Overview of Text Retrieval (TR)

Intelligent Information Retrieval

Lecture Plan

- What is text retrieval (TR) ?
- Document selection vs. document ranking
- Major research milestones
- Components in a TR system

What is Text Retrieval (TR)?

- There exists a collection of text documents
- User gives a query to express the information need
- A retrieval system returns relevant documents to users
- More often called “information retrieval” (IR) , but IR is actually much broader
  - May include non-textual information
  - May include text categorization or summarization…
- Known as “search technology” in industry

TR vs. Database Retrieval

- Information
  - Unstructured/free text vs. structured data
  - Ambiguous vs. well-defined semantics
- Query
  - Ambiguous vs. well-defined semantics
  - Incomplete vs. complete specification
- Answers
  - Relevant documents vs. matched records
- TR is an empirically defined problem!

TR is Hard!

- Under/over-specified query
  - Ambiguous: “buying CDs” (money or music?)
  - Incomplete: what kind of CDs?
  - What if “CD” is never mentioned in document?
- Vague semantics of documents
  - Ambiguity: e.g., word-sense, structural
  - Incomplete: Inferences required
- Even hard for people!
  - 80% agreement in human judgments

TR is "Easy"!

- TR CAN be easy in a particular case
  - Ambiguity in query/document is RELATIVE to the database
  - So, if the query is SPECIFIC enough, just one keyword may get all the relevant documents
- PERCEIVED TR performance is usually better than the actual performance
  - Users can NOT judge the completeness of an answer

Short vs. Long Term Info Need

- Short-term information need (Ad hoc retrieval)
  - "Temporary need", e.g., info about used cars
  - Information source is relatively static
  - User "pulls" information
  - Application example: library search, Web search
- Long-term information need (Filtering)
  - "Stable need", e.g., new data mining algorithms
  - Information source is dynamic
  - System "pushes" information to user
  - Applications: news filter

Importance of Ad hoc Retrieval

- Directly manages any existing large collection of information
- There are many many “ad hoc” information needs
- A long-term information need can be satisfied through frequent ad hoc retrieval
- Basic techniques of ad hoc retrieval can be used for filtering and other “non-retrieval” tasks, such as categorization, clustering, and automatic summarization.

Lecture Plan

- What is text retrieval (TR) ?
- Document selection vs. document ranking
- Major research milestones
- Components in a TR system

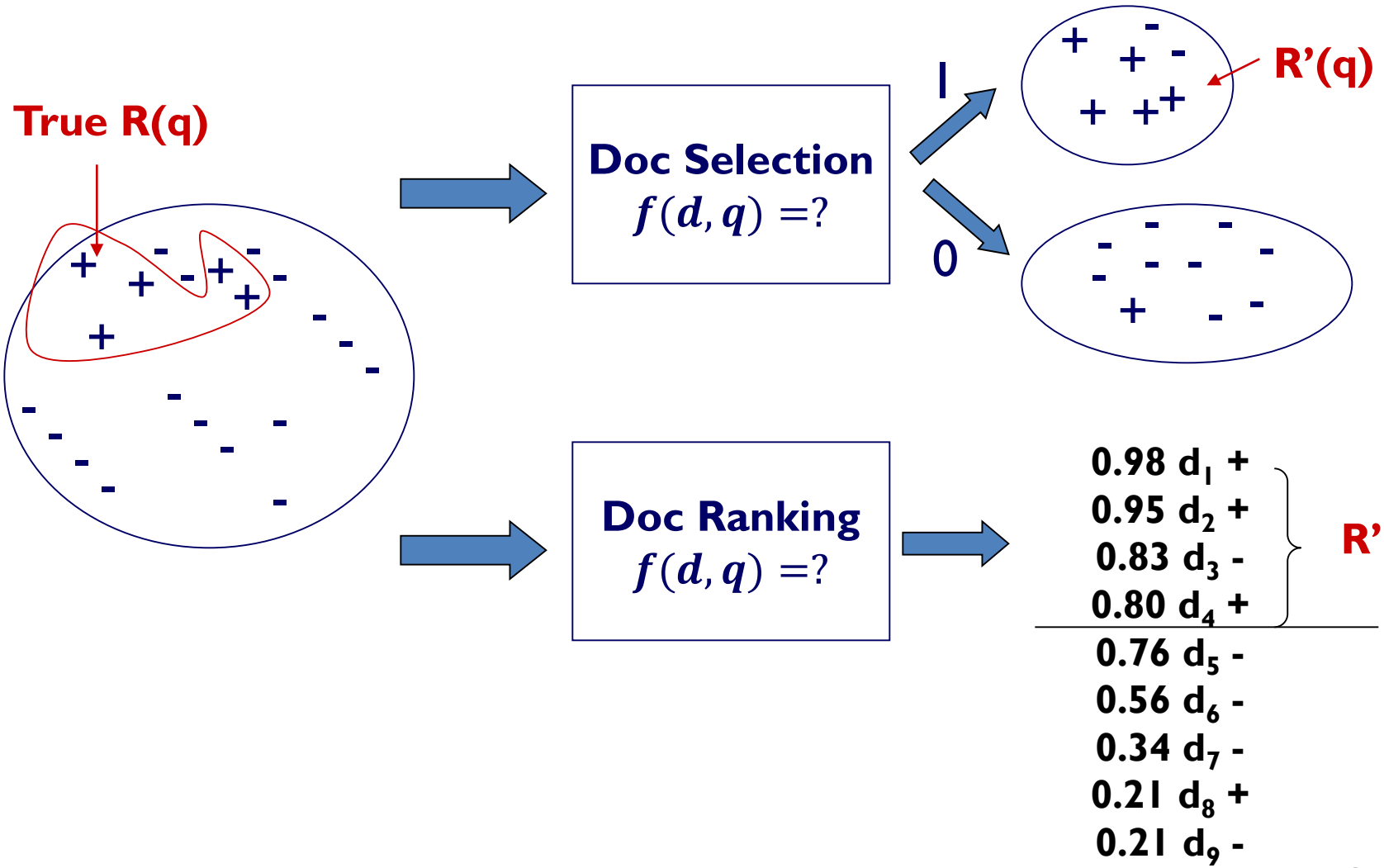
Formal Formulation of TR

- Vocabulary $V = \{w_1, w_2, \ldots, w_N\}$ of language
- Query $q = q_1, \ldots, q_m$, where $q_i \in V$
- Document $d_i = d_{i1}, \ldots, d_{im_i}$, where $d_{ij} \in V$
- Collection $C = \{d_1, \ldots, d_k\}$
- Set of relevant documents $R(q) \subseteq C$
  - Generally unknown and user-dependent
  - Query is a "hint" on which doc is in $R(q)$
- Task = compute $R'(q)$, an "approximate $R(q)$"

