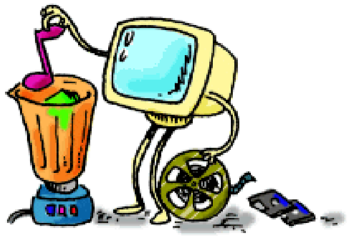
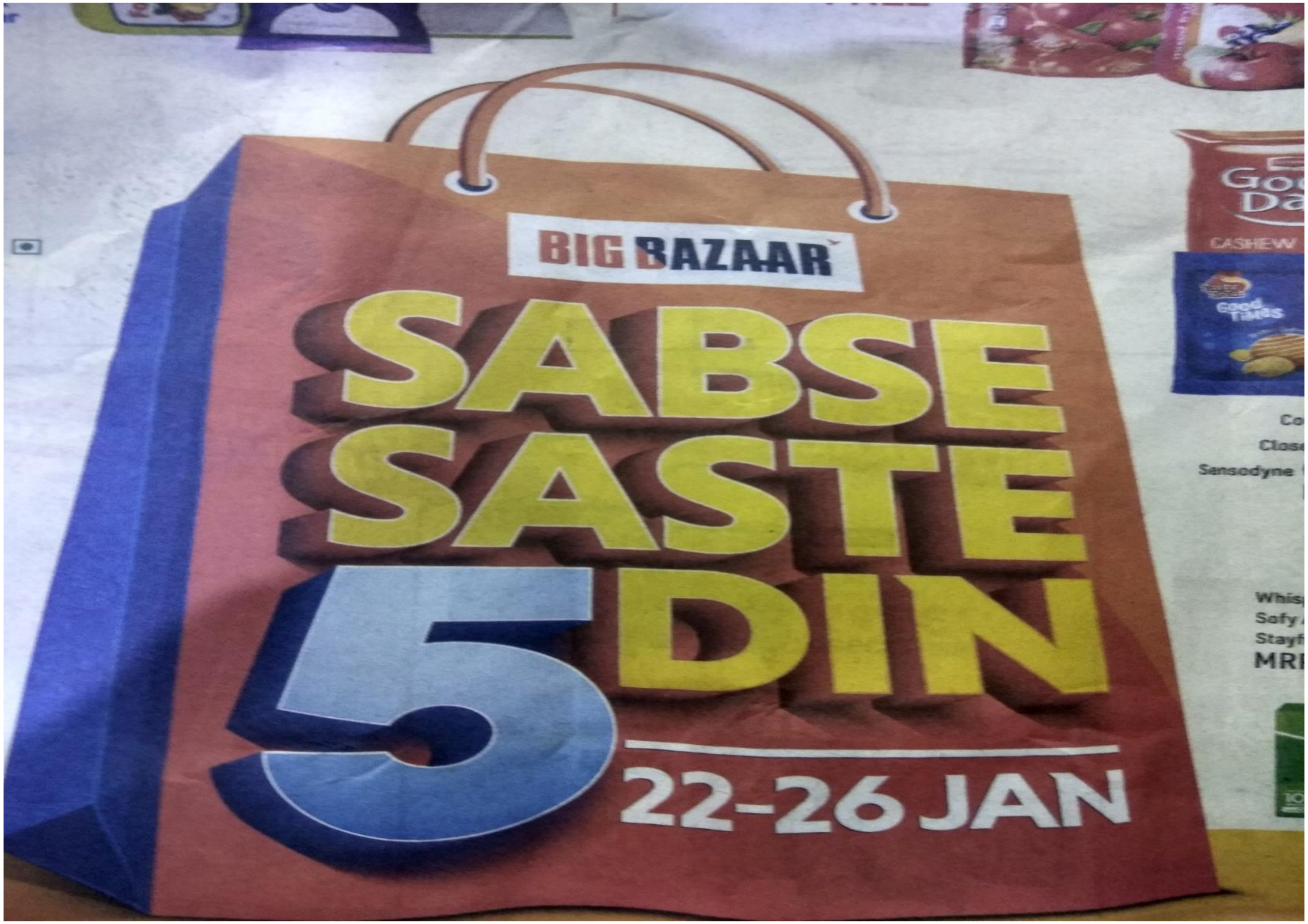


Sub:-Information Retrieval

PATTERSON MULTI
FREQUENCY INVERTED
ONE QUERIES CONFUSING PROGRESSED WEIGHTING
WORD BUSINESS
MONEY QUEUING DATA MATHEMATICAL
ENGINES MICROSOFT
PURPLE REUTERS KNOWING
SORT EVALUATE COLORFUL STEMMING DOCUMENTS DIFFICULT SCORE SPEEDY
STUDIED DIGGING AGGREGATE INTERESTING STATISTICS INTRACTABLE
GROWING DISCOVERING COSINE ENTHRALLING MAGICAL
BIG TECHNOLOGICAL SPECIAL COOL RELATIONAL AMAZING CORPUS QUERY BRIGHT NEEDED
UNSTRUCTURED HOW RANK INDEX CHALLENGE NEED
STOP IMPORTANT MATH
MODIFY TECHNICAL CONSUMING LARGE COMPLEX INFO FOR ORGANIZATION WEB PARSING DOCS OF TEXT AND GET HADOOP FAST
SEARCH
INFORMATION TO EFFICIENT IN LIST MAP WAY SCORES FUN
SLOW
EXPENSIVE SCIENCE CRAWL RETRIEVE METADATA A GOOGLE ALL VAST SEARCHING
WWW HARD PAGE TIME TRUST ABOUT NOT DOCUMENT CRAWLER
INDEXING ALGORITHMIC THE TRICKY
STUFF SAD OVERLOAD CRAWLERS FUTURE COMPLICATED
THINGS QUIRKY SEXY SCHOLARLY EVOLVING RETRIEVAL DATABASE
SORTING EXCITING SEARCHES FRUSTRATING EFFICIENTLY
WIKIPEDIA RETRIEVING
MOVE CRAWLING USEFUL DISTRIBUTED
KEY DIFFERENT VARIED DISTRIBUTEDLY
THROUGH
POSTING RELEVANT GATHER LINEAR
HAPPY POLITE

Information Retrieval From Images



Document Recognition (TAGGING/ -- ***LABELING***/ANNOTATION) (Application area)

Header
Style 1
Style 2
Page Number

This is GLA University → *Tagging* →

E,H,O,NER

Mr. Amitabh Bachhan

Is University ka syllabus kaisa hai?

I am going to the market

Mai Market jaunga

Main Market Jaunga

Main Main Market Jaunga

Cont..

- Application Area: related to the area of Human computer Interaction.
- Ex:
- **Paani ,pehla**
- **Paanee ,Pahla**
- **Paanie ,Pahlaa**
- **Paanii, Pehlaa etc…**

Cont..

