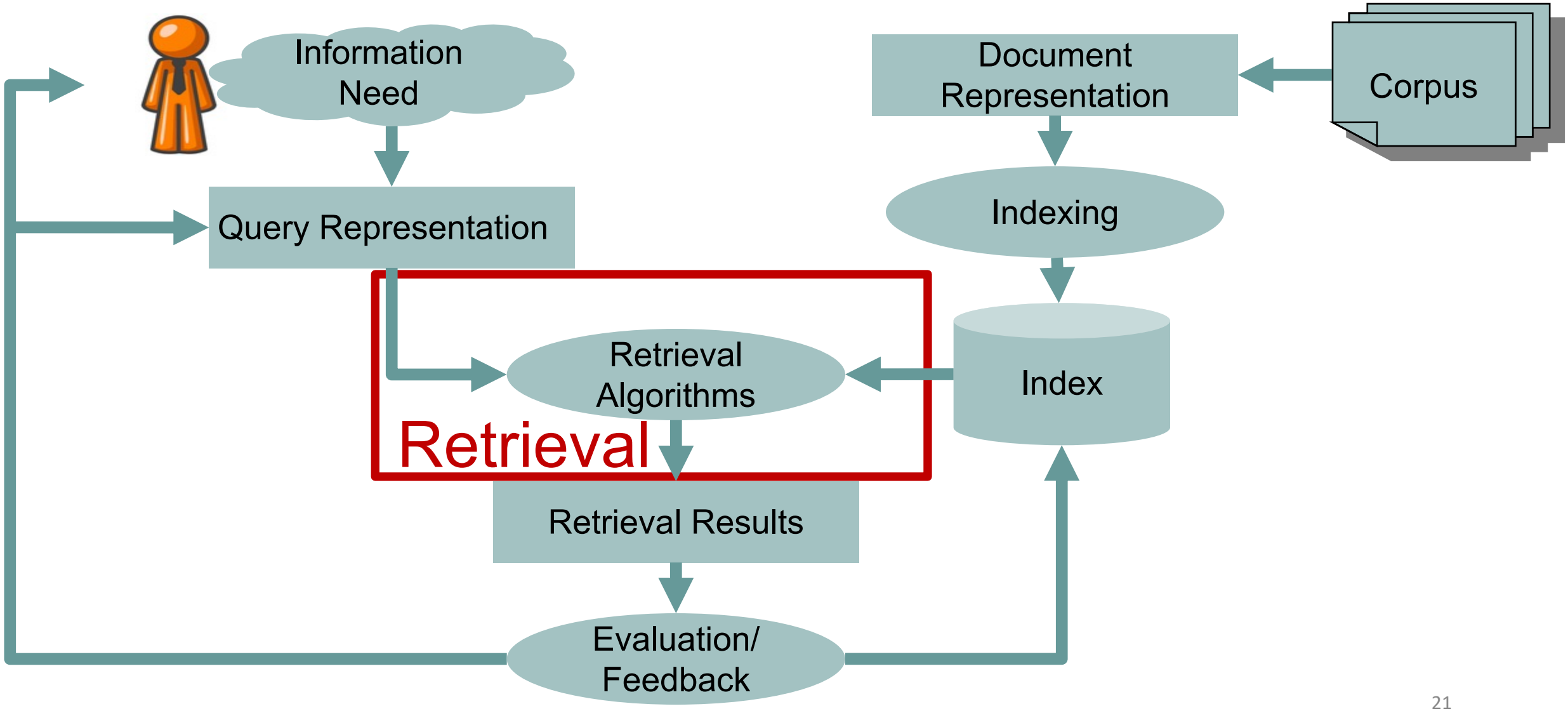
Document Retrieval Process



Evaluation

- Implicit (clicks, time spent) vs. Explicit (yes/no, grades)
- Done by the same user or by a third party (TREC-style)
- Judgments can be binary (Yes/No) or graded
- Assuming ranked or not
- Dimensions under consideration
    - Relevance (Precision, nDCG)
    - Novelty/diversity
    - Usefulness
    - Effort/cost
    - Completeness/coverage (Recall)
    - Combinations of some of the above (F1), and many more
- Relevance is the main consideration. It means
    - If a document (a result) can satisfy the information need
    - If a document contains the answer to my query
- The evaluation lecture (Tuesday by Nicola Ferror and Maria Maistro) will share much more interesting details

Document Retrieval Process



Outline

- What is Information Retrieval
  - Task, Scope, Relations to other disciplines
- Process
  - Preprocessing, Indexing, Retrieval, Evaluation, Feedback
- Retrieval Approaches
  - Boolean
  - Vector Space Model
  - BM25
  - Language Modeling
- Summary
  - What works
  - State-of-the-art retrieval effectiveness
  - Relations to the learning-based approaches

How to find relevant documents for a query?

- By keyword matching
  - boolean model
- By similarity
  - vector space model
- By imaging how to write out a query
  - how likely a query is written with this document in mind
  - generate with some randomness
  - query generation language model
- By trusting how other web pages think about the web page
  - pagerank, hits
- By trusting how other people find relevant documents for the same/similar query
  - Learning to rank

Boolean Retrieval

- Views each document as a set of words
- Boolean Queries use AND, OR and NOT to join query terms
  - Simple SQL-like queries
  - Sometimes with weights attached to each component
- It is like exact match: document matches condition or not
  - Perhaps the simplest model to build an IR system
- Many current search systems are still using Boolean
  - Professional searchers who want to under control of the search process
  - e.g. doctors and lawyers write very long and complex queries with Boolean operators

Summary: Boolean Retrieval

- Advantages:
  - Users are under control of the search results
  - The system is nearly transparent to the user
- Disadvantages:
  - Only give inclusion or exclusion of docs, not rankings
  - Users would need to spend more effort in manually examining the returned sets; sometimes it is very labor intensive
  - No fuzziness allowed so the user must be very precise and good at writing their queries
    - However, in many cases users start a search because they don't know the answer (document)

Ranked Retrieval

- Often we want to rank results
  - from the most relevant to the least relevant
  - Users are lazy
    - maybe only look at the first 10 results
- A good ranking is important
- Given a query q, and a set of documents D, the task is to rank those documents based on a ranking score or relevance score:
  - Score (q,d_i) in the range of [0,1]
  - from the most relevant to the least relevant
- A lot of IR research is about to determine score (q,d_i)