Computing R(q)

- Strategy 1: Document selection
  - $R(q) = \{d \in C | f(d, q) = 1\}$, where $f(d, q) \in \{0,1\}$ is an indicator function or classifier
  - System must decide if a doc is relevant or not ("absolute relevance")
- Strategy 2: Document ranking
  - $R(q) = \{d \in C | f(d, q) > \theta\}$, where $f(d, q) \in \Re$ is a relevance measure function; θ is a cutoff
  - System must decide if one doc is more likely to be relevant than another ("relative relevance")

Document Selection vs. Ranking



Problems of Doc Selection

- The classifier is unlikely accurate
  - "Over-constrained" query (terms are too specific): no relevant documents found
  - "Under-constrained" query (terms are too general): over delivery
  - It is extremely hard to find the right position between these two extremes
- Even if it is accurate, all relevant documents are not equally relevant
- Relevance is a matter of degree!

Ranking is often preferred

- Relevance is a matter of degree
- A user can stop browsing anywhere, so the boundary is controlled by the user
  - High recall users would view more items
  - High precision users would view only a few

We will talk about many different ranking methods later…

Lecture Plan

- What is text retrieval (TR) ?
- Document selection vs. document ranking
- Major research milestones
- Components in a TR system

Major Research Milestones

- Early days (late 1950s to 1960s): foundation and founding of the field
  - Luhn's work on automatic encoding
  - Cleverdon's Cranfield evaluation methodology and index experiments
  - Salton's early work on SMART system and experiments
- 1970s-1980s: a large number of retrieval models
  - Vector space model
  - Probabilistic models
- 1990s: further development of retrieval models and new tasks
  - Language models
  - TREC evaluation
- 2000s-present: more applications, especially Web search and interactions with other fields
  - Web search
  - Learning to rank
  - Scalability (e.g., MapReduce)

Indexing: auto vs. manual

Evaluation System

Indexing + Search Theory

Large-scale evaluation, beyond ad hoc retrieval

Web search
Machine learning
Scalability

Background: library search in 1950s

Card catalogue of Yale Univ.'s Sterling Memorial Library
(picture from Wikipedia)

Index cards are sorted in alphabetical orders:
- Title index
- Author index
- Subject index

Users can only sequentially search for items

Indexing was done manually

Clear separation of indexing and search

A typical title card (sorted by title)

F
Kee

The Clue of the Velvet Mask.

Keene, Carolyn.

The Clue of the Velvet Mask/ Carolyn Keene.

New York, Grosset, 1969c.

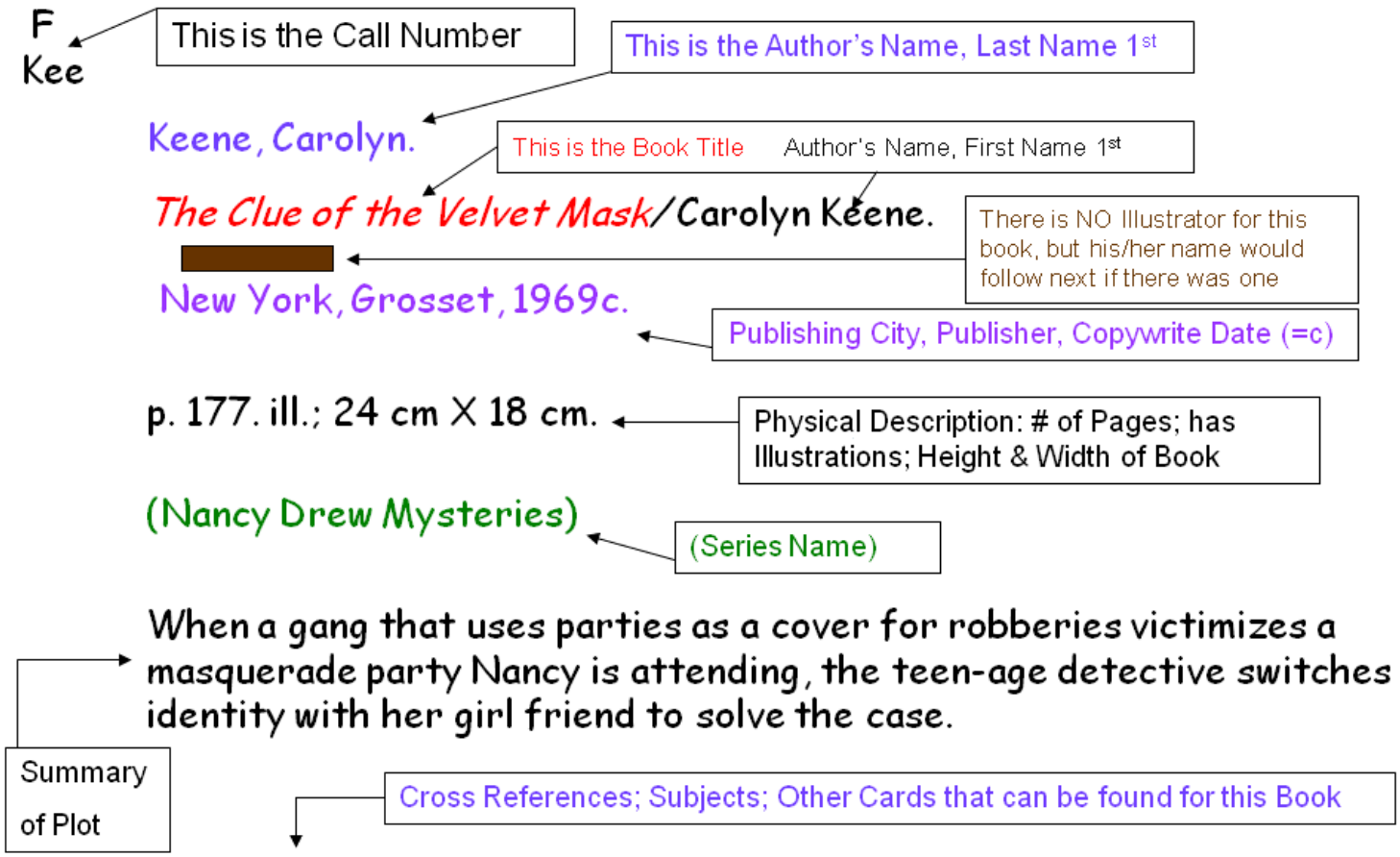
p. 177. ill.; 24 cm X 18 cm.
(Nancy Drew Mysteries)