- Text Analysis or Text Processing: Process of gleaning high quality and meaningful information (through devising of patterns and trends by means of statistical pattern learning) from text.
- Ex: Text categorization- H,E,O etc….
- Text clustering
- Sentiment analysis-- joy,sad,anger,hatred etc..
- Content/Intent extraction

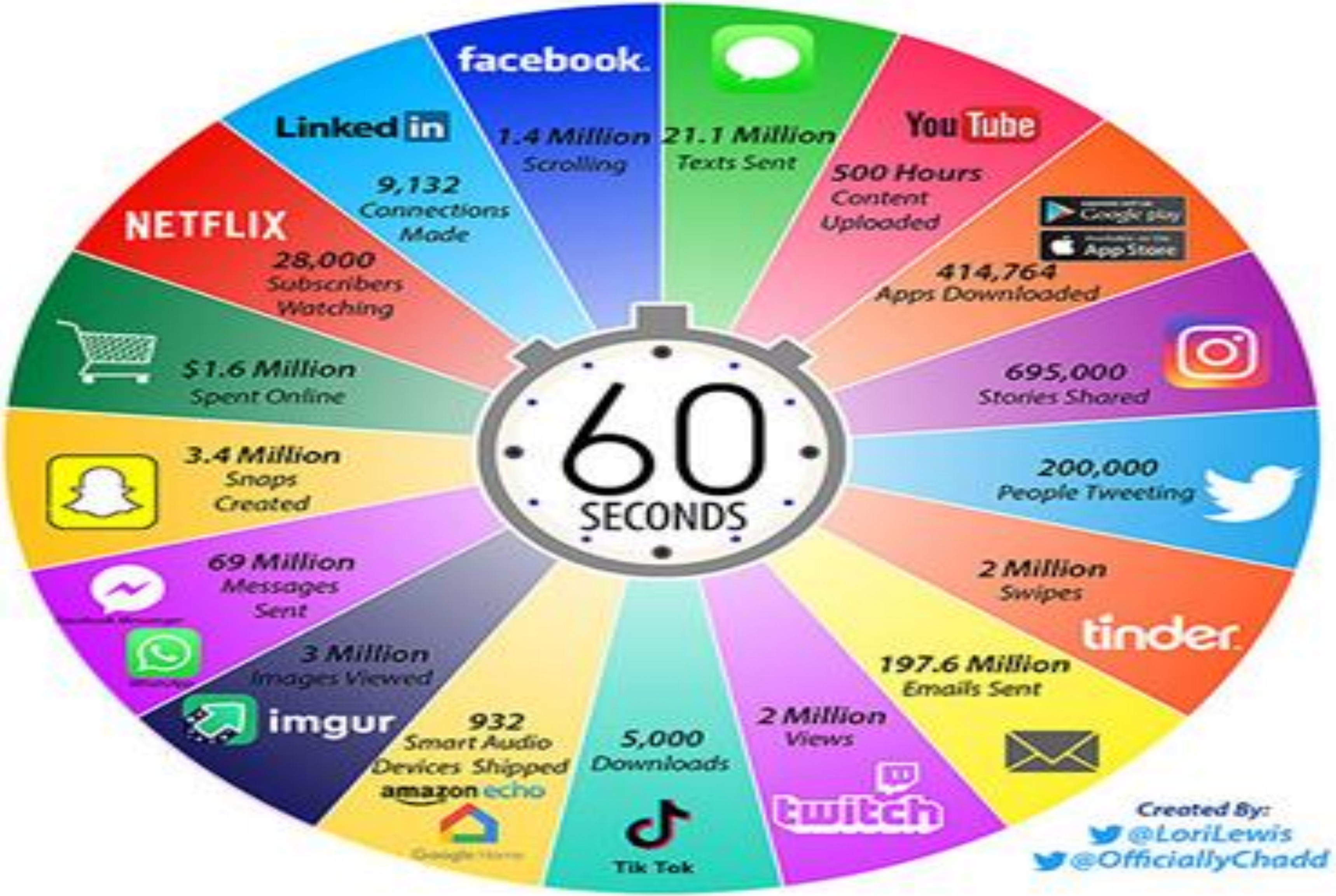
Dr. Shashi Shekhar
Associate Professor
Deptt. Of CEA
Contact Details:
Email: shashi.shekhar@gla.ac.in
Contact Number:
8474970309

Data comes from?

Internet 60 Sec. Use in 2021.

Data in 1 minute

2021 This Is What Happens In An Internet Minute



- We will move together

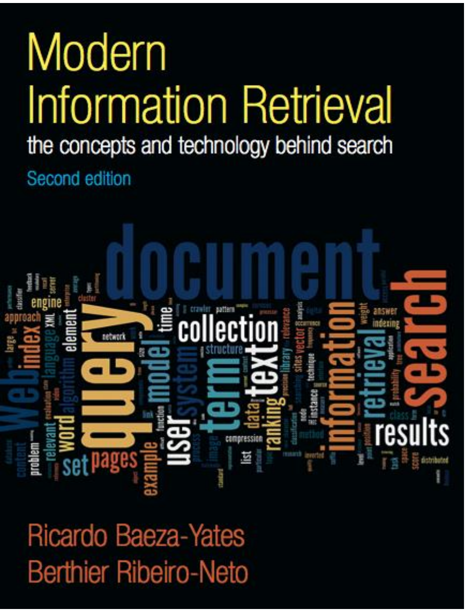
From Data

To

Decisions

Reference Books
Information Retrieval

- 1. Ricardo Baeza-Yate, Berthier Ribeiro-Neto, “**Modern Information Retrieval**”, Pearson Education Asia, 2007.



Ricardo Baeza-Yates is VP of Research for Europe and Latin America, leading the Yahoo! Research labs at Barcelona, Spain and Santiago, Chile, and also supervising the lab in Haifa, Israel.

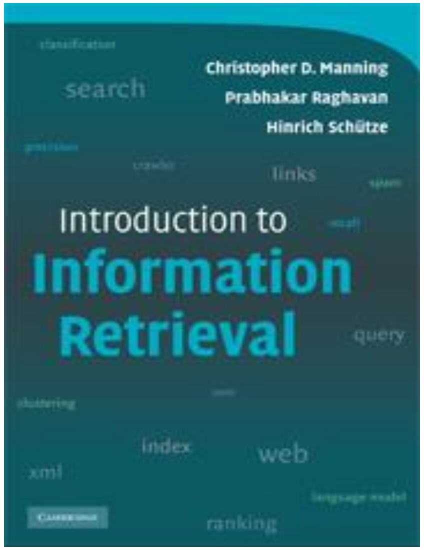
Berthier Ribeiro-Neto is an Associate Professor at the CS Dept .of the Federal University of Minas Gerais(Brazil). His research interests include information retrieval (IR) systems for the Web, digital libraries, and video on demand.

Ricardo Baeza - Yates

Berthier Ribeiro-Neto

Reference Books
Information Retrieval

- 4. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.



