Universidad de Granada

SECABA LAB
Quality Evaluation & Information Retrieval

Sistemas de recuperación de información y de recomendación:

Information Retrieval Systems (IRS)

Marzo, 2020

Views of the Problem of IR

- 1960-70's: ***finding material*** (usually documents) of an ***unstructured*** nature (usually text) that satisfies an ***information need*** from a ***small corpora*** of documents.

- 1980's: finding documents that satisfies an information need from within ***large collections*** (usually stored on computer).

- 1990's: ***searching for resources*** (pages, web sites, movies, pdf files, ect.) on the ***World Wide Web*** is the most recent "killer application".

IR=Web search

Web Challenges for IR

- Distributed Data: Documents spread over millions of different web servers.
- Volatile Data: Many documents change or disappear rapidly (e.g. dead links).
- Large Volume: Billions of separate documents.
- Unstructured and Redundant Data: No uniform structure, HTML errors, up to 30% (near) duplicate documents.
- Quality of Data: No editorial control, false information, poor quality writing, typos, etc.
- Heterogeneous Data: Multiple media types (images, video, VRML), languages, character sets, etc.

IR Activity

It is concerned

1. FIRSTLY, with retrieving ***relevant*** documents to a query, and,

2. SECONDLY, with retrieving from ***large*** sets of documents ***efficiently***.

IR Task

- Given:
  - A corpus of textual natural-language documents.
  - A user query in the form of a textual string.
- Find:
  - A ranked set of documents that are relevant to the query.

Relevance Concept

Relevance is a subjective judgment and may include:

- Satisfying the goals of the user and his/her intended use of the information (*information need*). (Traditional meaning)
- Personalized (recommender systems)
- Being on the proper subject. (quality)
- Being timely (recent information). (quality)
- Being authoritative (from a trusted source). (quality)