When a gang that uses parties as a cover for robberies victimizes a masquerade party Nancy is attending, the teen-age detective switches identity with her girl friend to solve the case.

1. Mystery and detective stories. 2. Robbery/Fiction. I. Nancy Drew Mystery Series. II. Title

(source: www.graves.k12.ky.us/powerpoints/elementary/symrrobertson.ppt)

What's on a card?



(source: www.graves.k12.ky.us/powerpoints/elementary/symrrobertson.ppt)

Milestone 1: Automatic Indexing

Luhn's ideas: automatic indexing

Hans Peter Luhn
(IBM)

- Important contributions of Luhn
  - Automatic indexing (using term frequency to select terms, KWIC)
  - Automatic abstracting (summarization)
  - Measuring similarity of documents based on their indexing terms
  - Selective dissemination of information (SDI, i.e., filtering)
  - Coined the term "business intelligence"

Luhn's idea: automatic indexing based on statistical analysis of text

It is here proposed that the frequency of word occurrence in an article furnishes a useful measurement of word significance. It is further proposed that the relative position within a sentence of words having given values of significance furnish a useful measurement for determining the significance of sentences. The significance factor of a sentence will therefore be based on a combination of these two measurements. " (Luhn 58)

LUHN, H.P., 'A statistical approach to mechanised encoding and searching of library information', *IBM Journal of Research and Development,* **1**, 309-317 (1957).
LUHN, H.P., 'The automatic creation of literature abstracts', *IBM Journal of Research and Development*, **2**, 159-165 (1958).

The notion of "resolving power of a word"