Vector Space Model

Vector Space Model

- Treat the query as a tiny document
- Represent the query and every document each as a word vector in a word space
- Rank documents according to their proximity to the query in the word space

Represent Documents in a Space of Word Vectors

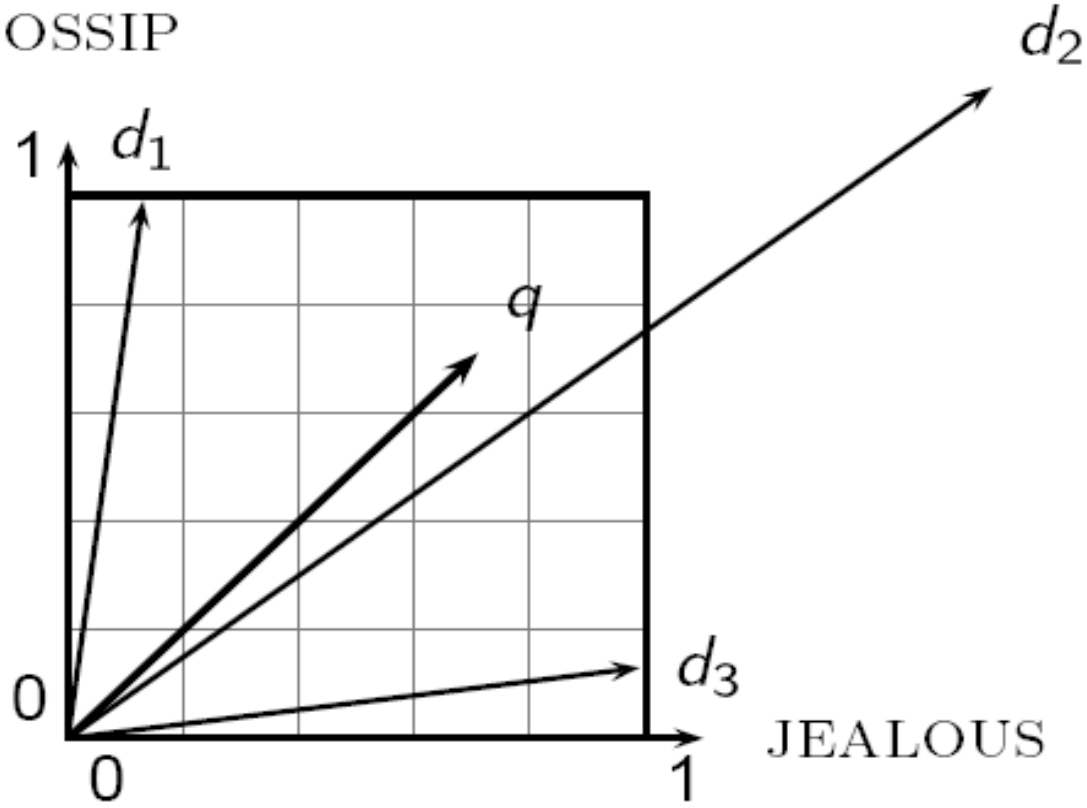
Suppose the corpus only has two words: 'Jealous' and 'Gossip'

They form a space of "Jealous" and "Gossip"

d1: gossip gossip jealous
gossip gossip gossip gossip
gossip gossip gossip gossip

d2: gossip gossip jealous
gossip gossip gossip gossip
gossip gossip gossip jealous
jealous jealous jealous jealous
jealous jealous gossip jealous

d3: jealous gossip jealous
jealous jealous jealous jealous
jealous jealous jealous jealous



q: gossip gossip jealous
gossip gossip gossip gossip
gossip jealous jealous
jealous jealous

Euclidean Distance

- If if p = (p1, p2,..., pn) and q = (q1, q2,..., qn) are two points in the Euclidean space, their Euclidean distance is

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

In a space of 'Jealous' and 'Gossip'

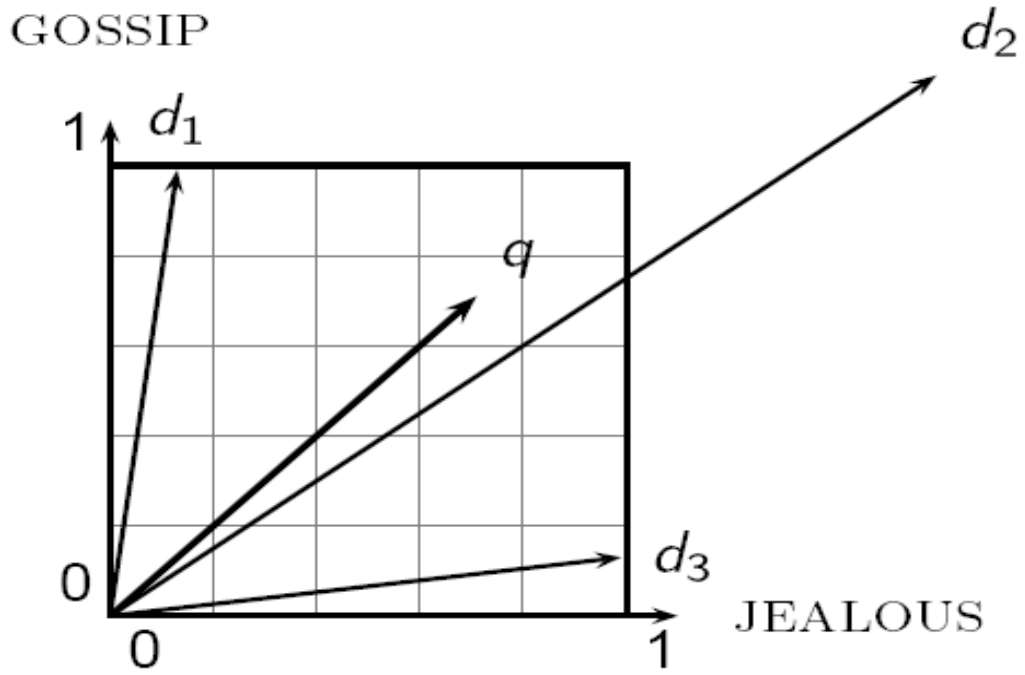
Here, if you look at the content (or we say the word distributions) of each document, d2 is actually the most similar document to q