Relevance Concept

Conceptual Evolution of Relevance

Information Properties

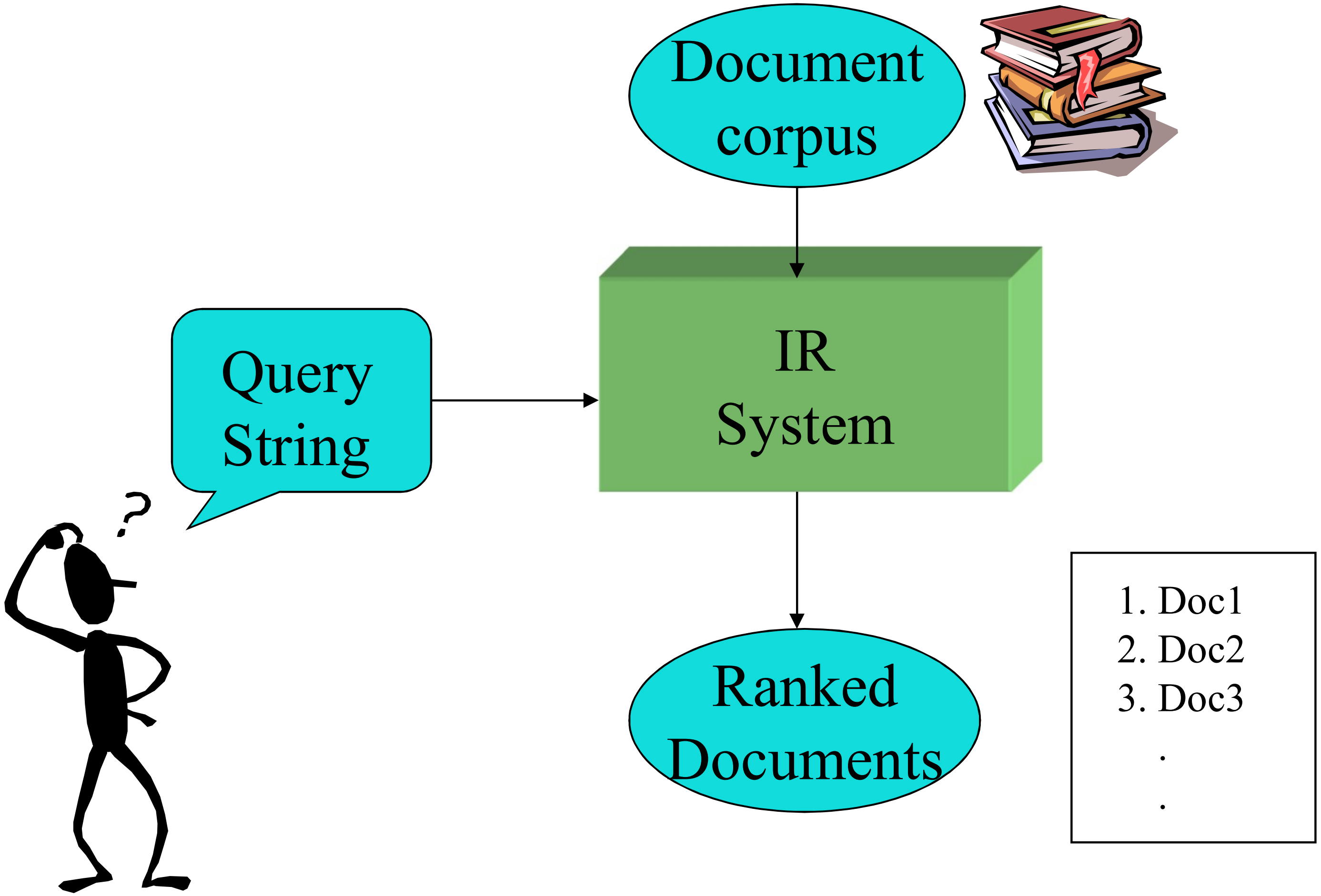
↓

User Properties

↓

<u>**QUALITY**</u>

Solution: IR System



Document corpus

Query String

IR System

Ranked Documents

1. Doc1
2. Doc2
3. Doc3

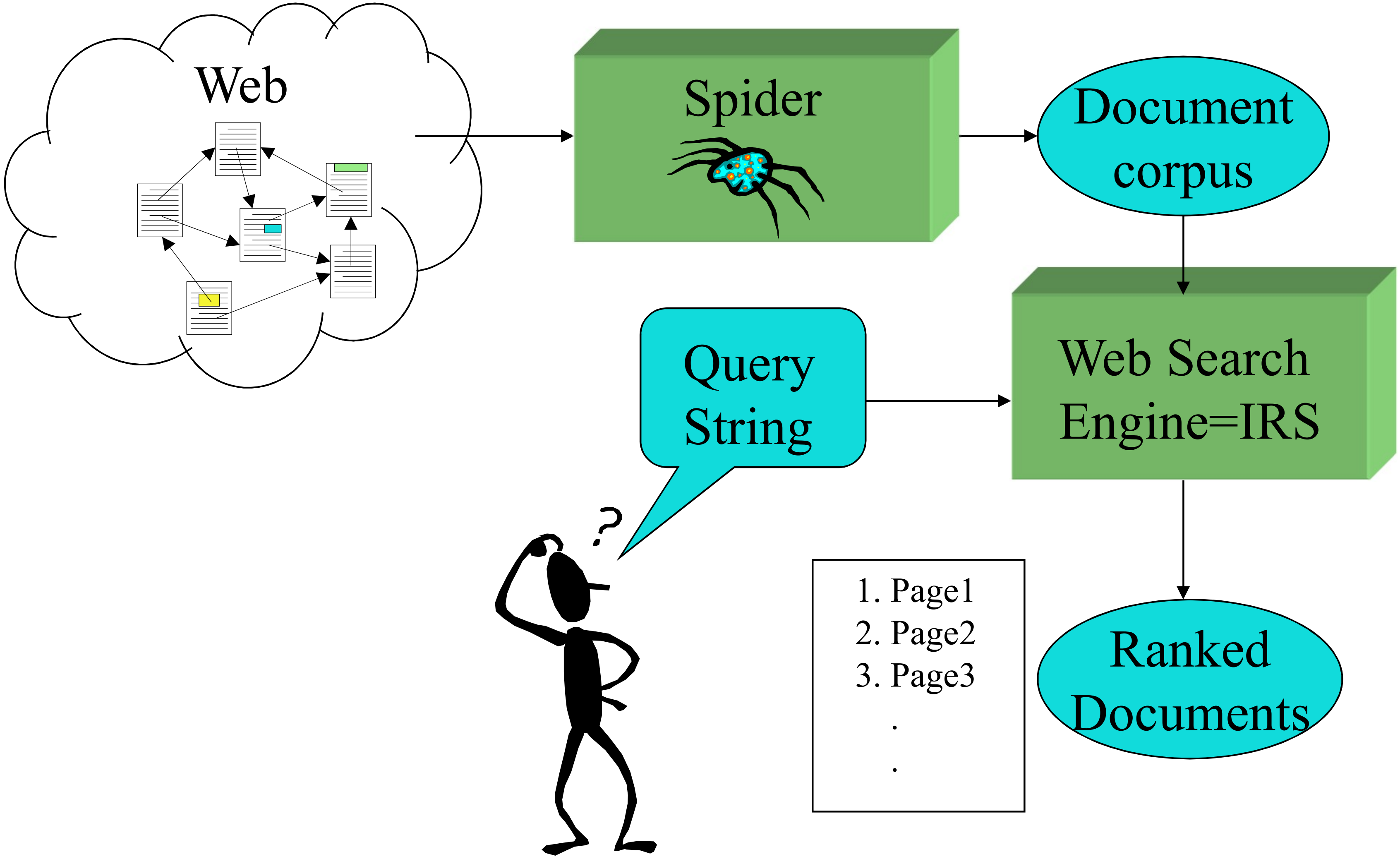
.

.

Web Solution: Web Search Engine

- Application of IR to web documents on the World Wide Web.
- Important Differences:
  - Must assemble document corpus by spidering the Web.
  - Can exploit the structural layout information in HTML (XML).
  - Documents change uncontrollably and growing quickly
  - Can exploit the link structure of the Web (Google).
  - Heterogeneous information
  - No quality control

Web Solution: Web Search Engine



Definition of IRS

- IRS: An automatic system ***to store*** documents in order ***to retrieve*** them later in response to user queries.

- Tasks to develop: to represent documents, to represent user queries, to evaluate relevance of documents, to rank documents, to improve IR activity (query expansion, relevance feedback),….

IRS vs DBMS

IRS	Relational DBMS
Probabilistic retrieval (Partial match)	Deterministic Retrieval (Exact match)
Un-Structured Information (free text)	Structured Information (relational table)

IRS vs DBMS

- Structured data tends to refer to information in "tables"

Employee	Manager	Salary
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

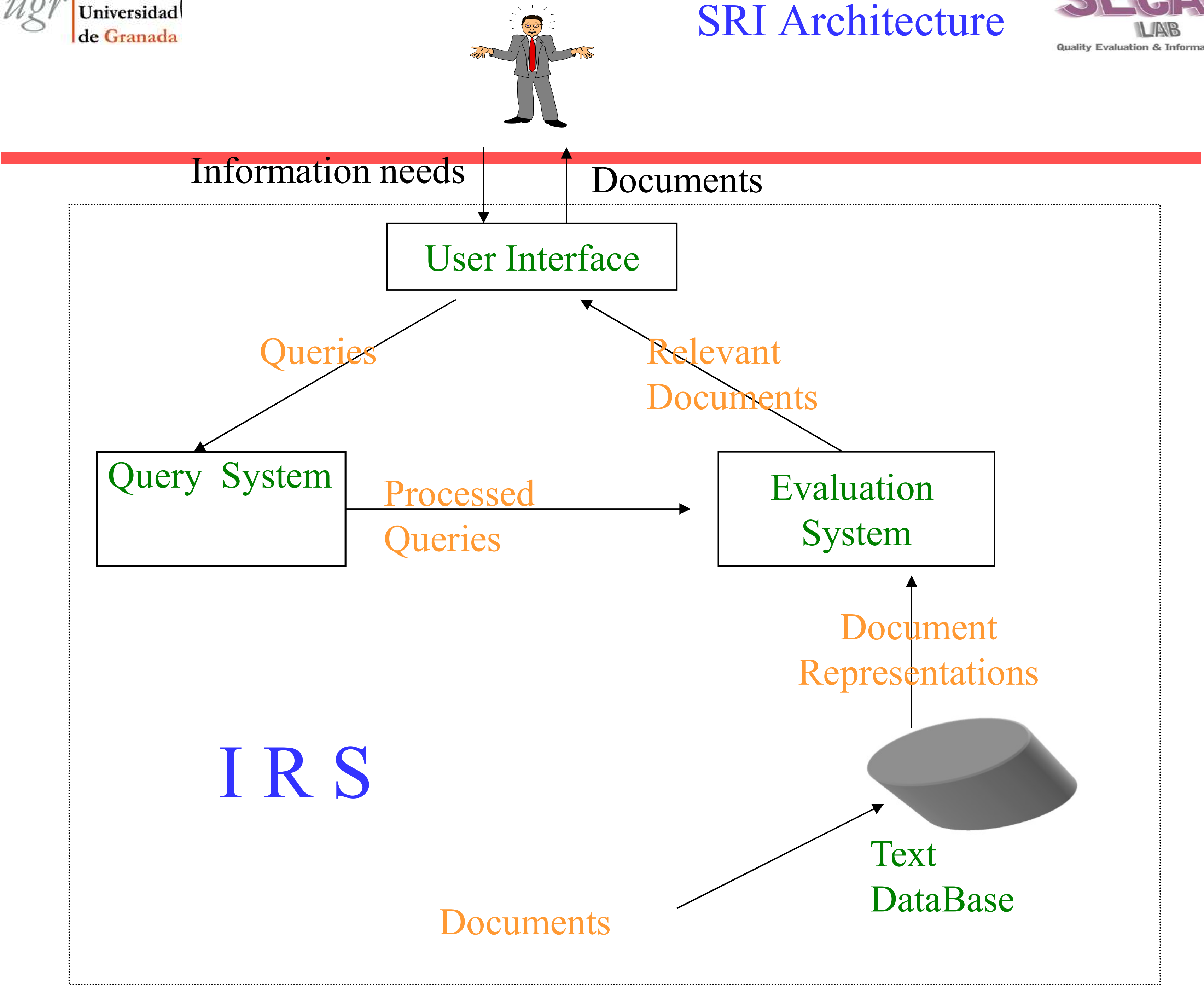
- Typically allows numerical range and exact match (for text) queries, e.g.,

Salary < 60000 AND Manager = Smith.

IRS Components

- Text Database stores the documents and their index based representations
- Query System defines the query language to represent the user information needs.
- Evaluation System evaluates user queries to retrieve the relevant documents
- User Interface manages user-IRS interaction: query input and document output.

SRI Architecture



Text Database: Indexing

- In practice, documents are not represented directly in a complete way.
- Surrogate: We use a surrogate of the documents to represent them in the databases:

Loss of information

- Documents are represented by means of the

INDEXING

- Indexing is based on the content of documents: defines each surrogate of a document as a set of index terms which are obtained according to the analysis of its content

Text Database: Indexing

- Goal of indexing: to predict the relevance of the documents with respect to future information needs, and in such a way to improve the retrieval of documents.