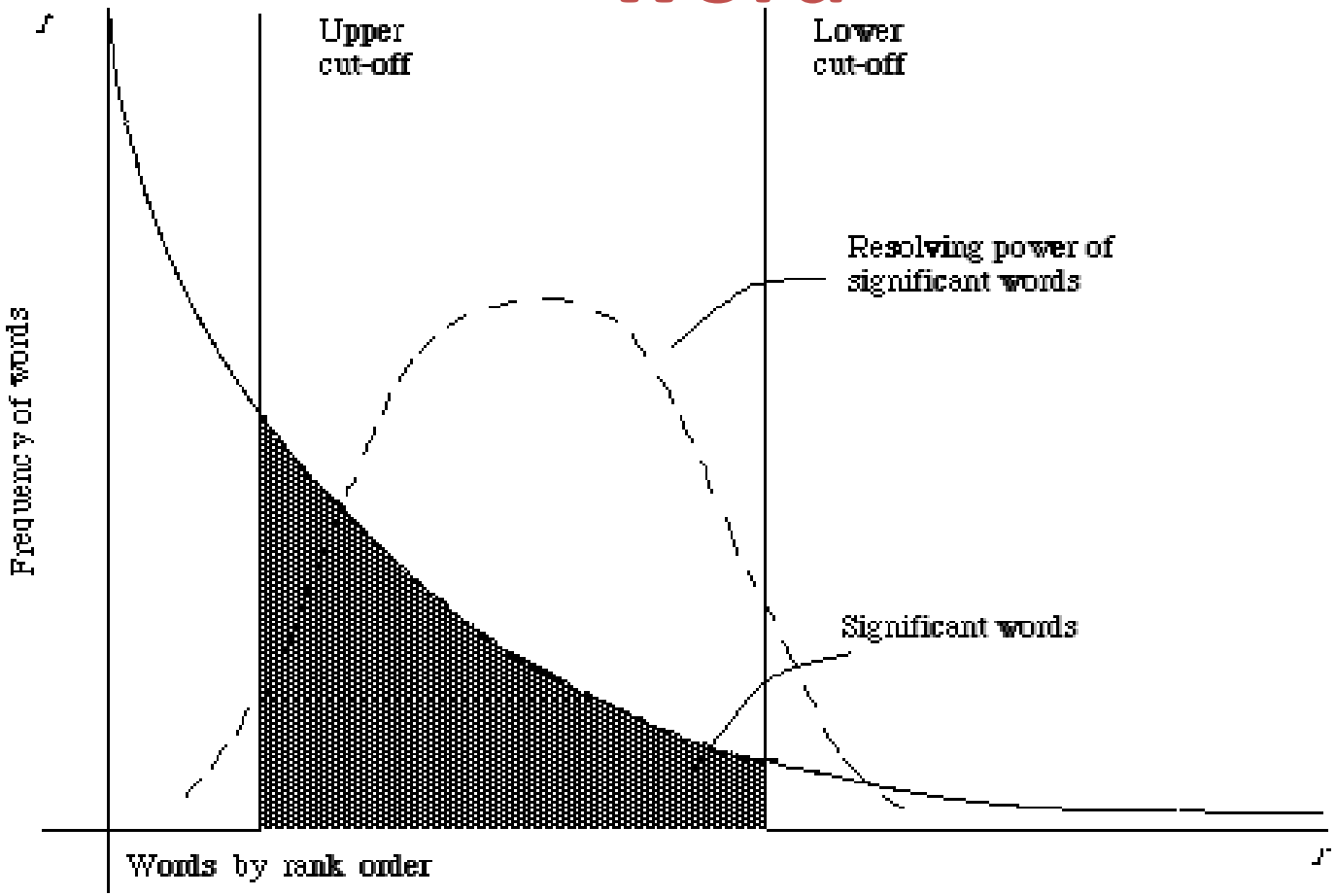


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz[44] page 120)

Probabilistic view of association and proximity

"the method to be developed here is a **probabilistic one** based on the physical properties of written texts. No consideration is to be given to the meaning of words or the arguments expressed by word combinations. Instead it is here argued that, whatever the topic, the **closer** certain words are **associated**, the more specifically an aspect of the subject is being treated. Therefore, wherever the greatest number of frequently occurring different words are found in greatest physical proximity to each other, the probability is very high that the information being conveyed is most representative of the article." (Luhn 58)

Automatic abstracting algorithm

[Luhn 58]

Portion of sentence bracketed by and including significant words not more than four non-significant words apart. If eligible, the whole sentence is cited.

Figure 2 **Computation of significance factor.** *The square of the number of bracketed significant words (4) divided by the total number of bracketed words (7) = 2.3.*

The idea of query-specific summarization

"In many instances condensations of documents are made emphasizing the relationship of the information in the document to a special interest or field of investigation. In such cases sentences could be weighted by assigning a premium value to a predetermined class of words."

Key Word in Context (KWIC)

KWIC is an acronym for **Key Word In Context**, the most common format for concordance lines. The term KWIC was first coined by Hans Peter Luhn.

KWIC is an **acronym** for Key Word In Context, ...	page 1
... Key Word In Context, the most **common** format for concordance lines.	page 1
... the most common format for **concordance** lines.	page 1
... is an acronym for Key Word In **Context**, the most common format ...	page 1
Wikipedia, The Free **Encyclopedia**	page 0
... In Context, the most common **format** for concordance lines.	page 1
Wikipedia, The **Free** Encyclopedia	page 0
KWIC is an acronym for **Key** Word In Context, the most ...	page 1
KWIC is an acronym for Key Word ...	page 1
... common format for concordance **lines**.	page 1
... for Key Word In Context, the **most** common format for concordance ...	page 1
Wikipedia, The Free Encyclopedia	page 0
KWIC is an acronym for Key **Word** In Context, the most common ...	page 1

Sorted

Probabilistic representation and similarity computation [Luhn 61]

Absolute and Relative Frequencies of Top-frequency Words Shares by at Least 2 Documents.

Word	Document A		Document B		Document C	
	abs.	rel.	abs.	rel.	abs.	rel.
Brain	12	.082	12	.109	29	.080
Experience	10	.069	7	.064	11	.030
Record	10	.069	3	.027	-	-
Area	9	.062	-	-	12	.033
Conscious	8	.055	3	.027	-	-
Patient	7	.048	8	.078	-	-
Dr. Penfield	6	.041	6	.055	-	-
Electric	6	.041	6	.055	-	-
Time	6	.041	5	.046	-	-
Hear	5	.034	9	.082	-	-
Stimulated	5	.034	4	.086	27	.074
Cortex	4	.027	-	-	26	.072
Detail	4	.027	4	.086	-	-
Function	4	.027	-	-	11	.030
Temporal	4	.027	5	.046	-	-
Respond	4	.027	-	-	11	.030

Coefficients

s (A, B) = .495
s (A, C) = .260
s (B, C) = .147

Method:

$s(X, Y) = \sum_i \min(f_i, g_i)$, where the sum is taken over all words shared by the documents X and Y. f_i is relative frequency of word number i in X and g_i is the same for Y.

An early idea about using unigram language model to represent text

What do you think about the similarity function?

Other early ideas related to indexing

- [Joyce & Needham 58]: Relevance-based ranking, vector-space model, query expansion, connection between machine translation and IR
- [Doyle 62]: Automatic discovery of term relations/clusters, "semantic road map" for both search and browsing (and text mining!)
- [Maron 61]: automatic text categorization
- [Borko 62]: categories can be automatically generated from text using factor analysis
- [Edmundson & Wyllys 61]: local-global relative frequency (kind of TF-IDF)
- Many more (e.g., citation index…)

Milestone 2:

Cranfield Evaluation Methodology

Background

- IR is an empirically defined problem, thus experiments must be designed to test whether one system is better than another
- However, early work on IR (e.g., Luhn's) mostly proposed ideas without rigorous testing
- Catalysts for experimental IR:
  - Hot debate over different languages for manual indexing
  - Automatic indexing vs. manual indexing
- How can we experimentally test an indexing method?

Cleverdon's Cranfield Tests

Cyril Cleverdon
(Cranfield Inst. of Tech, UK)

1957-1960: Cranfield I
- Comparison of indexing methods
- Controversial results (lots of criticisms)

1960-1966: Cranfield II
- More rigorous evaluation methodology
- Introduced precision & recall
- Decomposed study of each component in an indexing method
- Still lots of criticisms, but laid the foundation for evaluation that has a very long-term and broad impact

Cleverdon received the ACM SIGIR Salton Award in 1991.