Christopher Manning

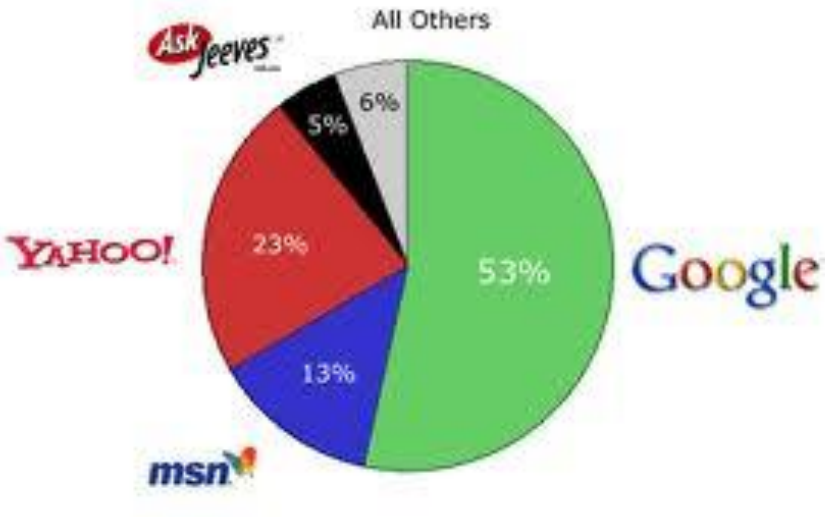
Associate Professor of Computer Science and Linguistics & Sony Faculty Scholar, Stanford University, USA

Prabhakar Raghavan

Chief Strategy Officer and Head, Yahoo! Labs ,Yahoo! 701 First Avenue,
Sunnyvale,USA

Hinrich Schutze

Chair of Theoretical Computational Linguistics
Institute for Natural Language Processing
University of Stuttgart, Germany



IR Course Goals

- To help you to understand IR Modeling, Text operations, search engines, evaluate and compare them, and modify them for specific applications
- Provide broad coverage of the important issues in information retrieval and search engines
  - includes underlying models and current research directions

Organization

- Lectures
  - 3 Lectures --Monday to Friday

- Course format
  - Lectures + Quiz + Programming Based assignments
  - Small IR Based Project.

Syllabus: BCSE0154 : INFORMATION RETRIEVAL SYSTEMS

- **I**

- **Introduction to IR:** IR Concepts, Boolean Retrievals- An Example Information Retrieval Problem, A First Take at Building an Inverted Index, Processing Boolean Queries.
- **The Term Vocabulary and Postings Lists:** Document Delineation and Character Sequence Decoding, Determining the Vocabulary of Terms.
- **Dictionaries and Tolerant Retrieval:** Search Structures for Dictionaries, Wildcard Queries, Spelling Correction, Phonetic Correction.
- **Index Construction:** Hardware Basics Blocked Sort-Based Indexing. Scoring, Term Weighting and the Vector Space Model: Parametric and Zone Indexes, Term Frequency and Weighting, The Vector Space Model for Scoring.

- **II**

- **Evaluation in Information Retrieval:** Information Retrieval System Evaluation, Standard Test Collections, Evaluation of Unranked Retrieval Sets, Evaluation of Ranked Retrieval Results.
- **XML Retrieval:** Basic XML Concepts, Challenges in XML Retrieval, A Vector Space Model for XML Retrieval, Evaluation of XML Retrieval, Text-Centric vs. Data-Centric XML Retrieval.
- **Web Search Basics:** Web Characteristics, Advertising as the Economic Model, The Search User Experience, Index Size and Estimation, Near-Duplicates and Shingling.
- **Web Crawling and Indexes:** Overview, Crawling, Distributing Indexes, Connectivity Servers.
- **Link Analysis:** The Web as a Graph, Page Rank, Hubs and Authorities.

Information Retrieval

- *"Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."* (Salton, 1968)
- Information retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which he is interested (Ricardo Baeza-Yate, Berthier Ribeiro-Neto).

Information Retrieval

- Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

  - These days we frequently think first of web search, but there are many other cases:
    - E-mail search
    - Searching your laptop for text,image,audio,video
    - Corporate knowledge bases
    - Legal information retrieval

IR Motivation

- Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the user.
- The emphasis is on the retrieval of information as opposed to the retrieval of data.
- An *Information Retrieval (IR) System* attempts to find

  relevant documents to respond to a user’s request.

Hard Parts of IR

- One word can have a zillion(extremely large) different semantic meanings
- Consider: Take
- – "take a place at the table"
- – "take money to the bank"
- – "take a picture"
- – "take a lot of time"
- – "take Medicine"

What is Different about IR from the rest of Computer Science

□ Most algorithms in computer science have a “right” answer:

Consider the two problems:

□ – Sort the following ten integers

□ – Find the highest integer among given integers

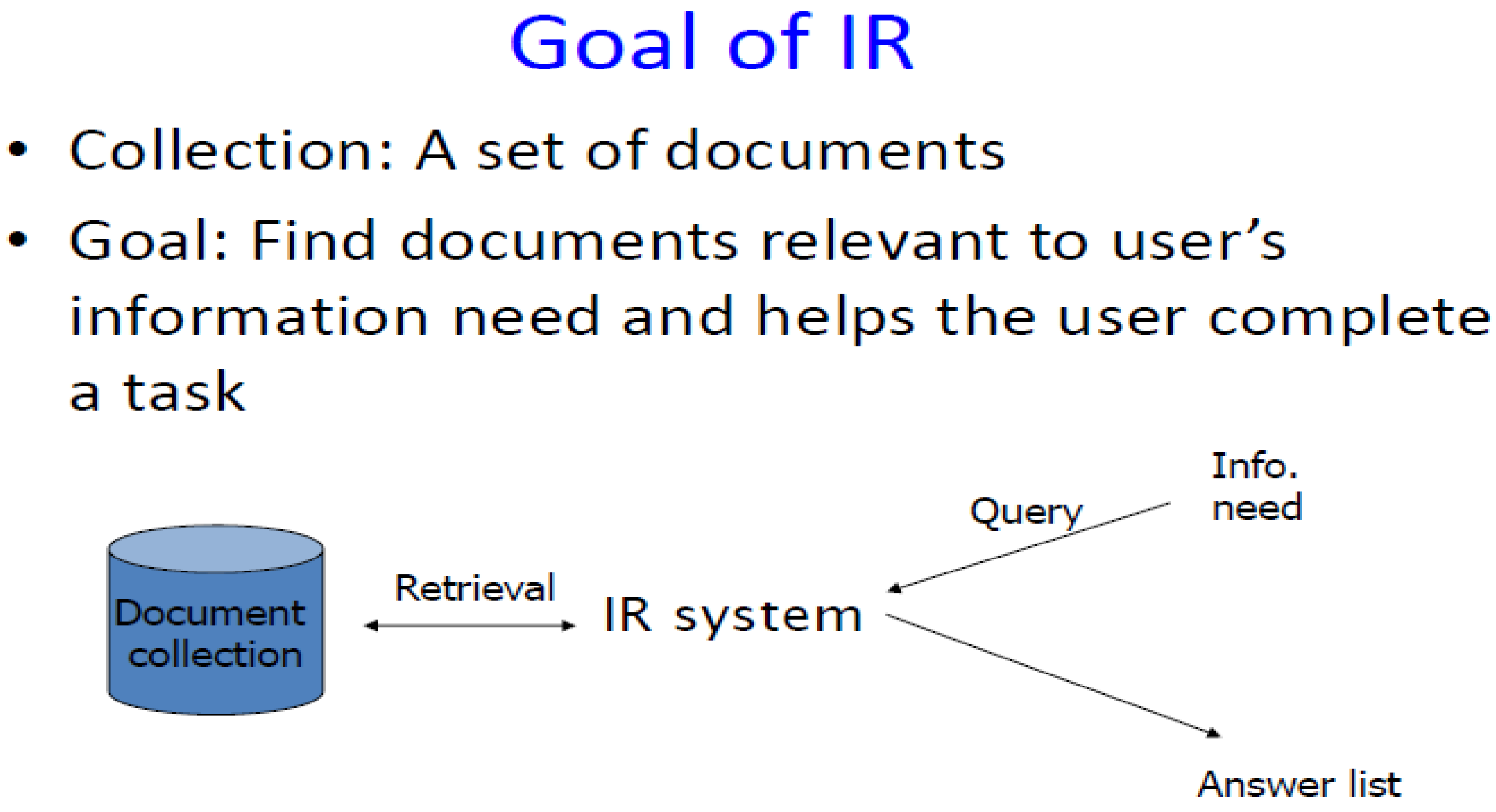
Now consider:

□ – *Find the document most relevant to “Apple as a Company”*

Measuring Effectiveness

- An algorithm is deemed incorrect if it does not have a “right” answer.
- A heuristic tries to guess something close to the right answer. Heuristics are measured on “how close” they come to a right answer.
- IR techniques are essentially heuristics because we do not know the right answer.
- So we have to measure how *close to the* right answer we can come.

IR system Goal



Goal of IR

- Collection: A set of documents
- Goal: Find documents relevant to user’s information need and helps the user complete a task

Important Terms used in IR

WORLD WIDE WEB

- The World-Wide Web was developed to be a pool of human knowledge and human culture, which would allow collaborators of remote sites to share their ideas and all aspects of a common project.
- It is a system of interlinked hypertext documents accessed via the Internet.
- With a web browser, one can view web pages that may contain text, images, animation, videos and other multimedia and navigate between them via hyperlinks.

Hypertext model

- A model of information retrieval based on representing document relationships as edges of a generic graph in which the documents are the nodes.
- The web can be viewed as a graph:
- The nodes represent individual pages
- The edges represent links between pages

Clustering

- The grouping of documents which satisfy a set of common properties.
- The aim is to assemble together documents which are related among themselves.

Data Retrieval

- The retrieval of items (tuples) whose contents satisfy the conditions specified in a (Relational algebra like) user query.

Document

- A unit of retrieval. It might be a paragraph, a section, a chapter, a Web page, an article, or a whole book.
- A document is a sequence of terms, expressing ideas about some topic in a natural language.

Index term: (or keyword)

- A pre-selected term which can be used to refer to the content of a document.
- In the Web, however, some search engines use all the words in a document as index terms.

Federated search

- It is an information retrieval technology that allows the simultaneous search of multiple searchable resources.
- A user makes a single query request which is distributed to the search engines participating in the group.
- The federated search then aggregates the results that are received from the search engines for presentation to the user.

Interoperability

- The working together of a number of computer systems, typically for a common purpose, such as when a number of digital libraries “support federated searching”,often enabled by standards and agreed-upon conventions including data formats and protocols.

Boolean Model

- The *Boolean retrieval model is a model for information retrieval in which we* can pose any query which is in the form of a Boolean expression of terms, that is, in which terms are combined with the operators AND, OR, and NOT.
- The model views each document as just a set of words.

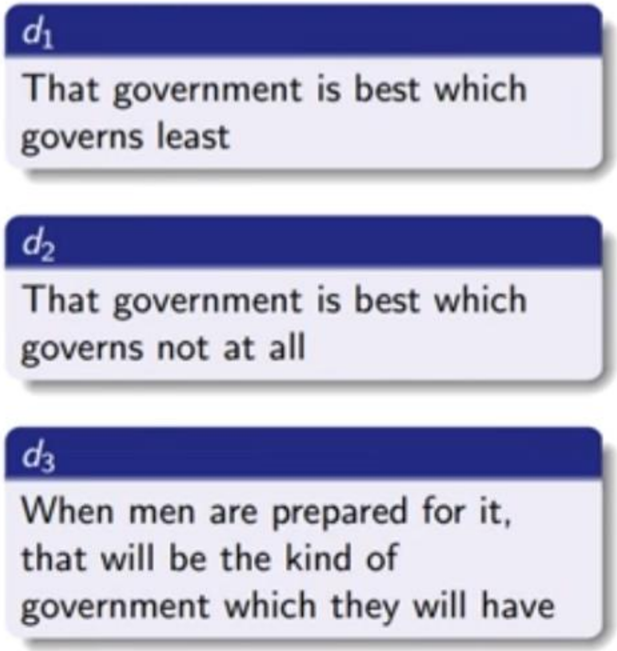
BOOLEAN MODEL

- The Boolean model of information retrieval is one of the earliest and simplest retrieval methods that use the method of exact matching to match documents according to the user's query wherein words are logically combined by using Boolean operators like AND, OR, and NOT.
- For example, the Boolean AND of two logical statements x and y means that both x AND y must be satisfied, while the Boolean OR of these two statements means that at least one of these statements must be satisfied.

The Boolean Model

- It is the oldest information retrieval (IR) model. The model is based on set theory and the Boolean algebra, where documents are sets of terms and queries are Boolean expressions on terms. The Boolean model can be defined as −
- **D** − A set of words, i.e., the indexing terms present in a document. Here, each term is either present (1) or absent (0).
- **Q** − A Boolean expression, where terms are the index terms and operators are logical products − AND, logical sum − OR and logical difference − NOT

BOOLEAN MODEL



d_1

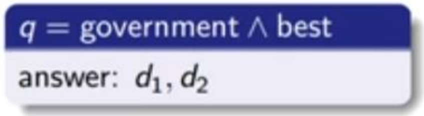
That government is best which governs least

d_2

That government is best which governs not at all

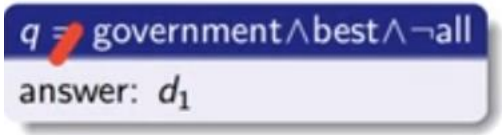
d_3

When men are prepared for it, that will be the kind of government which they will have