- Two indexing tools:
  - indexing by hand
  - automatic indexing.

Text Database: Indexing

- Problems of indexing by hand:
  - loss of consistency between users and expert indexers,
  - different levels of specificity,
  - dealing with big collections of documents, ect,.

Automatic Indexing

- Popular: Automatic indexing is the most important and applied, specially to deal with big collections of documents.
- Term frecuencies: Automatic indexing is based on the term frecuency, i.e, the computation of the number of times that a term occurs in a document.
- Using indexing:

A document = vector of weighted index terms

Automatic Indexing

- Weights of the index terms: The weight associated with each index term measures the description degree of the content of the document by the term or the relevance degree of the document with respect to term.
- Two Questions:

How to identify good index terms?

How to compute weights of the index terms?

Automatic Indexing: Choice of index terms

- Choice of index terms is based on the

  Representation Power of Terms

- Representation power (rp): To evaluate if a term is suitable or not to be index term

- Zifp’s Law: Rp is based on the Zif’s Law applied on the term frecuencies

Automatic Indexing: Zifp's law

- Zifp's laws: In his book

Human Behavior and the Principle of Least Effort,

Zipf argues that he has found a unifying principle, the Principle of the Least Effort, which underlies essentially the entire human condition. The Principle argues that people will act so as to minimize their probable average rate of work.

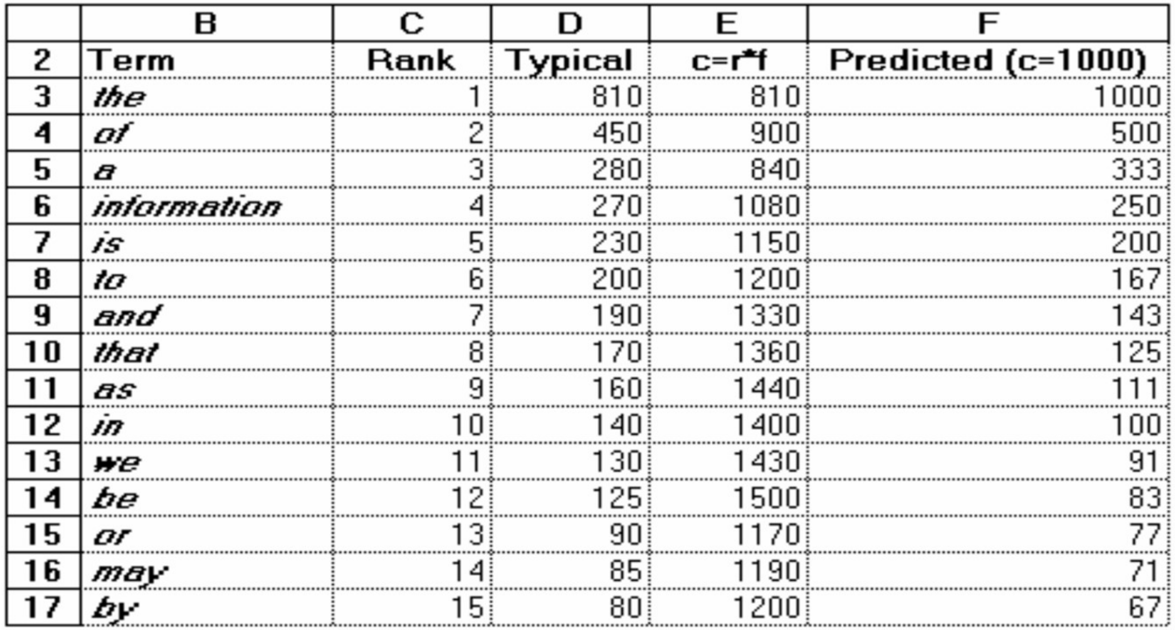
Automatic Indexing: Zifp’s Law

- Zifp’s Law applied in natural language: It is an empirical law that states that given some corpus of natural language, the frequency of any word is inversely proportional to its rank in the frequency table.

$$R \times F = C$$

  - R is the rank of the term
  - F is the frequency of the term
  - C is a constant

Automatic Indexing: Zifp’s Law



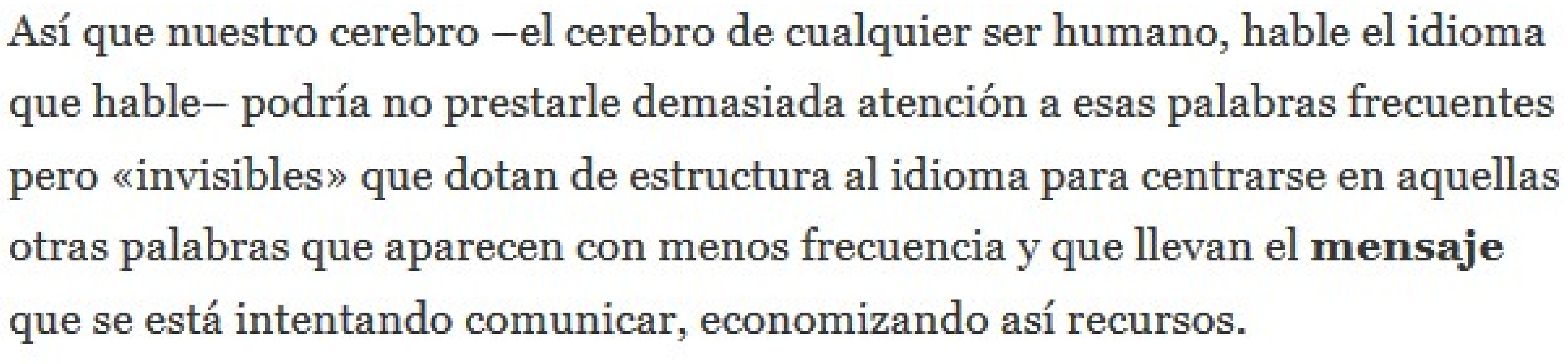
	B	C	D	E	F
2	Term	Rank	Typical	c=r*f	Predicted (c=1000)
3	*the*	1	810	810	1000
4	*of*	2	450	900	500
5	*a*	3	280	840	333
6	*information*	4	270	1080	250
7	*is*	5	230	1150	200
8	*to*	6	200	1200	167
9	*and*	7	190	1330	143
10	*that*	8	170	1360	125
11	*as*	9	160	1440	111
12	*in*	10	140	1400	100
13	*we*	11	130	1430	91
14	*be*	12	125	1500	83
15	*or*	13	90	1170	77
16	*may*	14	85	1190	71
17	*by*	15	80	1200	67

- This law says that the 50th most common word should occur with three times the frequency of the 150th most common word

Automatic Indexing: Zifp’s Law

Zipf’s law is useful as a rough description of the frequency distribution of words in human languages:

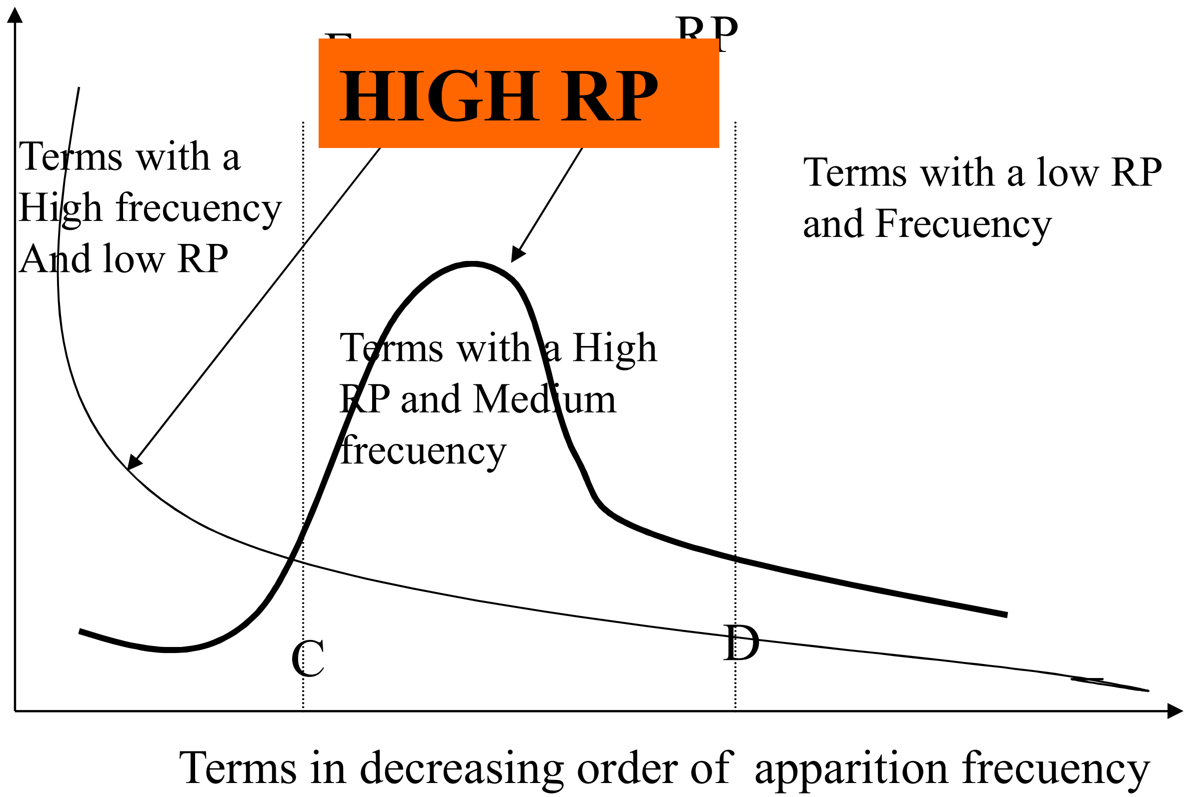
There are a few very common words (usually function words), a middling number of medium frequency words and many low frequency words.



Así que nuestro cerebro –el cerebro de cualquier ser humano, hable el idioma que hable– podría no prestarle demasiada atención a esas palabras frecuentes pero «invisibles» que dotan de estructura al idioma para centrarse en aquellas otras palabras que aparecen con menos frecuencia y que llevan el **mensaje** que se está intentando comunicar, economizando así recursos.

Automatic Indexing: Zifp’s Law vs RP

RP depends on the total apparition frecuency of the terms in a corpus of natural language



HIGH RP

Terms with a High frecuency And low RP

Terms with a High RP and Medium frecuency

Terms with a low RP and Frecuency

Automatic Indexing: Choice rules of index terms

- Question:

How to identify the points C and D in a collection of documents, i.e.,

Medium frecuency terms

which present high RP?

- Solution:

To apply some choice rules of index terms

Automatic Indexing: Choice rules of index terms

- R1: To apply stoplist or list of stopwords over the collection of documents:
  - to *exclude* high-frequency words (e.g. function words: “a”, “the”, “in”, “to”; pronouns: “I”, “he”, “she”, “it”).
  - Stopwords are language dependent.
- R2: Stemming to reduce tokens to “root” form of words to recognize morphological variation.
  - “computer”, “computational”, “computation” all reduced to same token “compute”

Automatic Indexing: Choice rules of index terms

- R3: To detect term phrases: many terms have more meaning together than alone
  - Cohesion degree between terms in phrases:

    $$cij=fij/(fi * fj)$$

  - Example: the word “system”.
- R4: To use synonymous thesaurus

Automatic Indexing: Compute the weights of index terms

- The term weights in a document should depend on their document frecuencies: More frequent terms in a document are more important, i.e. more indicative of the topic.

$$tf_{ij} = \text{frequency of term } i \text{ in document } j$$

- The term weights should depend on their document frecuency, i.e., their global apparition in the collections of documents, i.e., **the discrimination power**: Terms that appear in many *different* documents are *less* indicative of overall topic

Automatic Indexing: Inverse Document Frecuency (IDF)

- The inverse document frecuency measures the discrimination power of a term:

df_i = document frequency of term i

= number of documents containing term i

idf_i = inverse document frequency of term i,

$= \log_2 (N/ df_i)$

(N: total number of documents)

Automatic Indexing: TF-IDF Weighting

- A typical combined term importance indicator is *tf-idf weighting*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, *tf-idf* has been found to work well.
- *tf-idf* :
  - Increases with the number of occurrences within a document
  - Increases with the rarity of the term in the collection

Computing TF-IDF -- An Example

- Given a document containing terms with given frequencies:

  A(3), B(2), C(1)

- Assume collection contains 10,000 documents and document frequencies of these terms are:

  A(50), B(1300), C(250)

- Then:

A: tf = 3/3; idf = $\log_2(10000/50)$ = 7.6; tf-idf = 7.6

B: tf = 2/3; idf = $\log_2(10000/1300)$ = 2.9; tf-idf = 2.0

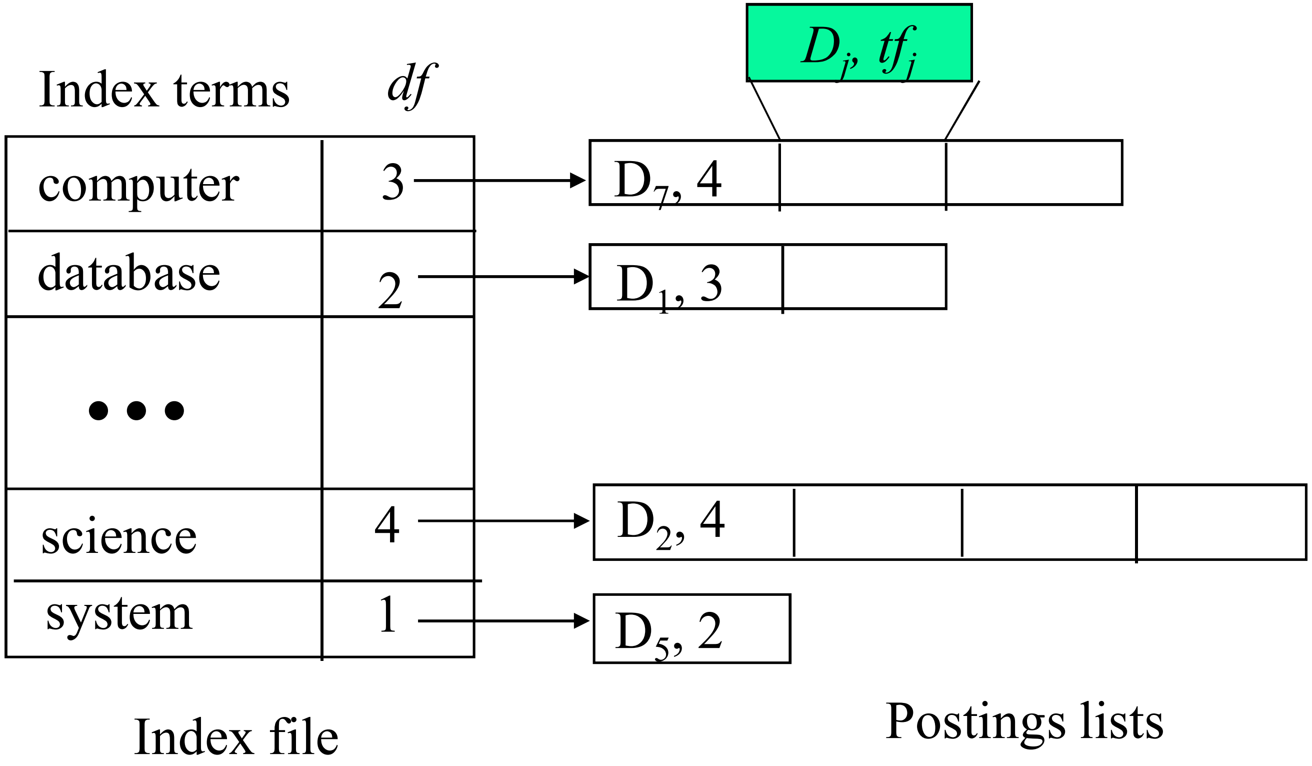
C: tf = 1/3; idf = $\log_2(10000/250)$ = 5.3; tf-idf = 1.8