Cranfield II Test: Experiment Design

- Decomposed study of contributions of different components of an indexing language
- Rigorous control of evaluation
  - Having complete judgments is more important than having a large set of documents
  - Document collection: 1400 documents (cited papers by 200 authors, no original papers by these authors)
  - Queries: 279 questions provided by authors of original papers
  - Relevance judgments:
    - Multiple levels: 1-5
    - Initially done by 6 students in 3 months; final judgments by the originators
  - Measures: precision, recall, fallout, prec-recall curve
  - Ranking method: coordination level (# matched terms)

Measures: Precision, Recall, and Fallout [Cleverdon 67]

	RELEVANT	NON-RELEVANT	
RETRIEVED	a	b	a + b
NOT RETRIEVED	c	d	c + d
	a + c	b + d	a + b + c + d = N (Total Collection)

FIGURE 1 2 x 2 CONTINGENCY TABLE

For the purpose of evaluating an information retrieval system, performance is presented by plotting the recall ratio $\left(\frac{100a}{a+c}\right)$ against either the precision ratio $\left(\frac{100a}{a+b}\right)$ or the fallout ratio $\left(\frac{100b}{b+d}\right)$. The fallout ratio is particularly useful when comparing performances of document collections of different sizes, but the precision ratio is more satisfactory for most of the results obtained in the Cranfield work.

Precision-Recall Curve [Cleverdon 67]

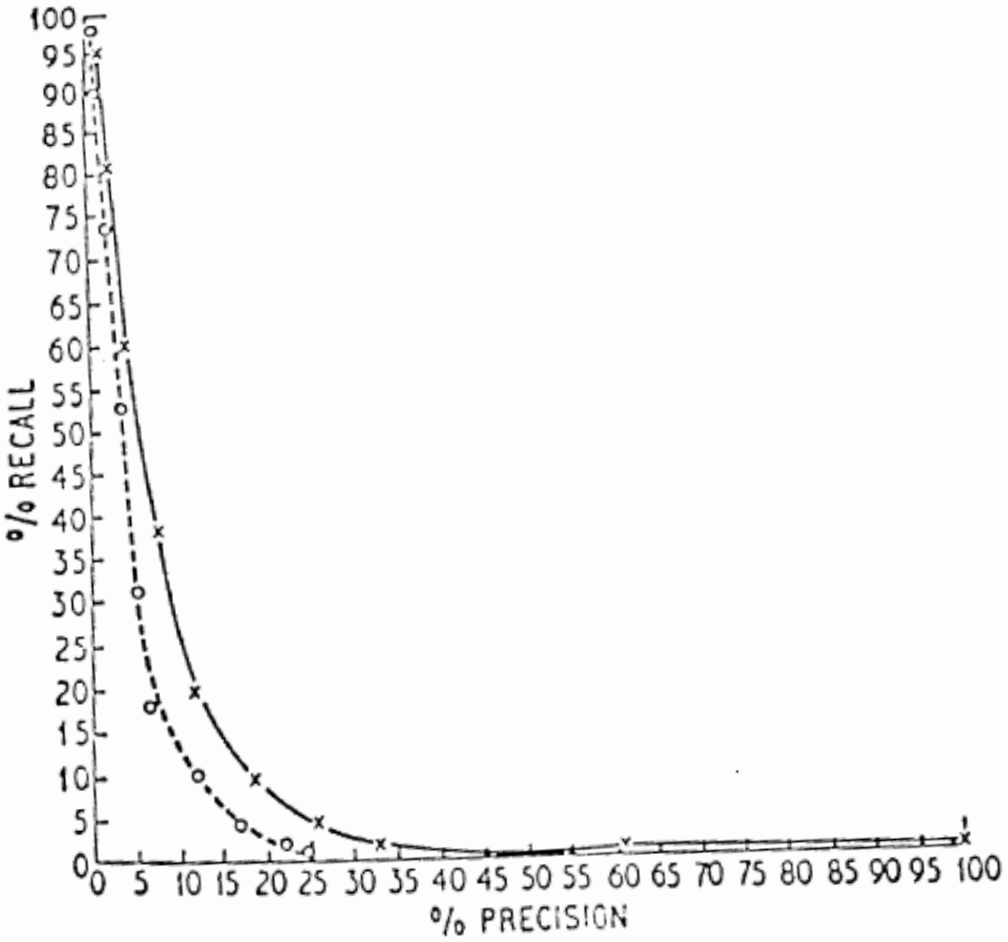


FIGURE 8 RECALL/PRECISION PLOT FOR INDEX LANGUAGES 1.1 AND 1.6 AS GIVEN IN FIGURES 6 AND 7

Cranfield II Test: Results [Cleverdon 67]

ORDER	NORMALISED RECALL	INDEXING LANGUAGE		
1	65.82	I-3	Single terms.	Word forms
2	65.23	I-2	Single terms.	Synonyms
3	65.00	I-1	Single terms.	Natural Language
4	64.47	I-6	Single terms.	Synonyms, word forms, quasi-synonyms
5	64.41	I-8	Single terms	Hierarchy second stage
6	64.05	I-7	Single terms.	Hierarchy first stage
7=	63.05	I-5	Single terms.	Synonyms. Quasi-synonyms
7=	63.05	II-11	Simple concepts.	Hierarchical and alphabetical selection
9	62.88	II-10	Simple concepts.	Alphabetical second stage selection
10=	61.76	III-1	Controlled terms.	Basic terms
10=	61.76	III-2	Controlled terms.	Narrower terms
12	61.17	I-9	Single terms.	Hierarchy third stage
13	60.94	IV-3	Abstracts.	Natural language
14	60.82	IV-4	Abstracts.	Word forms
15	60.11	III-3	Controlled terms.	Broader terms

For more information about Cranfield II test, see
Cleverdon, C.W., 1967, The Cranfield tests on index language devices. Aslib Proceedings, 19, 173-192.

Cranfield test methodology

- Specify a retrieval task
- Create a collection of sample documents
- Create a set of topics/queries appropriate for the retrieval task
- Create a set of relevance judgments (i.e., judgments about which document is relevant to which query)
- Define a set of measures
- Apply a method to (or run a system on) the collection to obtain performance figures

Milestone 3:

Smart IR System

Cranfield tests were done manually, how about doing all the tests with an automatic system?