However, d2 produces a bigger Eclidean distance score to q

d1: gossip gossip jealous
gossip gossip gossip gossip
gossip gossip gossip gossip

d2: gossip gossip jealous
gossip gossip gossip gossip
gossip gossip gossip jealous
jealous jealous jealous jealous
jealous jealous gossip jealous

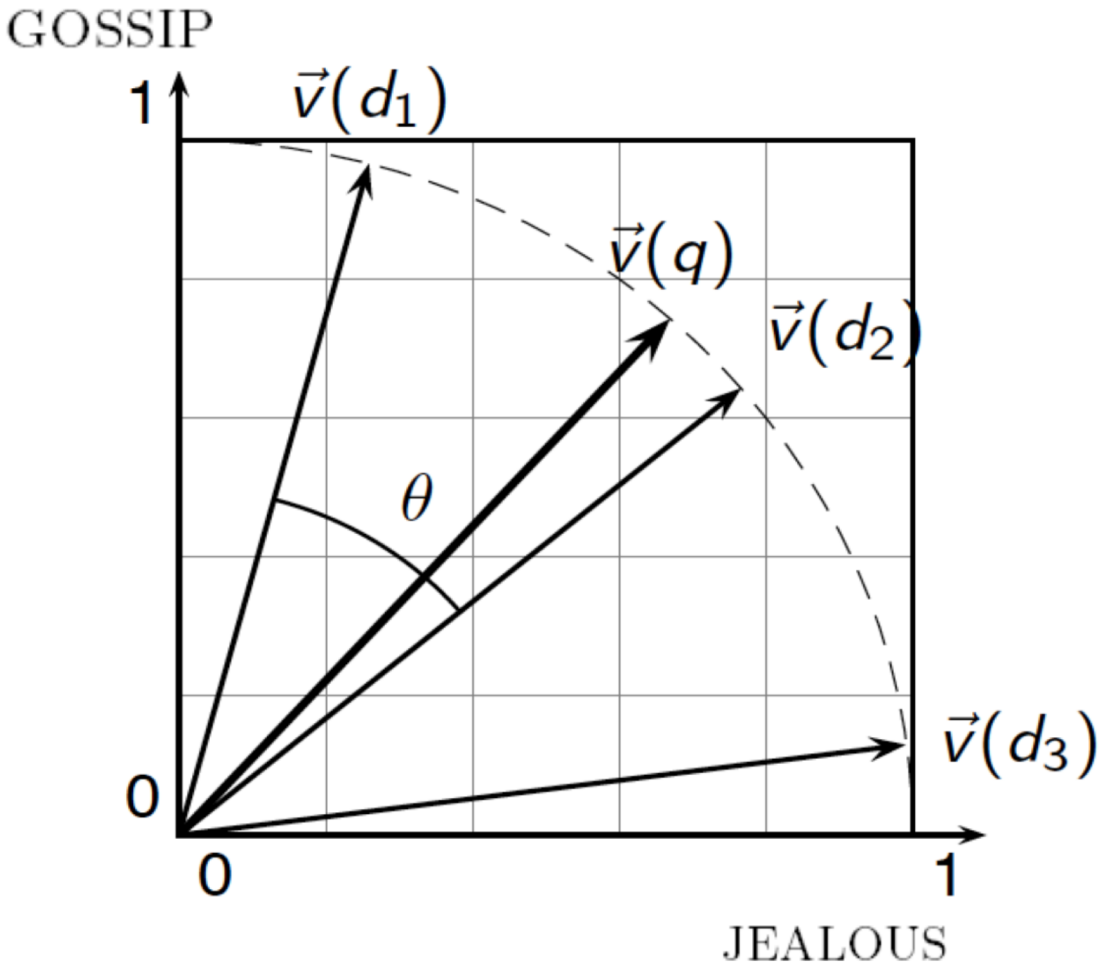
d3: jealous gossip jealous
jealous jealous jealous jealous
jealous jealous jealous jealous



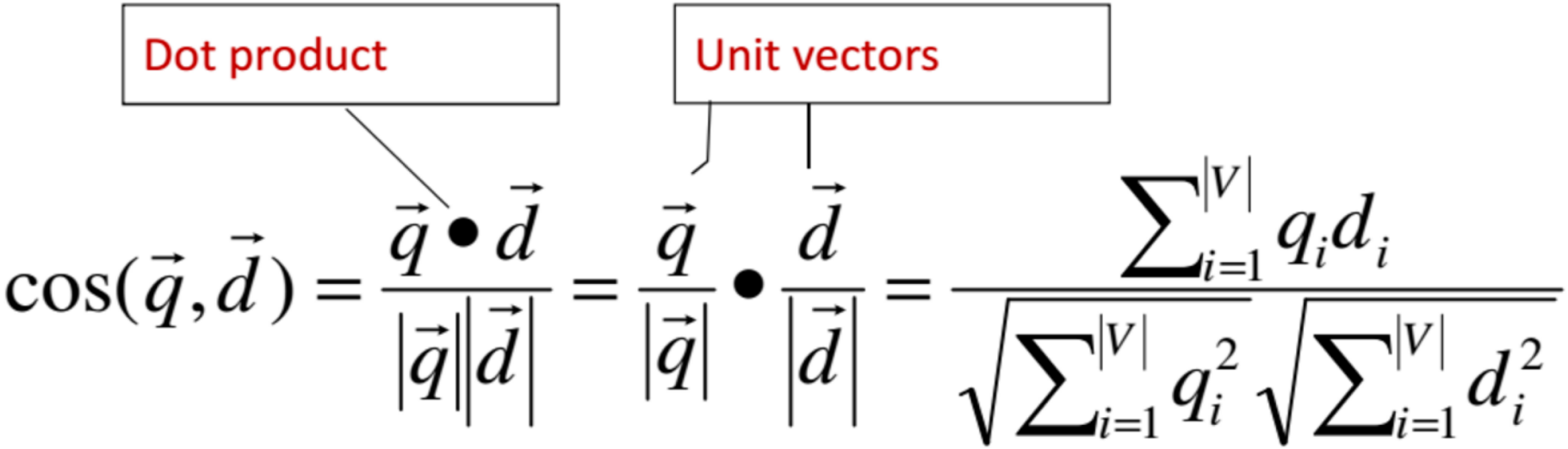
q: gossip gossip jealous
gossip gossip gossip gossip
gossip jealous jealous
jealous jealous

Use angle instead of distance

- Short query and long documents will always have big Euclidean distance
- Key idea: Rank documents according to their angles with query
- The angle between similar vectors is small, between dissimilar vectors is large
- This is equivalent to perform a document length normalization



Cosine Similarity



q_i is the tf-idf weight of term i in the query
d_i is the tf-idf weight of term i in the document

$\cos(\vec{q},\vec{d})$ is the cosine similarity of $\vec{q}$ and $\vec{d}$ … or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$.