Automatic Indexing: Inverted File

- Each document is represented by a vector of index terms weighted with weights in [0,1] (if we normalize them)
- In practice, document vectors are not stored directly; an inverted organization provides much better efficiency, i.e,

INVERTED FILE

Automatic Indexing: Inverted File



As the term weight is TF * IDF, therefore, we must wait until IDF are known (and therefore until all documents are indexed) before to determine them.

Automatic Indexing: Term-Document Matrix

- A collection of *n* documents can be represented in by a term-document matrix.
- An entry in the matrix corresponds to the “weight” of a term in the document; zero means the term has no significance in the document or it simply doesn’t exist in the document.

$$\begin{pmatrix} & T_1 & T_2 & \ldots & T_t \\ D_1 & w_{11} & w_{21} & \ldots & w_{t1} \\ D_2 & w_{12} & w_{22} & \ldots & w_{t2} \\ : & : & : & & : \\ : & : & : & & : \\ D_n & w_{1n} & w_{2n} & \ldots & w_{tn} \end{pmatrix}$$

IRS Models

- A IRS model specifies the details of:
  1. Text Database: Document representation
  2. Query System: Query representation
  3. Evaluation System: Retrieval function and relevance scores (binary or continuous)

The definition of the Text Database based on inverted file and tf-idf weighting is the most existing important and common one to all IRS

IRS Models:Common Preprocessing Steps

- Strip unwanted characters/markup (e.g. HTML tags, punctuation, numbers, etc.).
- Break into tokens (keywords) on whitespace.
- Stem tokens to “root” words
  - computational → comput
- Remove common stopwords (e.g. a, the, it, etc.).
- Detect common phrases (possibly using a domain specific dictionary).
- Build inverted index (keyword → list of docs containing it).

IRS Models

- Boolean model
- Vector Space model
- Probabilistic model
- Fuzzy model

Boolean IRS Model: Query System

- Traditionally: A document is represented as a set of keywords.
- Boolean Queries: Boolean expressions of keywords, connected by AND, OR, and NOT, including the use of brackets to indicate scope.
  - [[Rio & Brazil] | [Hilo & Hawaii]] & hotel & !Hilton]
- Weighted terms in queries are not allowed
- Boolean language is not easy for all users

Boolean IRS Model: Query System

- Posible queries:
  - q=term (atomic query)
  - ~q (negated query)
  - q & q (conjunctive query)
  - q | q (disjunctive query)
- Weighted terms in queries are not allowed
- Boolean language is not easy for all users

Boolean IRS Model: Evaluation System

- Retrieval Function: total matching functions.
- Output: Document is relevant or not.
- Normal form: Before to evaluate a Boolean query it is transformed in a normal form (DNF or CNF) to facilitate the evaluation of queries.
- AND= Intersection of sets
- OR= Join of sets
- NOT= Complementary of a set

Boolean IRS Model: Evaluation System

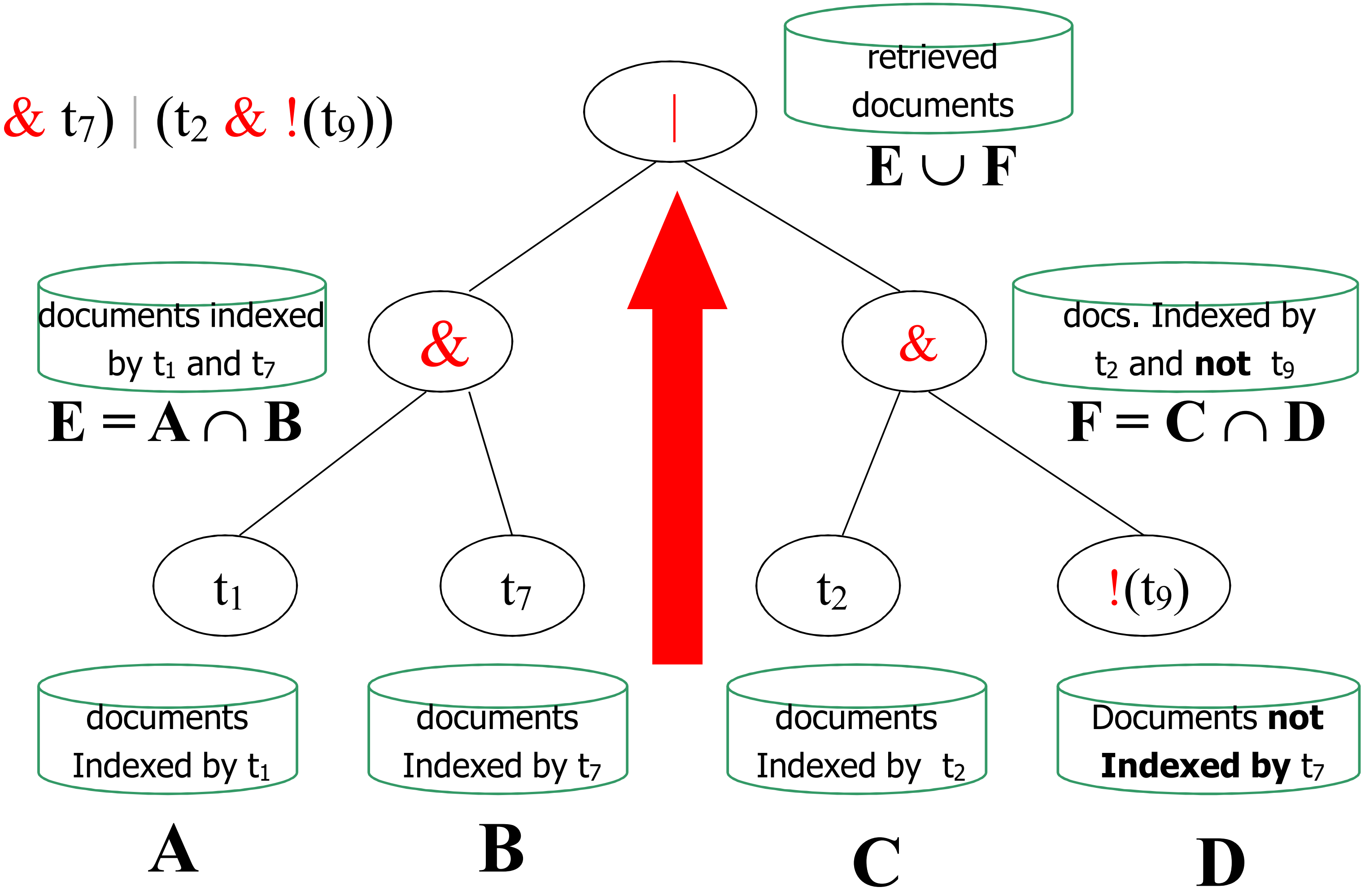
- Bottom-up evaluation procedure:

1. Firstly to evaluate the atoms of queries,

2. Secondly, the documents are evaluated according to their relevance to Boolean combinations of atomic components, and so on, working in a bottom-up method until the whole query is processed.