SMART System

SMART: System for the Mechanical Analysis and Retrieval of Text

Gerard Salton
(Harvard, Cornell)

1961-1965: SMART system develop
(Gerard Salton + Michael Lesk)

-First automatic retrieval system
-Term weighting + vector similarity
-Experimented with many ideas for indexing
-Did statistical significance test
-Major findings:

+ **weighted terms help**
+ **automatic indexing is as good as manual indexing**
+ **Indexing based on abstracts outperforms titles**
+ linguistic phrases and statistical phrases are similar

About the SMART system

Developed on IBM 7094
(time-sharing system, 0.35MIPS, 32KB memory)

Early development: (1961-1965):
Michael Lesk
First UNIX implementation(v8, 1980):
Edward Fox
The widely used SMART toolkit
(v10/11, 1980-1990s)
Chris Buckley

SMART was the most popular IR toolkit (in C) widely used in 1990s by IR researchers and some machine learning researchers

Features of SMART system

A. Basic Organization. The SMART system is a fully automatic document retrieval system operating on the IBM 7094. The system does not rely on manually assigned keywords or index terms for the identification of documents and search requests, nor does it use primarily the frequency of occurrence of certain words or phrases included in the document texts. Instead, the system goes beyond simple word-matching procedures by using a variety of intellectual aids in the form of synonym dictionaries, hierarchical arrangements of subject identifiers, phrase generating methods, and the like, in order to obtain the content identifications useful for the retrieval process.

The following facilities incorporated into the SMART system for purposes of document analysis are of principal interest:

(a) a system for separating English words into stems and affixes, which can be used to reduce incoming texts into *word stem* form;

(b) a synonym dictionary, or thesaurus, used to replace significant word stems by *concept numbers*, each concept representing a class of related word stems;

(c) a *hierarchical arrangement* of the concepts included in the thesaurus which makes it possible, given any concept number, to find its "parent" in the hierarchy, its "sons," its "brothers," and any of a set of possible cross-references;

(d) *statistical association* methods, used to compute similarity coefficients between words, word stems, or concepts, based on co-occurrence patterns between these entities in the sentences of a document, or in the documents of a collection; associated items can then serve as content identifiers in addition to the original ones;

(e) *syntactic analysis* methods, which are used to generate phrases consisting of several words or concepts; each phrase serves as an indicator of document content, provided certain prespecified syntactic relations obtain between the phrase components;

SMART Features (cont.)

(f) *statistical phrase recognition* methods, which operate like the preceding syntactic procedures by using a preconstructed phrase dictionary, except that no test is made to ensure that the syntactic relationships between phrase components are satisfied;

(g) *request-document matching* procedures, which make it possible to use a variety of different correlation methods to compare analyzed documents with analyzed requests, including concept weight adjustments and variations in the length of the document texts being analyzed.

Stored documents and search requests are processed by the system *without any prior manual analysis* using one of several hundred automatic content analysis methods, and those documents which most nearly match a given search request are identified. Specifically, a correlation coefficient is computed to indicate the degree of similarity between each document and each search request, and documents are then ranked in decreasing order of the correlation coefficient [4–6]. A cutoff can then be picked, and documents above the chosen cutoff can be withdrawn from the file and turned over to the user as answers to the search request.

The search process may be controlled by the user in that a request can be processed first in a standard mode. After analysis of the output produced, feedback information can then be returned to the system where it is used to reprocess the request under altered conditions. The new output can again be examined, and the search can be iterated until the right kind and amount of information are obtained [7, 8].

Overall Results

TABLE IX. Overall Merit for Eight Processing Methods Used With Three Document Collections

M: merit measure (normalized recall plus normalized precision);
D: dictionary used (D1: suffix "s," D2: word stem, D3: thesaurus, D4: statistical phrase, D5: word-word association)

Order	*IRE-3*			*CRAN-1*			*ADI*		
	D	*Method*	*M*	*D*	*Method*	*M*	*D*	*Method*	*M*
1	D4	Stat. phrase	1.686	D3	Thesaurus	1.579	D4	Stat. phrase	1.45
2	D3	Thesaurus	1.665	D1	Suffix "s"	1.574	D3	Thesaurus	1.44
3	D2	Stems	1.570	D4	Stat. phrase	1.566	D5	Concon	1.36
4	D5	Concon	1.559	D5	Concon	1.556	D2	Stems	1.33
5	D1	Suffix "s"	1.530	D2	Stems	1.534	D2	No weights	1.29
6	D2	No weights	1.494	D2	No weights	1.477	D2	Title only	1.29
7	D2	Overlap	1.455	D2	Title only	1.430	D1	Suffix "s"	1.28
8	D3	Title only	1.369	D2	Overlap	1.407	D2	Overlap	1.24
Range	0.317			0.172			0.215		