Exercise: Cosine Similarity

Consider two documents D_1, D_2 and a query Q, which document is more similar to the query?

$D_1 = (0.5, 0.8, 0.3)$, $D_2 = (0.9, 0.4, 0.2)$,
$Q = (1.5, 1.0, 0)$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

Answers:

Answers:

Consider two documents D_1, D_2 and a query Q

D_1 = (0.5, 0.8, 0.3), D_2 = (0.9, 0.4, 0.2), Q = (1.5, 1.0, 0)

$$Cosine(D_1, Q) = \frac{(0.5 \times 1.5) + (0.8 \times 1.0)}{\sqrt{(0.5^2 + 0.8^2 + 0.3^2)(1.5^2 + 1.0^2)}}$$
$$= \frac{1.55}{\sqrt{(0.98 \times 3.25)}} = 0.87$$

$$Cosine(D_2, Q) = \frac{(0.9 \times 1.5) + (0.4 \times 1.0)}{\sqrt{(0.9^2 + 0.4^2 + 0.2^2)(1.5^2 + 1.0^2)}}$$
$$= \frac{1.75}{\sqrt{(1.01 \times 3.25)}} = 0.97$$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

What are those numbers in a vector?

$D_1 = (0.5, 0.8, 0.3)$, $D_2 = (0.9, 0.4, 0.2)$, $Q = (1.5, 1.0, 0)$

- They are term weights
- They are used to indicate the importance of a term in a document

Term Frequency

- How many times a term appears in a document

- Some terms are common,
  - less common than the stop words
  - but still quite common
    - e.g. “Information Retrieval” is uniquely important in NBA.com
    - e.g. “Information Retrieval” appears at too many pages in SIGIR web site, so it is not a very important term in those pages.
- How to discount their term weights?

Inverse Document Frequency (idf)

- df_t is the document frequency of t
  - the number of documents that contain t
  - it inversely measures how informative a term is
- The IDF of a term t is defined as

$$\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$$

- Log is used here to “dampen” the effect of idf.
- N is the total number of documents
- Note it is a property of the term and it is query independent

tf-idf weighting

- Product of a term's tf weight and idf weight regarding a document

$$w_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}(N / \mathrm{df}_t)$$

- Best known term weighting scheme in IR
- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection
- Note: term frequency takes two inputs (the term and the document) while IDF only takes one (the term)

tf-idf weighting has many variants

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$				

Standard tf-idf weighting scheme: Lnc.ltc

- A very standard weighting scheme is: lnc.ltc
  - Document:
    - L: logarithmic tf (l as first character)
    - N: no idf
    - C: cosine normalization
  - Query:
    - L: logarithmic tf (l in leftmost column)
    - t: idf (t in second column)
    - C: cosine normalization …
- Note: here the weightings differ in queries and in documents

Summary: Vector Space Model

- Advantages
  - Simple computational framework for ranking documents given a query
  - Any similarity measure or term weighting scheme could be used
- Disadvantages
  - Assumption of term independence
  - Ad hoc

BM25

The *(Magical)* Okapi BM25 Model

- BM25 is one of the most successful retrieval models
- It is a special case of the Okapi models
  - Its full name is Okapi BM25
- It considers the length of documents and uses it to normalize the term frequency
- It is virtually a probabilistic ranking algorithm though it looks very ad-hoc
- It is intended to behave similarly to a two-Poisson model
- We will talk about Okapi in general

What is Behind Okapi?

- [Robertson and Walker 94]
- A two-Poisson document-likelihood Language model
  - Models within-document term frequencies by means of a mixture of two Poisson distributions
- Hypothesize that occurrences of a term in a document have a random or stochastic element
  - It reflects a real but hidden distinction between those documents which are “about” the concept represented by the term and those which are not.
- Documents which are “about” this concept are described as “elite” for the term.
- Relevance to a query is related to eliteness rather than directly to term frequency, which is assumed to depend only on eliteness.

Two-Poisson Model

- Term weight for a term t:

$$w = \log \frac{(p'\lambda^{tf}e^{-\lambda} + (1-p')\mu^{tf}e^{-\mu})\ (q'e^{-\lambda} + (1-q')e^{-\mu})}{(q'\lambda^{tf}e^{-\lambda} + (1-q')\mu^{tf}e^{-\mu})\ (p'e^{-\lambda} + (1-p')e^{-\mu})}$$

where lambda and mu are the Poisson means for tf
In the elite and non-elite sets for t

p’ = P(document elite for t| R)
q’ = P(document elite for t| NR)