EXAMPLE

$(t_1 \,\&\, t_7) \,|\, (t_2 \,\&\, !(t_9))$


|
retrieved documents
E ∪ F
&
documents indexed by t_1 and t_7
E = A ∩ B
&
docs. Indexed by t_2 and **not** t_9
F = C ∩ D
t_1
t_7
t_2
$!(t_9)$
documents Indexed by t_1
documents Indexed by t_7
documents Indexed by t_2
Documents **not Indexed by** t_7
A
B
C
D

Boolean IRS Model: Problems

- Very rigid: AND means all; OR means any.
- Difficult to express complex user requests.
- Difficult to control the number of documents retrieved.
  - *All* matched documents will be returned.
- Difficult to rank output.
  - *All* matched documents logically satisfy the query.
- Difficult to perform relevance feedback.
  - If a document is identified by the user as relevant or irrelevant, how should the query be modified?

Vector-Space IRS Model: Query System

- If we have *t* terms to index the documents then we work with a vector space with dimension *t*
- Key idea: Do the same for queries: represent them as vectors in the space of dimension *t*
- Both documents and queries are expressed as *t*-dimensional vectors:

$$d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$$

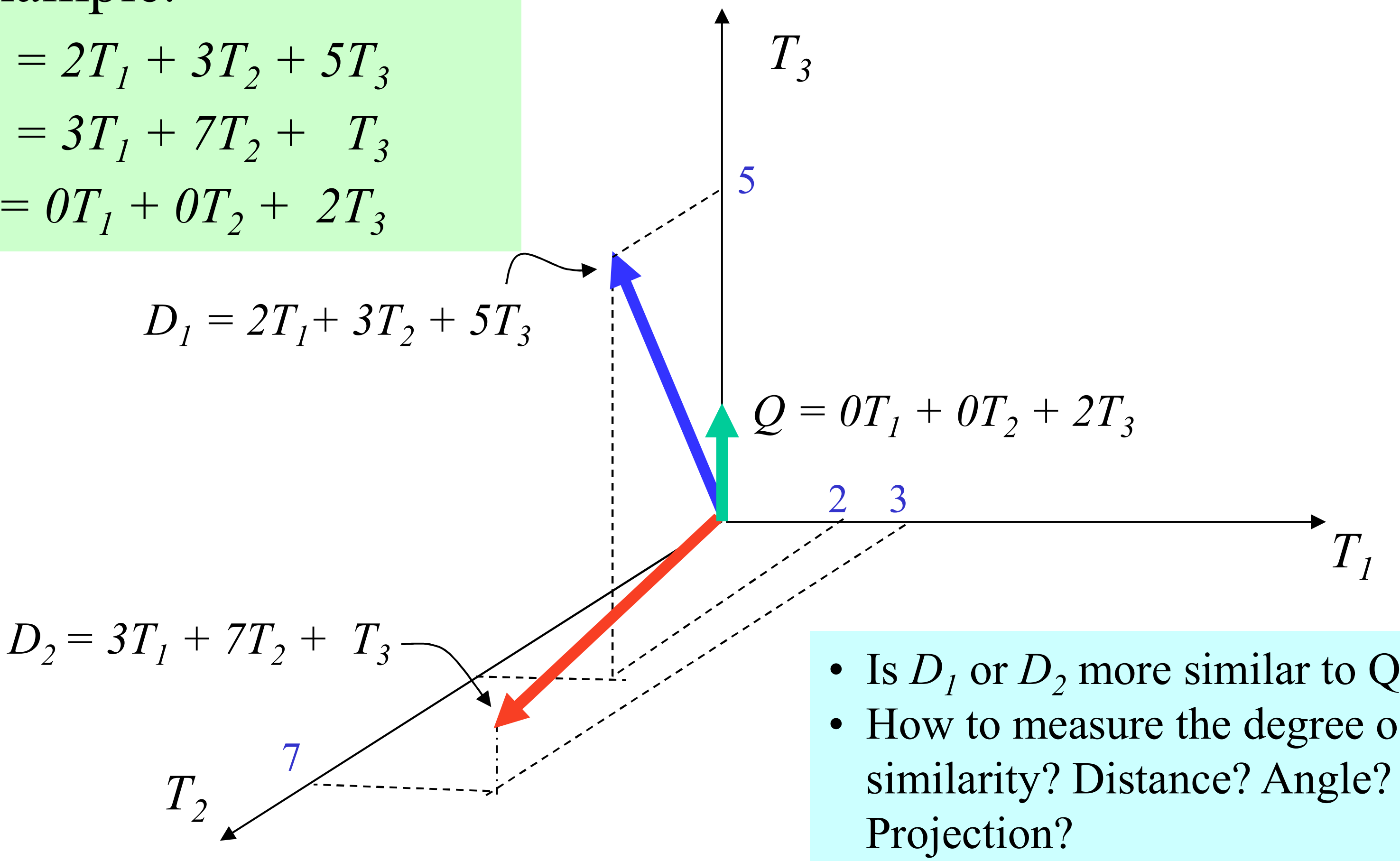
Vector-Space IRS: Graphic Representation

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$



- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

Vector-Space IRS Model: Evaluation System

- Key idea: Rank documents according to their proximity to the query in the vector space
- Retrieval function is based on a similarity or proximity of vectors
- A similarity measure is a function that computes the *degree of similarity* between vectors.
- Using a similarity degrees:
  - It is possible to rank the retrieved documents.
  - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Vector-Space IRS Model: Similarity Measures

1. INNER (DOT or SCALAR) PRODUCT

$$\mathrm{sim}(\boldsymbol{d}_j,\boldsymbol{q}) = \boldsymbol{d}_{\mathrm{j}}\bullet\boldsymbol{q} = \sum_{i=1}^{t} w_{ij}w_{iq}$$

where w_{ij} is the weight of term *i* in document *j* and w_{iq} is the weight of term *i* in the query

2. ANGLE MEASURES: cosine measure

$$\mathrm{CosSim}(\boldsymbol{d}_j,\boldsymbol{q}) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t}(w_{ij}\cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

3. DISTANCE MEASURES: Euclidean distance (p=2)

$$SIM(d,q) = \left[\sum_i \left|t_i - q_i\right|^p\right]^{\frac{1}{p}}$$

Inner Product -- Examples

Binary:

	retrieval	database	architecture	computer	text	management	information
– D =	1,	1,	1,	0,	1,	1,	0
– Q =	1,	0 ,	1,	0,	0,	1,	1

Size of vector = size of vocabulary = 7

0 means corresponding term not found in document or query

sim(D, Q) = 3

Weighted:

$D_1 = 2T_1 + 3T_2 + 5T_3 \qquad D_2 = 3T_1 + 7T_2 + 1T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

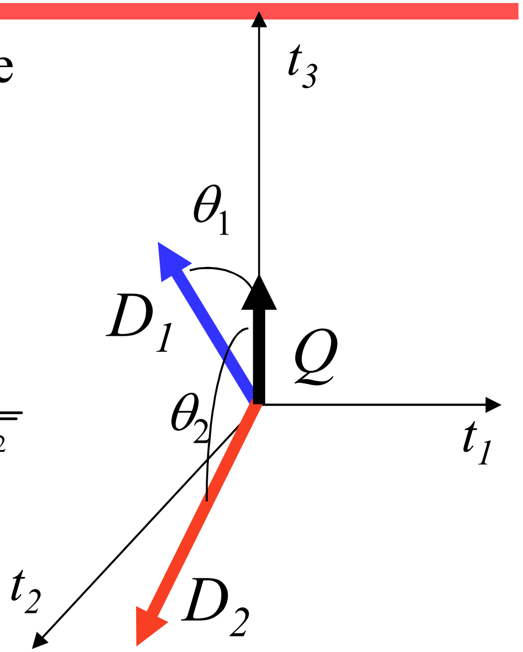
sim(D_1 , Q) = 2*0 + 3*0 + 5*2 = 10

sim(D_2 , Q) = 3*0 + 7*0 + 1*2 = 2

Cosine Similarity Measure

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.