$q = \text{government} \land \text{best}$

answer: d_1, d_2



$q = \text{government} \land \text{best} \land \neg \text{all}$

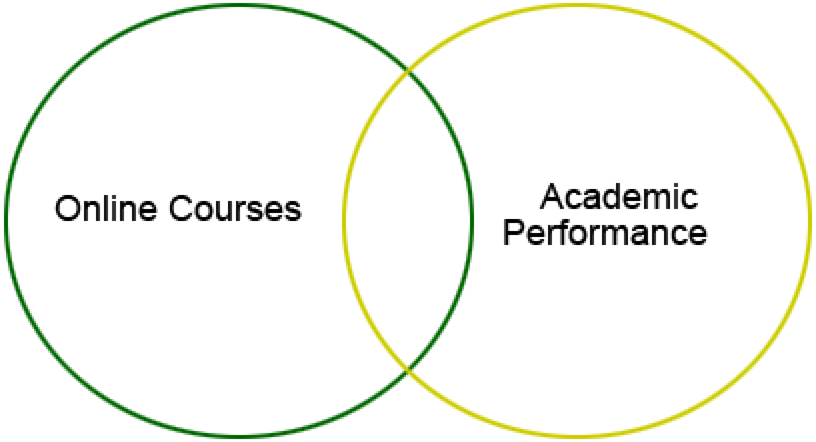
answer: d_1

BOOLEAN MODEL…

- In this model, large number of logical statements can be combined using the three Boolean operators.
- The three Boolean operators are **AND**, **OR** and **NOT**.
- This model operates by considering which keywords of the user query are present in a document.
- Thus, if keywords are found in a document, then document is called relevant.
- Infact, there is no concept of a partial match between documents and queries. This strategy can lead to poor performance .

BOOLEAN MODEL…

- AND
- Use **AND** to narrow your search: all of your search terms will present in the retrieved records.
- The oval in the middle of the Venn diagram below represents the result set for this search. It is the combination of these two search terms.
- Example: Online courses AND academic performance



BOOLEAN MODEL…

- **OR**
- Use **OR** to broaden your search by connecting two or more synonyms.
- Example: online courses OR Web-based instruction OR distance education
- The database retrieves all the unique records containing one term, the other, or both.
- **NOT**
- Use **NOT** to exclude term(s) from your search results.
- Example: higher education NOT community colleges

The Boolean Model…

Simple model based on **set theory** and **boolean algebra**

Queries specified as boolean expressions

- quite intuitive and precise semantics
- neat formalism
- example of query

$$q = k_a \wedge (k_b \vee \neg k_c)$$

Term-document frequencies in the term-document matrix are all binary

- $w_{ij} \in \{0, 1\}$: weight associated with pair (k_i, d_j)
- $w_{iq} \in \{0, 1\}$: weight associated with pair (k_i, q)

The Boolean Model…

The similarity of the document d_j to the query q is defined as

$$sim(d_j, q) = \begin{cases} 1 & \text{if } \exists c(q) \mid c(q) = c(d_j) \\ 0 & \text{otherwise} \end{cases}$$

The Boolean model predicts that each document is either relevant or non-relevant

Drawbacks of the Boolean Model

- Retrieval based on binary decision criteria with no notion of partial matching
- No ranking of the documents is provided (absence of a grading scale)
- Information need has to be translated into a Boolean expression, which most users find awkward
- The Boolean queries formulated by the users are most often too simplistic
- The model frequently returns either too few or too many documents in response to a user query

Advantages of the Boolean Model

The advantages of the Boolean model are as follows −

- The simplest model, which is based on sets.
- Easy to understand and implement.
- It only retrieves exact matches
- It gives the user, a sense of control over the system.

Vector Space Model

- Consider the following important points to understand more about the Vector Space Model −
- The index representations (documents) and the queries are considered as vectors embedded in a high dimensional Euclidean space.
- The similarity measure of a document vector to a query vector is usually the cosine of the angle between them.

Cosine Similarity Measure Formula

Cosine is a normalized dot product, which can be calculated with the help of the following formula −

$$\boxed{Score(\vec{d}\,\vec{q}) = \frac{\sum_{k=1}^{m} d_k \,.\, q_k}{\sqrt{\sum_{k=1}^{m} (d_k)^2} \,.\, \sqrt{\sum_{k=1}^{m} m(q_k)^2}}}$$

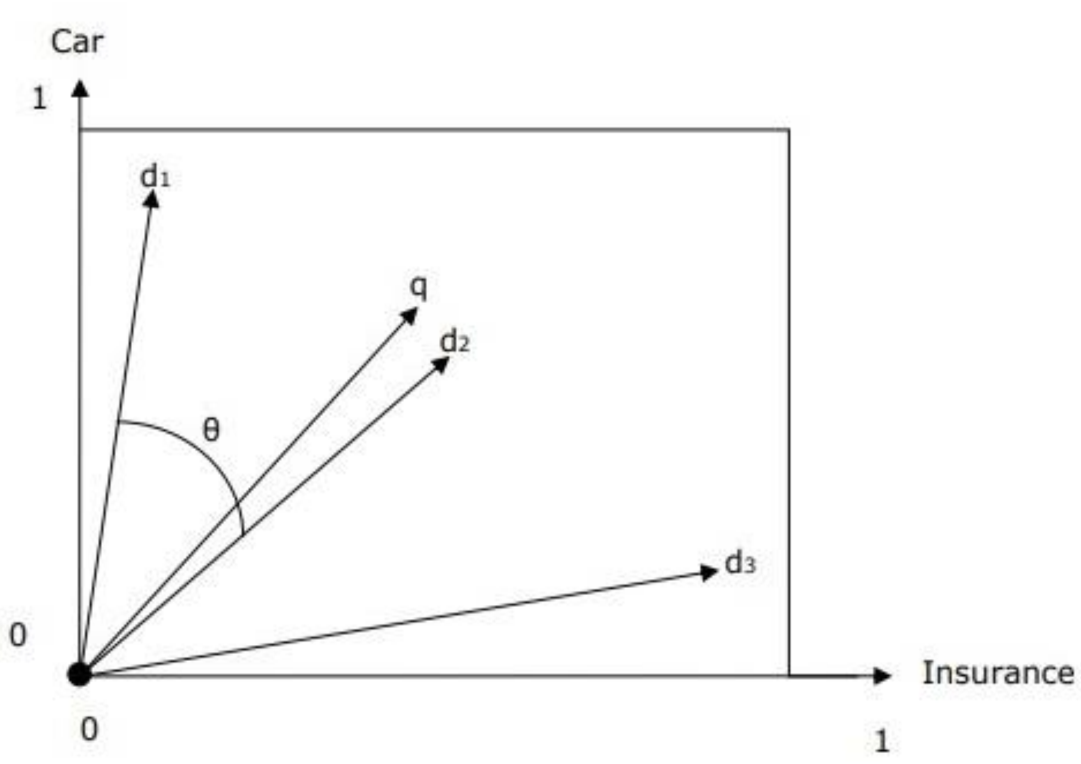
$$Score(\vec{d}\,\vec{q}) = 1\ when\ d = q$$

$$Score(\vec{d}\,\vec{q}) = 0\ when\ d\ and\ q\ share\ no\ items$$

Vector Space

- Vector Space Representation with Query and Document
- The query and documents are represented by a two-dimensional vector space. The terms are ***car*** and ***insurance***. There is one query and three documents in the vector space.