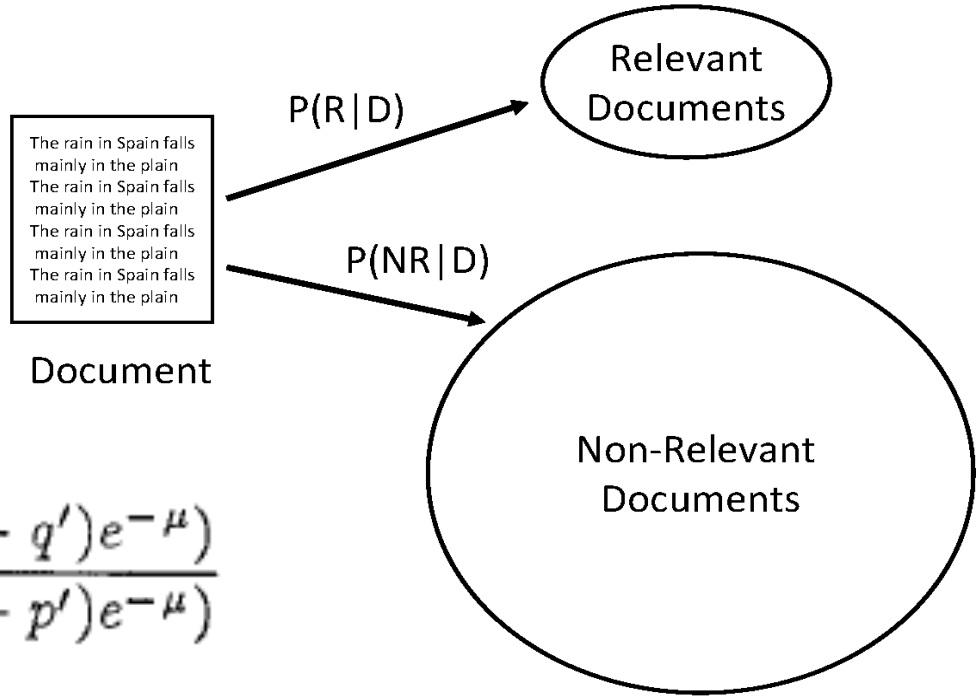


Figure adapted from “Search Engines: Information Retrieval in Practice” Chap 7

Characteristics of Two-Poisson Model

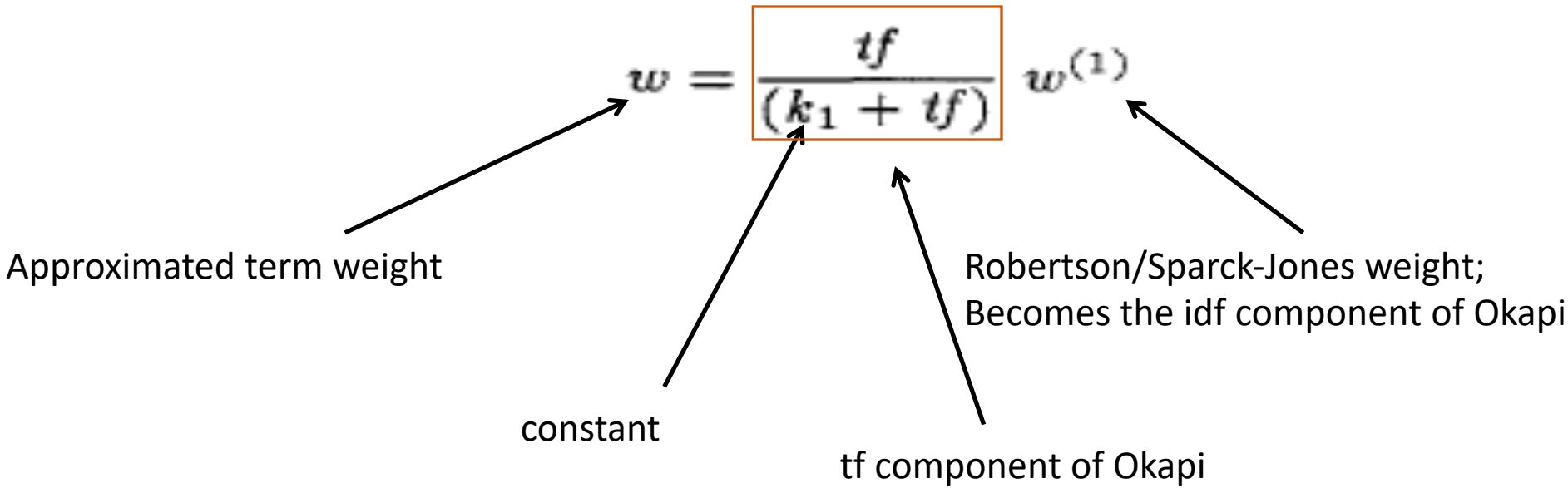
- It is zero for tf=0;
- It increases monotonically with tf;
- but to an asymptotic maximum;
- The maximum approximates to the Robertson/Sparck-Jones weight that would be given to a direct indicator of eliteness.

$$w = \log \frac{p(1-q)}{q(1-p)},$$

p = P(term present| R)
q = P(term present| NR)

Constructing a Function

- Constructing a function
  - Such that tf/(constant + tf) increases from 0 to an asymptotic maximum
- A rough estimation of 2-poisson



Okapi Model

- The complete version of Okapi BMxx models

$$\sum_{t \in Q} \left(\log \frac{N - df_t + 0.5}{df_t + 0.5} \right) \frac{(k_1 + 1) tf_t}{k_1 \left((1 - b) + b \frac{doclen}{avg\_doclen} \right) + tf_t} \frac{(k_3 + 1) qtf_t}{k_3 + qtf_t}$$

idf (Robertson-Sparck Jones weight)

tf

user related weight

Original Okapi: k1 = 2, b=0.75, k3 = 0

BM25: k1 = 1.2, b=0.75, k3 = a number from 0 to 1000

Exercise: Okapi BM25

- Query with two terms, “president lincoln”, (*qtf = 1)*
- No relevance information (*r and R are* zero)
- N = 500,000 documents
- *“president”* occurs in 40,000 documents (df_1 = 40, 000)
- *“lincoln”* occurs in 300 documents (df_2 = 300)
- “president” occurs 15 times in the doc (tf_1 = 15)
- *“lincoln”* occurs 25 times in the doc (tf_2 = 25)
- document length is 90% of the average length (*dl/avdl* = .9)
- k_1 = 1.2, b = 0.75, and k_3 = 100
- $K = 1.2 \cdot (0.25 + 0.75 \cdot 0.9) = 1.11$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

Answer:

Answer: Okapi BM25

$$
\begin{aligned}
BM25(Q,D) \quad = \quad & \\
& \log \frac{(0+0.5)/(0-0+0.5)}{(40000-0+0.5)/(500000-40000-0+0+0.5)} \\
& \times \frac{(1.2+1)15}{1.11+15} \times \frac{(100+1)1}{100+1} \\
& + \log \frac{(0+0.5)/(0-0+0.5)}{(300-0+0.5)/(500000-300-0+0+0.5)} \\
& \times \frac{(1.2+1)25}{1.11+25} \times \frac{(100+1)1}{100+1}
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \log 460000.5/40000.5 \cdot 33/16.11 \cdot 101/101 \\
& + \log 499700.5/300.5 \cdot 55/26.11 \cdot 101/101 \\
= \quad & 2.44 \cdot 2.05 \cdot 1 + 7.42 \cdot 2.11 \cdot 1 \\
= \quad & 5.00 + 15.66 = 20.66
\end{aligned}
$$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