$$\text{CosSim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\vec{d}_j \bullet \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$



$D_1 = 2T_1 + 3T_2 + 5T_3$ CosSim(D_1 , Q) = 10 / $\sqrt{(4+9+25)(0+0+4)}$ = 0.81

$D_2 = 3T_1 + 7T_2 + 1T_3$ CosSim(D_2 , Q) = 2 / $\sqrt{(9+49+1)(0+0+4)}$ = 0.13

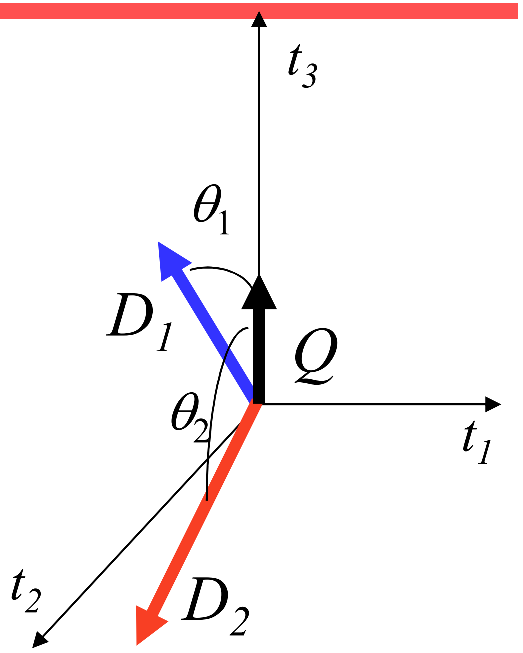
$Q = 0T_1 + 0T_2 + 2T_3$

D_1 is 6 times better than D_2 using cosine similarity but only 5 times better using inner product.

Euclidean Distance

$$SIM(d,q) = \left[\sum_i |t_i - q_i|^p \right]^{\frac{1}{p}}$$

$D_1 = 2T_1 + 3T_2 + 5T_3$
$D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$



TO COMPUTE THE EUCLIDEAN DISTANCE

Comments on Vector Space Model

- Provides partial matching and ranked results.
- Tends to work quite well in practice despite obvious weaknesses.
- Allows efficient implementation for large document collections.
- Missing semantic information (e.g. word sense).
- Missing syntactic information (e.g. phrase structure, word order, proximity information).
- Assumption of term independence (e.g. ignores synonomy).
- Lacks the control of a Boolean model (e.g., *requiring* a term to appear in a document).
  - Given a two-term query "A B", may prefer a document containing A frequently but not B, over a document that contains both A and B, but both less frequently.

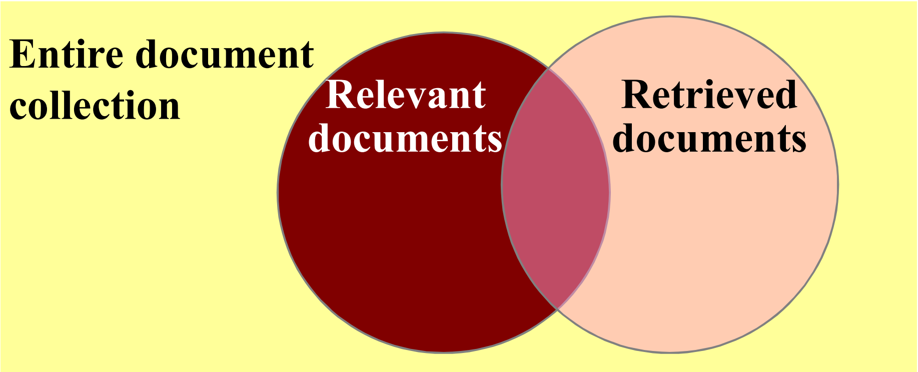
IRS Evaluation

- Two kinds of evaluation measures:
  - **Efficiency**: answer time or storage measures (Not used)
  - **Effectiveness**: to evaluate if the IRS works correctly o not

  The IRS evaluation is measured by the IRS ***effectiveness*** which is related to the ***relevance*** of retrieved items.
  - Relevance, from a human standpoint, is:
    - Subjective: Depends upon a specific user's judgment.
    - Situational: Relates to user's current needs.
    - Cognitive: Depends on human perception and behavior.
    - Dynamic: Changes over time.

- Analytic performance: Benchmarking tools which requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document

Effectiveness Measures: Precision and Recall



	retrieved	not retrieved
irrelevant	retrieved & irrelevant(A)	Not retrieved & irrelevant (B)
relevant	retrieved & relevant(C)	not retrieved but relevant (D)

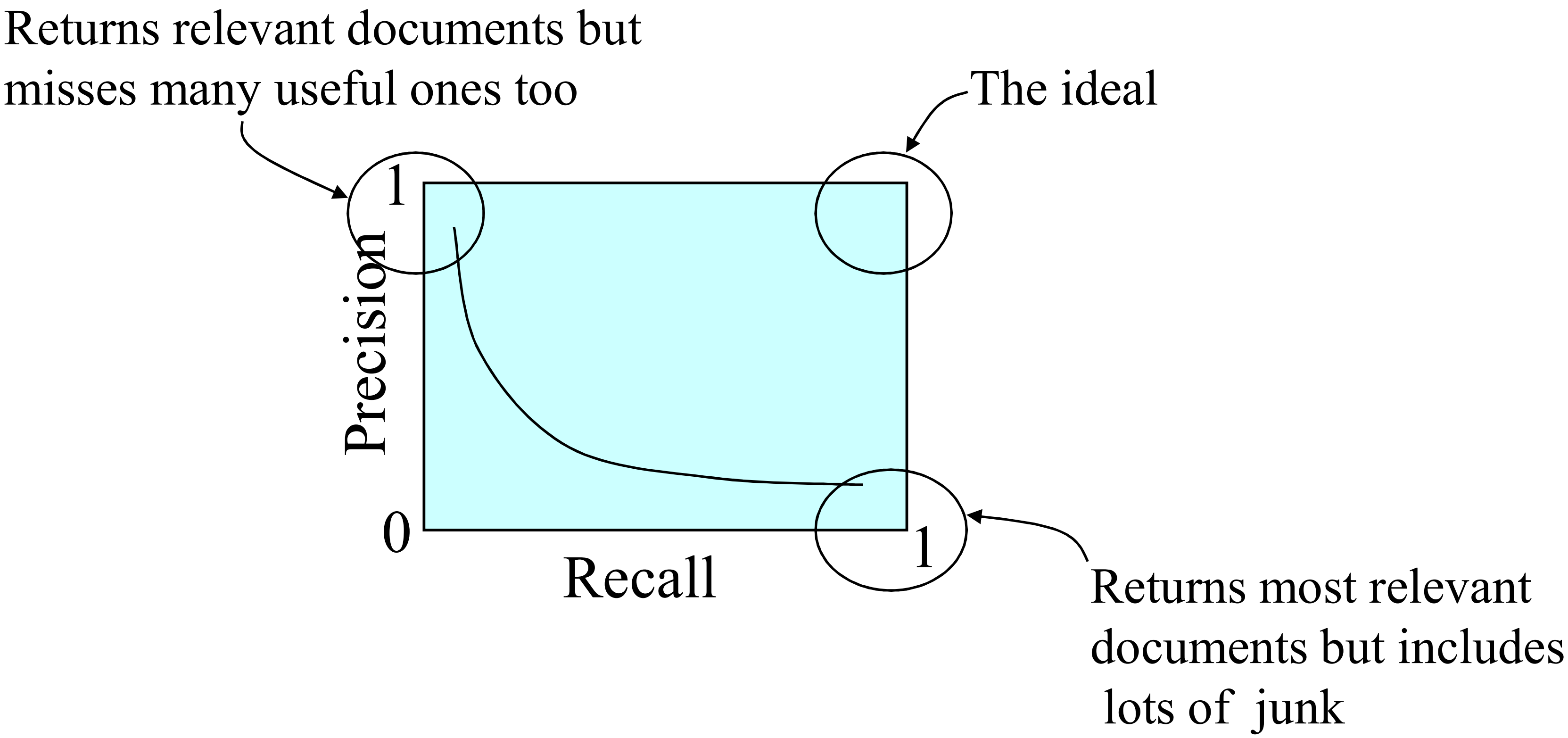
$$recall = \frac{Number\ of\ relevant\ documents\ retrieved\ (C)}{Total\ number\ of\ relevant\ documents\ (C+D)}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved\ (C)}{Total\ number\ of\ documents\ retrieved\ (C+A)}$$