Title only, overlap similarity, and no weights are clearly the worst

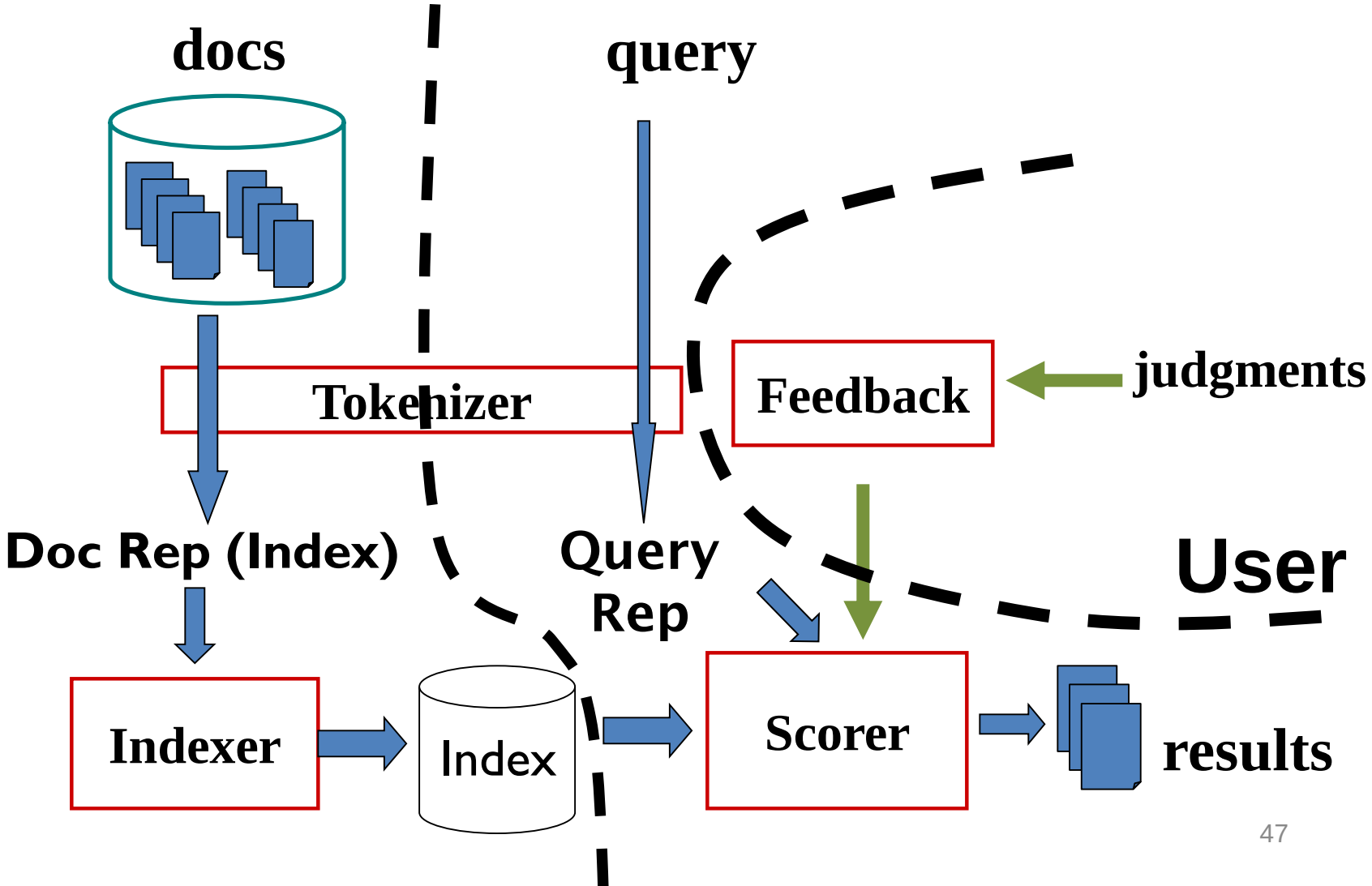
Key Findings

- Term weighting is very useful (better than binary values)
- Cosine similarity is better than the overlap similarity measure
- Using abstracts for indexing is better than using titles only
- Synonyms are helpful
- Automatic indexing may be as effective as manual indexing

Lecture Plan

- What is text retrieval (TR) ?
- Document selection vs. document ranking
- Major research milestones
- Components in a TR system