Vector space



- The top ranked document in response to the terms car and insurance will be the document $\mathbf{d_2}$ because the angle between **q** and $\mathbf{d_2}$ is the smallest. The reason behind this is that both the concepts car and insurance are salient in d_2 and hence have the high weights. On the other side, $\mathbf{d_1}$ and $\mathbf{d_3}$ also mention both the terms but in each case, one of them is not a centrally important term in the document.

Statistical Approaches

- Word 2 Vec concepts,
- TF-IDF computation,
- Inverted Index construction ,
- Document Incidence Matrix construction,
- Text similarity methods—Similarity coefficient ,Jaccard similarity, Cosine similarity.

- Word vectors are **simply vectors of numbers that represent the meaning of a word.** ... In simpler terms, a word vector is a row of real-valued numbers.
- **Where Do Word Vectors Come From?**
- An excellent question at this point is, *Where do these dimensions and weights come from?!* There are two common ways through which word vectors are generated:
- Counts of word/context co-occurrences
- Predictions of context given word (skip-gram neural network models, i.e. word2vec)

Word 2Vec

- *Similar words are mapped together in the vector space. Notice how close "****cat" and "dog" are to "pet,"*** *how clustered "elephant," "lion," and "tiger" to Zoo, Wild etc..*

Inverted index

- A text index composed of a vocabulary and a list of occurrences.
- It is an index data structure storing a mapping from content, such as words or numbers, to its locations in a database file, or in a document or a set of documents.
- The purpose of an inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database.

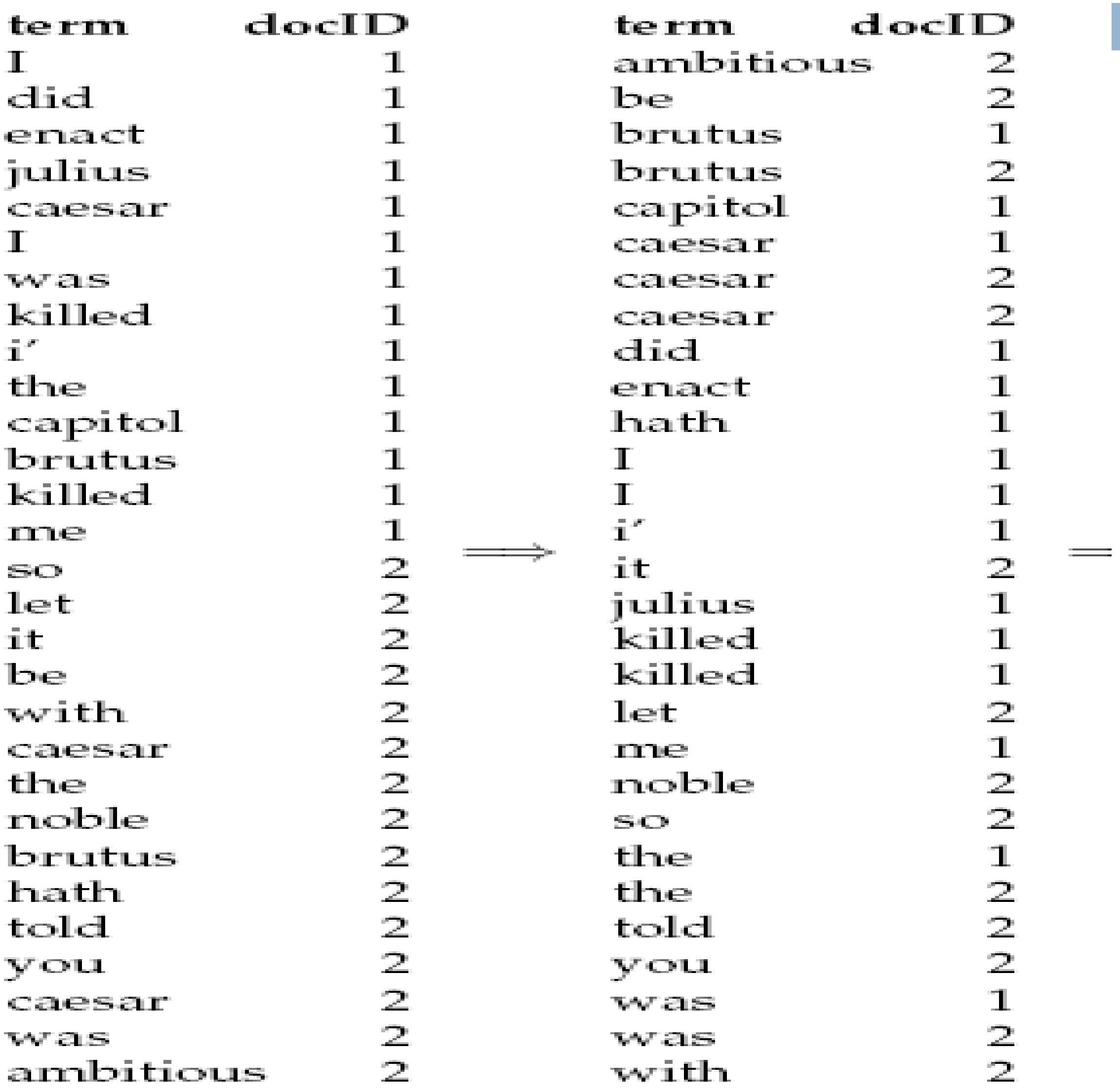
Building Inverted index

- 1. Collect the documents to be indexed:
- 2. Tokenize the text, turning each document into a list of tokens:
- 3. Do linguistic preprocessing, producing a list of normalized tokens, which are the indexing terms.
- 4. Index the documents that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

Ex. Of inverted index

- **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:

Tokenize the text, turning each document into a list of tokens:



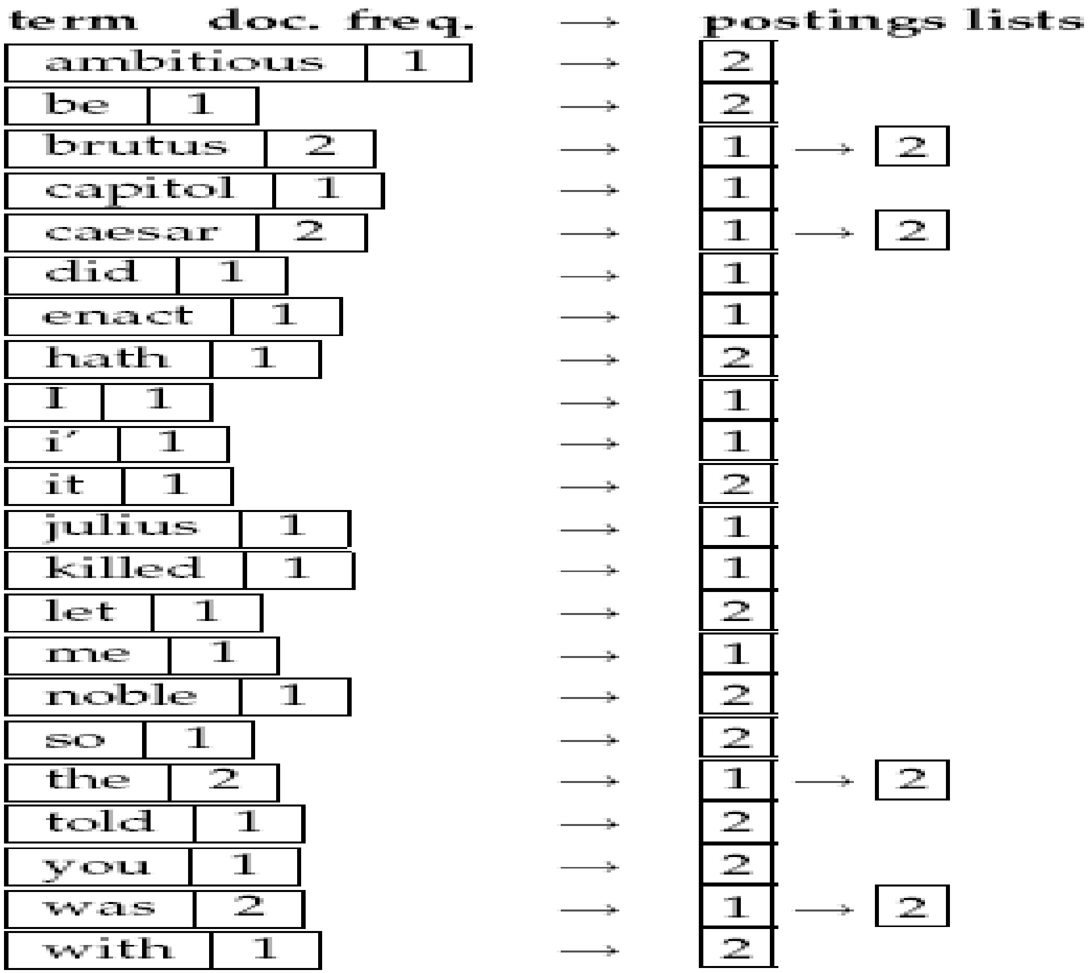
term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

⟹

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

=

Building an index by sorting & grouping

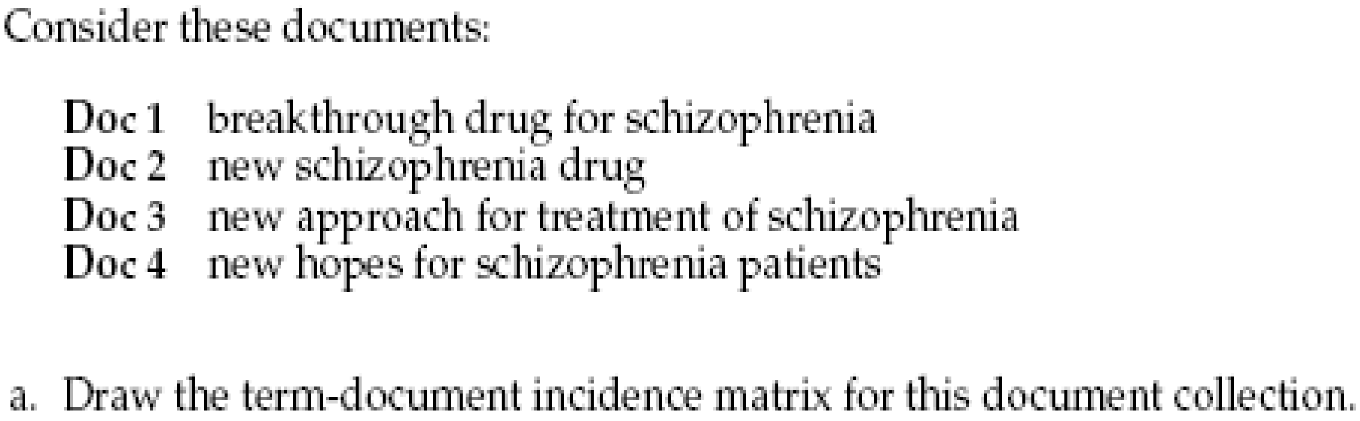


term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Inverted index

- Ambitious → 2
- be → 2
- Brutus→ 1→2
- capitol →1
- Caeser→1→2 and so on for others

Document incidence Matrix



Consider these documents:

Doc 1 breakthrough drug for schizophrenia
Doc 2 new schizophrenia drug
Doc 3 new approach for treatment of schizophrenia
Doc 4 new hopes for schizophrenia patients

a. Draw the term-document incidence matrix for this document collection.

b. Draw the inverted index representation for this collection,

For the document collection shown what are the returned results for these queries:

A) schizophrenia AND drug

B) for AND NOT(drug OR approach)

TERM DOCUMENT INCIDENCE MATRIX

	D1	D2	D3	D4
Approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
for	1	0	1	1
hopes	0	0	0	1
new	0	1	1	1
of	0	0	1	0
patients	0	0	0	1
Schizophrenia	1	1	1	1
treatment	0	0	1	0

Inverted Index:

- Approach -> 3
- breakthrough ->1
- drug ->1->2
- for ->1->3->4
- hopes ->4
- new -.>2->3->4
- of ->3
- patients ->4
- schizophrenia ->1->2->3->4
- treatment >3

- Build inverted index
- Design term Doc. Matrix

	D1	D2	D3	D4
approach	0	0	1	0

approach as token available in Document 3 therefore term document matrix for word approach is:

Approach 0 0 1 0

So we have 0/1 vector for each term

Entry is 1 if term occurs

Entry is 0 if term does not occur

Answer

- For the document collection shown what are the returned results for these queries:
- **A) schizophrenia AND drug**
- **B) for AND NOT(drug OR approach)**

- (i) doc1, doc2
- (ii) doc4

inverted index Question

- Draw the inverted index that would be built for the following document collection.
- **Doc 1** new home sales top forecasts
- **Doc 2** home sales rise in july
- **Doc 3** increase in home sales in july
- **Doc 4** july new home sales rise

Answer

- **Inverted Index:**
- forecast->1
- home->1->2->3->4
- in->2->3
- increase->3
- july->2->3
- new->1->4
- rise->2->4
- sale->1->2->3->4
- top->1

Jaccard Similarity and Shingling

- We will study how to define the distance between sets, specifically with the Jaccard distance.
- Jaccard distance measures the distance between documents.