Effect of term frequencies in BM25

Frequency of "president"	Frequency of "lincoln"	BM25 score
15	25	20.66
15	1	12.74
15	0	5.00
1	25	18.2
0	25	15.66

Language Modeling

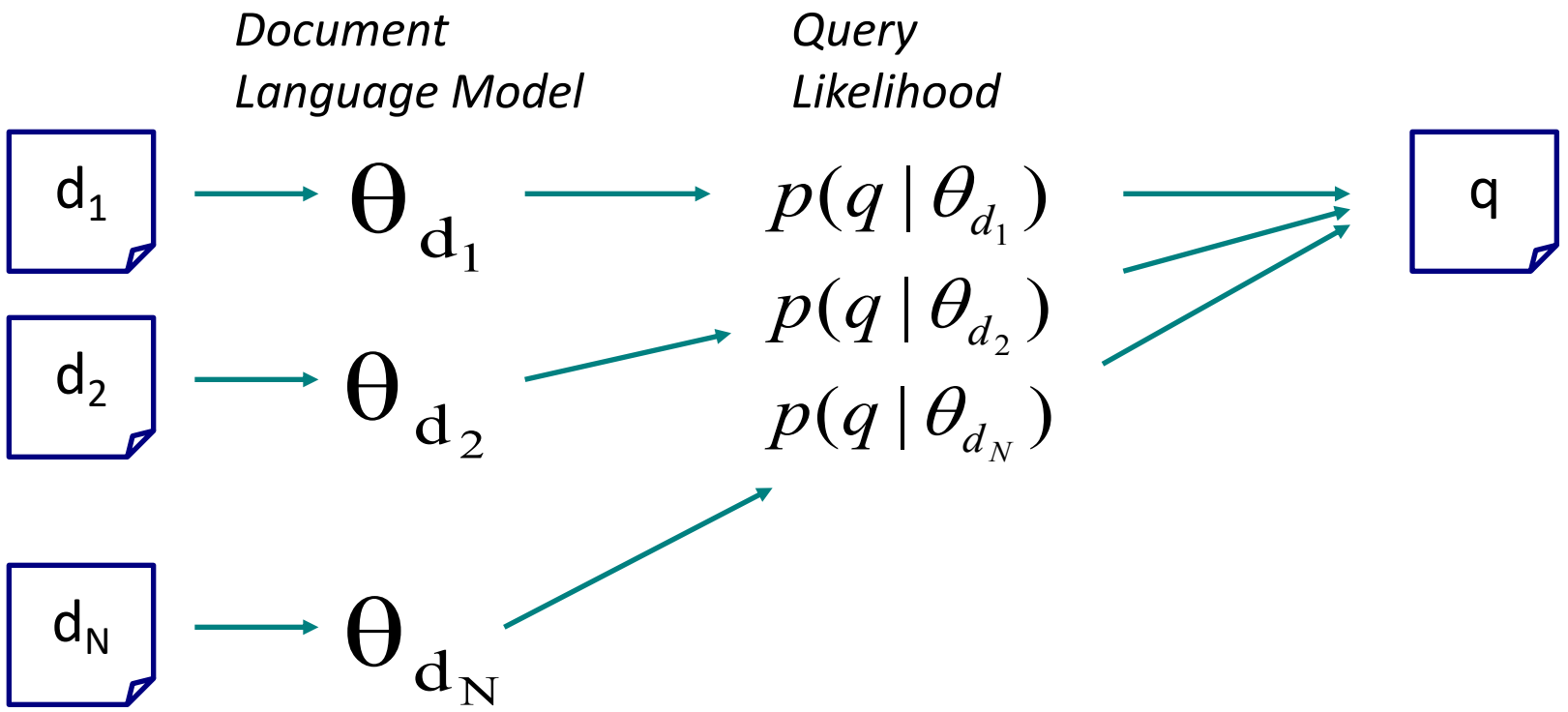
Using language models in IR

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

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

  - $P(q)$ 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 prior to "high-quality" documents, e.g., those with high PageRank.
  - $P(q|d)$ is the probability of *q* given *d*
- Ranking according to $P(q|d)$ and $P(d|q)$ is equivalent

Query-likelihood LM



- Scoring documents with query likelihood
- Known as *the* language modeling (LM) approach to IR

Adapted from Mei, Fang and Zhai's "A study of poison query generation model in IR"

A different language model for each document

language model of d_1

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.01
the	.2	said	.03
a	.1	likes	.02
frog	.01	that	.04
		...	...

language model of d_2

w	$P(w\|.)$	w	$P(w\|.)$
STOP	.2	toad	.02
the	.15	said	.03
a	.08	likes	.02
frog	.01	that	.05
		...	...

String = frog said that toad likes frog STOP

$P(\text{string}|M_{d1}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.02 = 0.0000000000048 = 4.8 \cdot 10^{-12}$

$P(\text{string}|M_{d2}) = 0.01 \cdot 0.03 \cdot 0.05 \cdot 0.02 \cdot 0.02 \cdot 0.01 \cdot 0.02 = 0.0000000000120 = 12 \cdot 10^{-12}$ $P(\text{string}|M_{d1}) < P(\text{string}|M_{d2})$

Thus, document d_2 is "more relevant" to the string "frog said that toad likes frog STOP" than d_1 is.

Binomial Distribution

- Discrete
- Series of trials with only two outcomes, each trial being independent from all the others
- Number *r* of successes out of *n* trials given that the probability of success in any trial is θ :

$$b(r;n,\theta) = \binom{n}{r}\theta^r(1-\theta)^{n-r}$$

Multinomial Distribution

- The multinomial distribution is a generalization of the binomial distribution.