Typical TR System Architecture



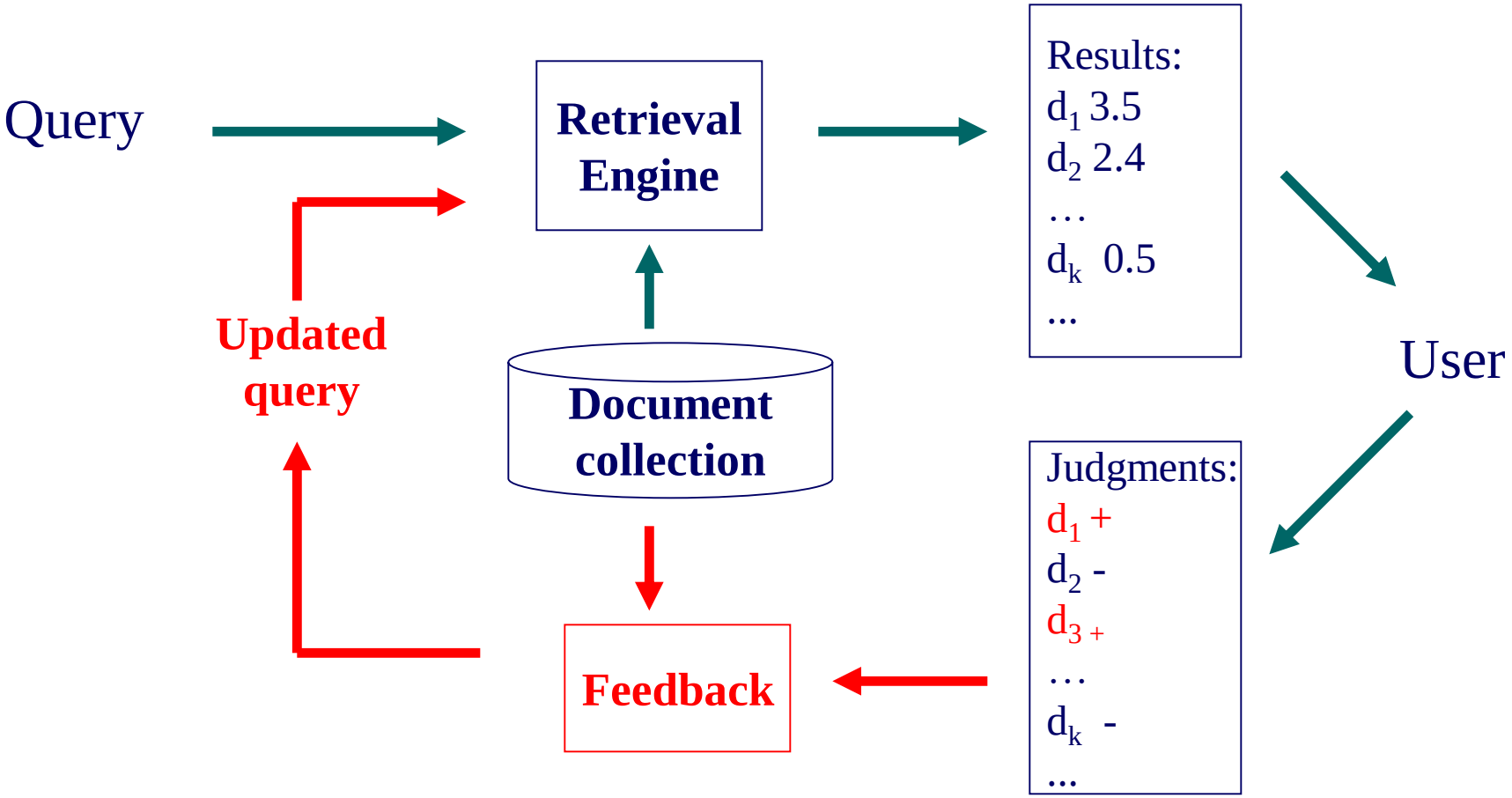
Text Representation/Indexing

- Making it easier to match a query with a document
- Query and document should be represented using the same units/terms
- Controlled vocabulary vs. full text indexing
- Full-text indexing is more practically useful and has proven to be as effective as manual indexing with controlled vocabulary

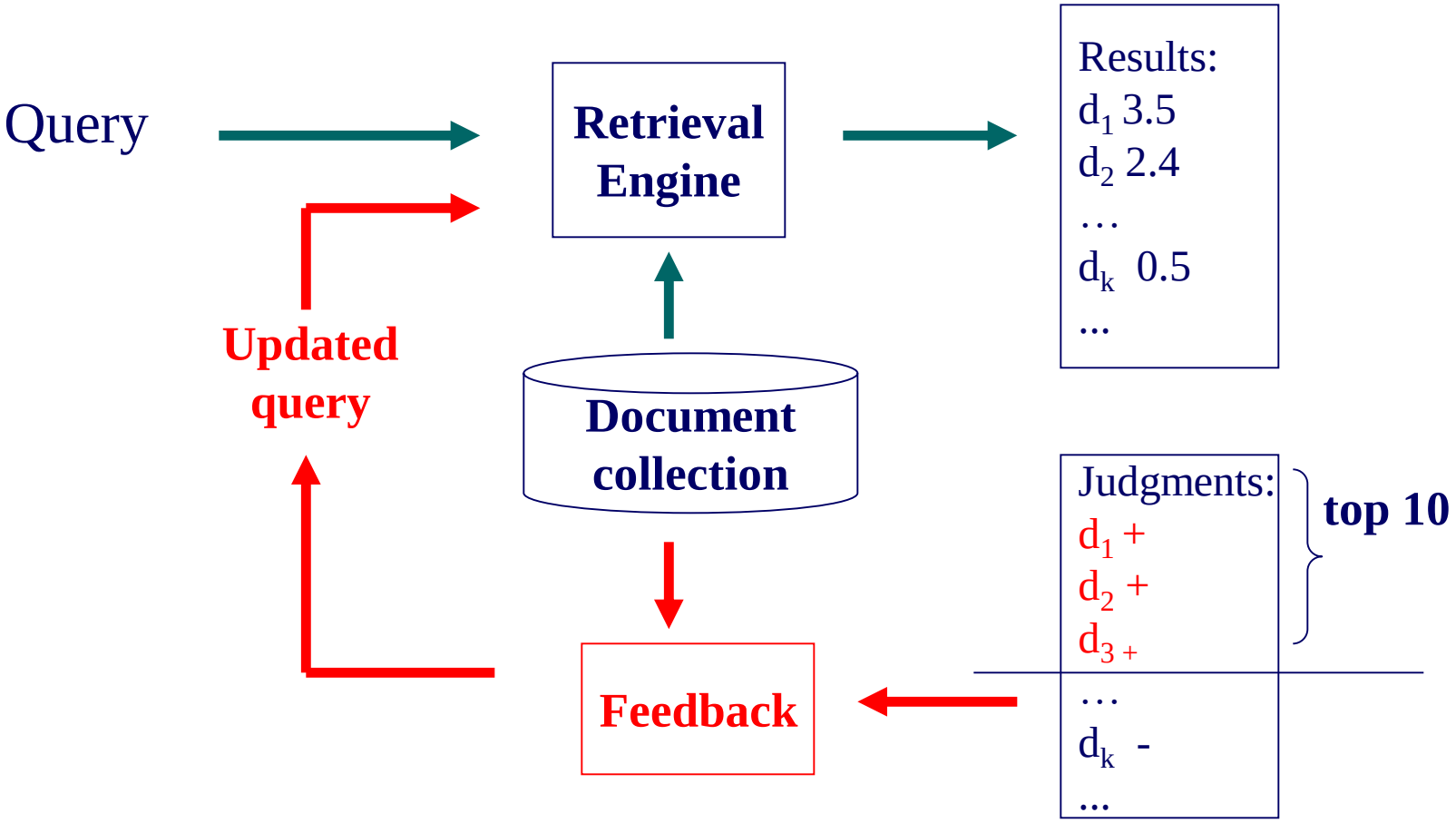
Tokenization

- Normalize lexical units: Words with similar meanings should be mapped to the same indexing term
- Stemming: Mapping all inflectional forms of words to the same root form, e.g.
  - computer -> compute
  - computation -> compute
  - computing -> compute (but king->k?)
- Porter's Stemmer is popular for English

Relevance Feedback



Pseudo/Blind/Automatic Feedback



Questions?