Precision and Recall

- Precision:
  - The ability to retrieve top-ranked documents that are mostly relevant.
- Recall:
  - The ability of the search to find ***all*** of the relevant items in the corpus.
  - **Problem**: Total number of relevant items is sometimes not available

Trade-off between Recall and Precision



Returns relevant documents but misses many useful ones too

The ideal

Returns most relevant documents but includes lots of junk

Precision and Recall present opposite behaviour in IRS

F-Measure

- One measure of performance that takes into account both recall and precision.
- Harmonic mean of recall and precision:

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R}+\frac{1}{P}}$$

- Compared to arithmetic mean, both need to be high for harmonic mean to be high.

E Measure (parameterized F Measure)

- A variant of F measure that allows weighting emphasis on precision over recall:

$$E = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R}+\frac{1}{P}}$$

- Value of β controls trade-off:
  - β = 1: Equally weight precision and recall (E=F).
  - β > 1: Weight recall more.
  - β < 1: Weight precision more.

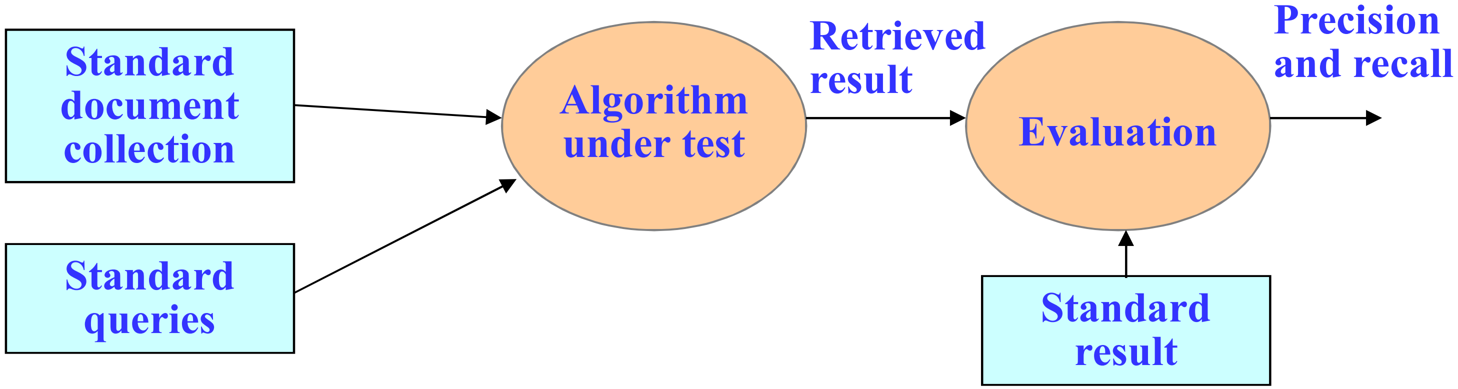
IRS Evaluation: Experimental Setup for Benchmarking

- Performance is measured by ***benchmarking***. That is, the retrieval effectiveness of a system is evaluated on a *given set of documents*, *queries*, and *relevance judgments*.

- Performance data is valid only for the particular environment under which the system is evaluated.

IRS Evaluation: Benchmarks

- A benchmark collection contains:
  - A set of standard documents and queries/topics.
  - A list of relevant documents for each query.
- Standard collections for traditional IR:



Standard document collection

Standard queries

Algorithm under test

Retrieved result

Evaluation

Standard result

Precision and recall

IRS Evaluation: Test Collections

- Early Test Collections: Previous experiments were based on the SMART collection which is fairly small. (ftp://ftp.cs.cornell.edu/pub/smart)

Collection Name	Number Of Documents	Number Of Queries	Raw Size (Mbytes)
CACM	3,204	64	1.5
CISI	1,460	112	1.3
CRAN	1,400	225	1.6
MED	1,033	30	1.1
TIME	425	83	1.5

- TREC Collections: Text REtrieval Conference .

IRS Evaluation: The TREC Objectives

- Provide a common ground for comparing different IR techniques.
  - Same set of documents and queries, and same evaluation method.
- Sharing of resources and experiences in developing the benchmark.
  - With major sponsorship from government to develop large benchmark collections.
- Encourage participation from industry and academia.
- Development of new evaluation techniques, particularly for new applications.
  - Retrieval, routing/filtering, non-English collection, web-based collection, question answering.

IRS: Improving the performance

1. Provide users more information about the IRS performance:
   - Use of thesaurus
   - Help to represent queries
2. Provide user information to the IRS (RELEVANCE FEEDBACK):
   - After initial retrieval results are presented, allow the user to provide feedback on the relevance of one or more of the retrieved documents.
   - Use this feedback information to reformulate the query.
   - Produce new results based on reformulated query.

IRS: Improving the performance Applying Tools of Related Areas

- Library and Information Science
- Artificial Intelligence
- Natural Language Processing
- Machine Learning

Library and Information Science

- Focused on the human user aspects of information retrieval (human-computer interaction, user interface, visualization) (Cognitive aspects of IR).
- Concerned with effective categorization of human knowledge.
- Concerned with *citation analysis* and *bibliometrics* (structure of information).
- Recent work on *digital libraries* brings it closer to CS & IR.

Artificial Intelligence

- Focused on the representation of knowledge, reasoning, and intelligent action.
- Formalisms for representing knowledge and queries.
  - Soft Computing Tools: Fuzzy Logic, Genetic Algorithm, Neuronal Networks, ect.

Natural Language Processing

- Focused on the syntactic, semantics, and pragmatic analysis of natural language text and discourse.
- Ability to analyze syntax (phrase structure) and semantics could allow retrieval based on *meaning* rather than keywords.

Natural Language Processing: IR Directions

- Methods for determining the sense of an ambiguous word based on context (*word sense disambiguation*).
- Methods for identifying specific pieces of information in a document (*information extraction*).
- Methods for answering specific NL questions from document corpora.

Machine Learning

- Focused on the development of computational systems that improve their performance with experience.
- Automated classification of examples based on learning concepts from labeled training examples (*supervised learning*).
- Automated methods for clustering unlabeled examples into meaningful groups (*unsupervised learning*).