- The binomial distribution counts successes of an event (for example, heads in coin tosses).
- The parameters:
  - *N* (number of trials)
  - θ (the probability of success of the event)

- The multinomial counts the number of a set of events (for example, how many times each side of a die comes up in a set of rolls).
  - The parameters:
  - *N* (number of trials)
  - $\theta_1 .. \theta_k$(the probability of success for each category)

Each is estimated by Maximum Likelihood Estimation (MLE)

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

Multinomial Distribution

- $W_1, W_2, .. W_k$ are variables

Number of possible orderings of N balls

$$P(W_1 = n_1, ..., W_1 = n_k \mid N, \theta_1, .., \theta_k) = \frac{N!}{n_1! n_2! .. n_k!} \theta_1^{n_1} \theta_2^{n_2} .. \theta_k^{n_k}$$

order invariant selections

$$\sum_{i=1}^{k} n_i = N \quad \sum_{i=1}^{k} \theta_i = 1$$

A binomial distribution is the multinomial distribution with k=2 and $\theta_1, \theta_2 = 1 - \theta_2$

Assume events (terms being generated) are independent

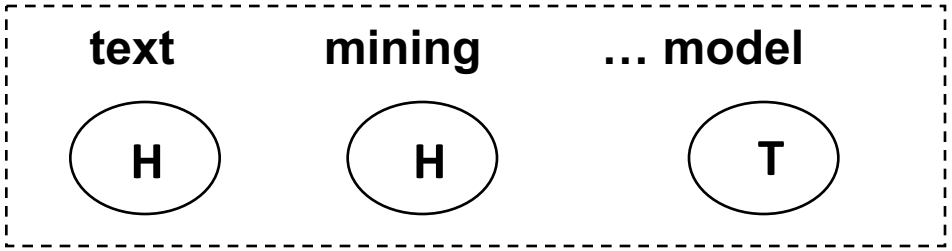
Multi-Bernoulli vs. Multinomial

Doc: d

text
mining
model
clustering
text
model
text
…

Multi-Bernoulli:

Flip a coin for each word

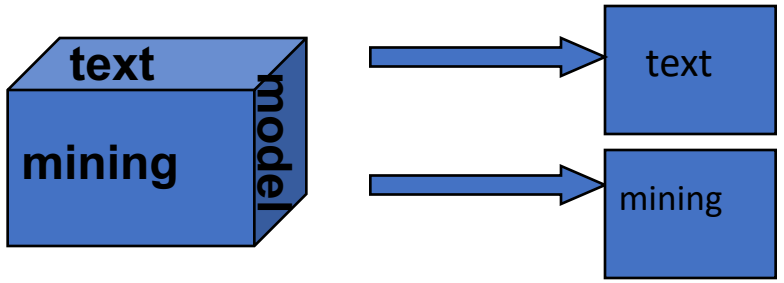


Query q:
"text mining"

$$p(q \mid d) = \prod_{w \in q} p(w = 1 \mid d) \prod_{w \notin q} p(w = 0 \mid d)$$

Multinomial:

Roll a dice to choose a word



Query q:
"text mining"

$$p(q \mid d) = \prod_{j=1}^{|V|} p(w_j \mid d)^{c(w_j, q)}$$

Adapted from Mei, Fang and Zhai's "A study of poison query generation model in IR"

Issue

- Issue: a single t with $P(t|M_d) = 0$ will make $P(q|M_d) = \prod P(t|M_d)$ zero
- Smooth the estimates to avoid zeros

Dirichlet Distribution & Conjugate Prior

- If the prior and the posterior are the same distribution, the prior is called a **conjugate prior** for the likelihood

- The Dirichlet distribution is the conjugate prior for the multinomial, just as beta is conjugate prior for the binomial.

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

Gamma function

Dirichlet Smoothing

- Let's say the prior for $\theta_1, .., \theta_k$ is

$$Dir(\alpha_1, .., \alpha_k)$$

- From observations to the data, we have the following counts $n_1, .., n_k$
- The posterior distribution for $\theta_1, .., \theta_k$, given the data, is

$$Dir(\alpha_1 + n_1, .., \alpha_k + n_k)$$

- So the prior works like pseudo-counts
  - it can be used for smoothing

JM Smoothing:

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

- Also known as *the Mixture Model*
- Mixes the probability from the document with the general collection frequency of the word.
- Correctly setting λ is very important for good performance.
  - 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

Poisson Query-likelihood LM

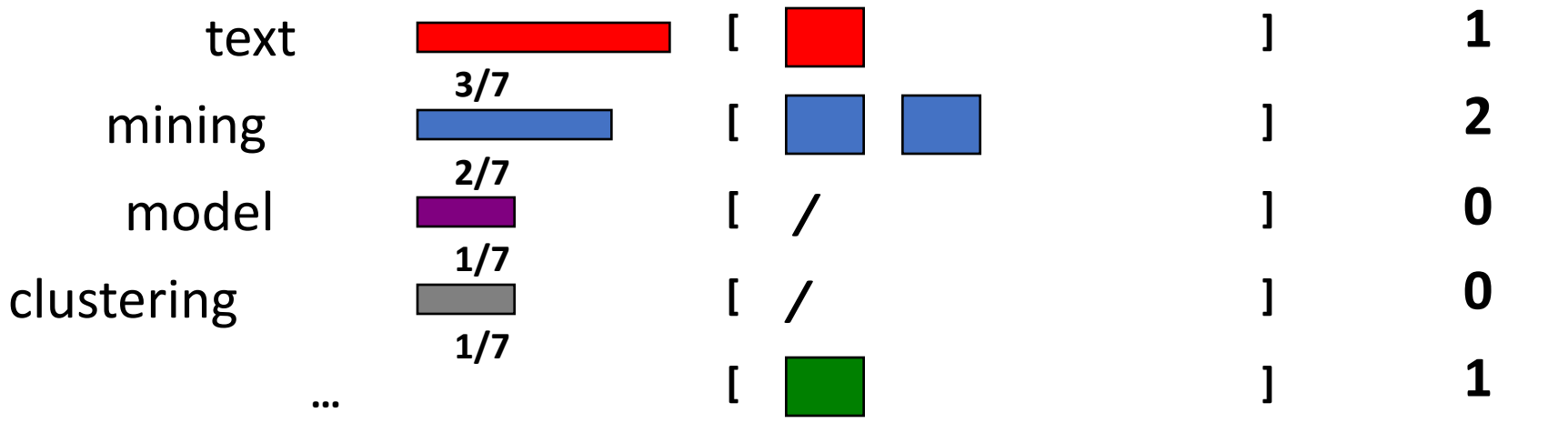
Poisson:
Each term is written

text
mining
model
mining
text
clustering
text
…

Receiver: Query

Rates of arrival : λ_i

Duration: **|q|**



Query q :
"mining text mining systems"

$$p(q \mid d) = \frac{e^{-3/7|q|}(\frac{3}{7}|q|)^1}{1!} \frac{e^{-2/7|q|}(\frac{2}{7}|q|)^2}{2!} \frac{e^{-1/7|q|}(\frac{1}{7}|q|)^0}{0!} \frac{e^{-1/7|q|}(\frac{1}{7}|q|)^0}{0!} \frac{e^{-\lambda_i|q|}(\lambda_i|q|)^1}{1!}$$

Slides adapted from Mei, Fang and Zhai's "A study of poison query generation model in IR"

Comparison

	multi-Bernoulli	multinomial	Poisson
$p(q \mid d)$	$\prod_{w \in q} p(w=1 \mid d) \prod_{w \notin q} p(w=0 \mid d)$	$\prod_{j=1}^{\lvert V \rvert} p(w_j \mid d)^{c(w_j, q)}$	$\prod_{j=1}^{\lvert V \rvert} p(c(w_j, q) \mid d)$
Event space	Appearance /absence	*Vocabulary*	frequency
Model frequency?	No	Yes	Yes
Model length? (document/query)	No	Implicitly yes	Yes
w/o Sum-to-one constraint?	Yes	No	Yes
Per-Term Smoothing	Easy	Hard	Easy
Closed form solution for mixture of models?	No	No	Yes

Slides adapted from Mei, Fang and Zhai's "A study of poison query generation model in IR"

Summary: Language Modeling

- LM vs. VSM:
  - LM: based on probability theory
  - VSM: based on similarity, a geometric/ linear algebra notion
- Modeling term frequency in LM is better than just modeling term presence/absence
  - Multinomial model performs better than multi-Bernoulli
- Mixture of Multinomials for the background smoothing model has been shown to be effective for IR
  - LDA-based retrieval [Wei & Croft SIGIR 2006]
  - PLSI [Hofmann SIGIR '99]
- Probabilities are inherently "length-normalized"
  - When doing parameter estimation
- Mixing document 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.

Outline

- What is Information Retrieval
  - Task, Scope, Relations to other disciplines
- Process
  - Preprocessing, Indexing, Retrieval, Evaluation, Feedback
- Retrieval Approaches
  - Boolean
  - Vector Space Model
  - BM25
  - Language Modeling
- Summary
  - What works?
  - State-of-the-art retrieval effectiveness – what should you expect?
  - Relations to the learning-based approaches

What works?

- Term Frequency (tf)
- Inverse Document Frequency (idf)
- Document length normalization
- Okapi BM25
  - Seems ad-hoc but works so well (popularly used as a baseline)
  - Created by human experts, not by data
- Other more justified methods could achieve similar effectiveness as BM25
  - They help better deep understanding of IR, related disciplines

What might not work?

- You might have heard of other topics/techniques, such as
  - Pseudo-relevance feedback
  - Query expansion
  - N-gram instead of unit gram
  - Semantically-heavy annotations
  - Sophisticated understanding of documents
  - Personalization (Read a lot into the user)
- .. But they usually don't work reliably (not as much as what we expect and sometimes worsen the performance)
  - Maybe more research needs to be done
  - Or, maybe they are not the right directions

At the heart is the metric

- How our users feel good about the search results
  - Sometimes it could be subjective
- The approaches that we discusses today do not directly optimize the metrics (P, R, nDCG, MAP etc)
- These approaches are considered more conventional, without making use of large amount of data that can be learned models from
- Instead, they are created by researchers based on their own understanding of IR and they hand-crafted or imagined most of the models
  - And these models work very well
  - Salute to the brilliant minds

Learning-based Approaches

- More recently, learning-to-rank has become the dominating approach
  - Due to vast amount of logged data from Web search engines
- The retrieval algorithm paradigm
  - Has become data-driven
  - Requires large amount of data from massive users
  - IR is formulated as a supervised learning problem
  - directly uses the metrics as the optimization objectives
  - No longer guess what a good model should be, but leave to the data to decide
- The Deep learning lecture (Thursday by Bhaskar Mitra, Nick Craswell, and Emine Yilmaz) will introduce them in depth

References

- IR Textbooks used for this talk:
  - Introduction to Information Retrieval. C.D. Manning, P. Raghavan, H. Schütze. Cambridge UP, 2008.
  - Foundations of Statistical Natural Language Processing. Christopher D. Manning and Hinrich Schütze.
  - Search Engines: Information Retrieval in Practice. W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009.
  - Modern Information Retrieval: The Concepts and Technology behind Search. by Ricardo Baeza-Yates, Berthier Ribeiro-Neto. Second condition. 2011.
- Main IR research papers used for this talk:
  - Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. Robertson, S. E., & Walker, S. SIGIR 1994.
  - Document Language Models, Query Models, and Risk Minimization for Information Retrieval. Lafferty, John and Zhai, Chengxiang. SIGIR 2001.
  - A study of Poisson query generation model for information retrieval. Qiaozhu Mei, Hui Fang, Chengxiang Zhai. SIGIR 2007.
- Course Materials/presentation slides used in this talk:
  - Barbara Rosario's "Mathematical Foundations" lecture notes for textbook "Statistical Natural Language Processing"
  - Textbook slides for "Search Engines: Information Retrieval in Practice" by its authors
  - Oznur Tastan's recitation for 10601 Machine Learning
  - Textbook slides for "Introduction to Information Retrieval" by Hinrich Schütze and Christina Lioma
  - CS276: Information Retrieval and Web Search by Pandu Nayak and Prabhakar Raghavan
  - 11-441: Information Retrieval by Jamie Callan
  - A study of Poisson query generation model for information retrieval. Qiaozhu Mei, Hui Fang, Chengxiang Zhai

Thank You

Dr. Grace Hui Yang
InfoSense
Georgetown University, USA

Contact: huiyang@cs.georgetown.edu