Applications:

- Given two homework assignments (reports) how can a computer detect if one is likely to have been plagiarized from the other without understanding the content?
- In trying to index WebPages, how does Google avoid listing duplicates or mirrors?
- How does a computer quickly understand emails, for either detecting spam or placing effective advertisers?

- The key to answering these questions will be convert the data (homeworks, webpages, emails) into an object in an abstract space that we know how to measure distance, and how to do it efficiently
  - **Consider two sets A = {0, 1, 2, 5, 6} and B = {0, 2, 3, 5, 7, 9}. How similar are A and B?**
- The Jaccard similarity is defined
- **JS(A, B) = |A ∩ B| / |A ∪ B|**

  = |{0, 2, 5}|/ |{0, 1, 2, 3, 5, 6, 7, 9}|

  = 3/ 8 = 0.375

Example slide: Jaccard coefficient

- S_{AB}=A□B/A□B
-
- A={7,3,2,4,1}
- B={4,1,9,7,5}

****

A={0,1,2,5,6}

B={0,2,3,4,5,7,9}

	Fruit Features	Sphere shape	Sweet	Sour	Crunchy
Object 1	Apple	Yes	Yes	Yes	Yes
Object 2	Banana	No	Yes	No	No

- Vector of Apple= 1 1 1 1
- Vector of Banana= 0 1 0 0

- Using Boolean Algebra (A and B is TRUE if Both True, A or B is False if both false)
- Intersection Set is equivalent to AND
- While Union Operation is equivalent to OR
- A= 1 1 1 1
- B= 0 1 0 0
- S_{AB}=A□B/A□B
- Compute Jaccard similarity for Apple and Banana

□ Similarity between 2 strings using Jaccard similarity

□ S=S1□$2/S1□$2

□ S1□$2: No. of common bigrams in String S1 and S2

□ S1□$2: Total no. of bigrams in String S1 and S2 excluding duplicates

□ Example

□ S1= night

□ S2=nacht

□ X={ni,ig,gh,ht}

□ Y={na,ac,ch,ht}

□ Intersection is 1(ht is common) union is 7

□ Similarity B/W S1 and S2=1/7=0.14

Question

- Calculate the Jaccard similarity between
- S1= shashi
- S2= ashish

Documents to sets

- How do we apply this set machinery to documents?
- Bag of words model: Here each document is treated as an unordered set of words.
- Shingling approach or Shingles: A k-shingle is a consecutive set of k words. So the set of all 1-shingles is exactly the bag of words model. An alternative name to k-shingle is an k-gram. These mean the same thing

- D1 : I am Sam.
- D2 : Sam I am.
- D3 : I do not like green eggs and ham.
- D4 : I do not like them, Sam I am.
- The (k = 1)-shingles of D1UD2UD3UD4 are: {[I], [am], [Sam], [do], [not], [like], [green], [eggs], [and], [ham], [them]}.

- The (k = 2)-shingles of D1UD2UD3UD4 are:
- {[I am], [am Sam], [Sam Sam], [Sam I], [am I], [I do], [do not], [not like], [like green], [green eggs], [eggs and], [and ham], [like them], [them Sam]}.
- Character level. We can also create k-shingles at the character level. The (k = 3)-character shingles of D1 U D2 are:
- {[iam], [ams], [msa], [sam], [ami], [mia]}.
- The (k = 4)-character shingles of D1UD2 are:
- {[iams], [amsa], [msam], [sams], [sami], [amia], [miam]}.

- Jaccard with Shingles So how do we put this together. Consider the **(k = 2)-**shingles for each D1, D2, D3, and D4:
- **D1 : I am Sam.**
- **D2 : Sam I am.**
- **D3 : I do not like green eggs and ham.**
- **D4 : I do not like them, Sam I am.**
- **Now K=2 shingles-→**
- **D1 : [I am], [am Sam]**
- **D2 : [Sam I], [I am]**
- **D3 : [I do], [do not], [not like], [like green], [green eggs], [eggs and], [and ham]**
- **D4 : [I do], [do not], [not like], [like them], [them Sam], [Sam I], [I am]**

- Now compute the Jaccard similarity b/w

Q1. (D1,D2)

Q2. (D1,D4)

Q3. (D3,D4)

- JS(D1, D2) = 1/3 ≈ 0.333
- JS(D1, D4) = 1/8 = 0.125
- JS(D3, D4) = 2/7 ≈ 0.286
- JS(D3, D4) = 3/11 ≈ 0.273

What is a Document?

- Examples:
  - web pages, email, books, news stories, scholarly papers, text messages, Word™, Powerpoint™, PDF etc.
- Common properties
  - Significant text content
  - Some structure (e.g., title, author, date for papers; subject, sender, destination for email)

Calculating similarity

- Consider a case insensitive query and document collection with a query **Q and a** document collection consisting of the following three documents:
- Q: *"gold silver truck"*
- *D1: "Shipment of gold damaged in a fire"*
- *D2: "Delivery of silver arrived in a silver truck"*
- *D3: "Shipment of gold arrived in a truck"*

Cosine similarity

- Cosine similarity is the cosine of the angle between two *n*-dimensional vectors in an *n*-dimensional space. It is the dot product of the two vectors divided by the product of the two vectors' lengths (or magnitudes).

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- Values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar.

- Use-cases - when to use the Cosine Similarity algorithm
- We can use the Cosine Similarity algorithm to work out the similarity between two things. We might then use the computed similarity as part of a recommendation query. For example, to get movie recommendations based on the preferences of users who have given similar ratings to other movies that you've seen.

Calculate Cosine similarity

- For vectors
- (3,8,7,5,2,9) & (10,8,6,6,4,5)

- These two lists of numbers have a Cosine similarity of **0.863**. We can see how this result is derived by breaking down the formula:

$$\left(similarity(A,B) = \frac{3\cdot 10+8\cdot 8+7\cdot 6+5\cdot 6+2\cdot 4+9\cdot 5}{\sqrt{3^2+8^2+7^2+5^2+2^2+9^2} \; x \; \sqrt{10^2+8^2+6^2+6^2+4^2+5^2}} = \frac{219}{15.2315 \, x \, 16.6433} = 0.8639 \right)$$

- Here are two very short texts to compare:
  - Julie loves me more than Linda loves me
  - Jane likes me more than Julie loves me
- We want to know how similar these texts are, purely in terms of word counts (and ignoring word order).

- We begin by making a list of the words from both texts:
- **me Julie loves Linda than more likes Jane**

Now we count the number of times each of these words appears in each text:

me	2	2
Jane	0	1
Julie	1	1
Linda	1	0
likes	0	1
loves	2	1
more	1	1
than	1	1

- We are not interested in the words themselves though. We are interested only in those two vertical vectors of counts.

Vectors

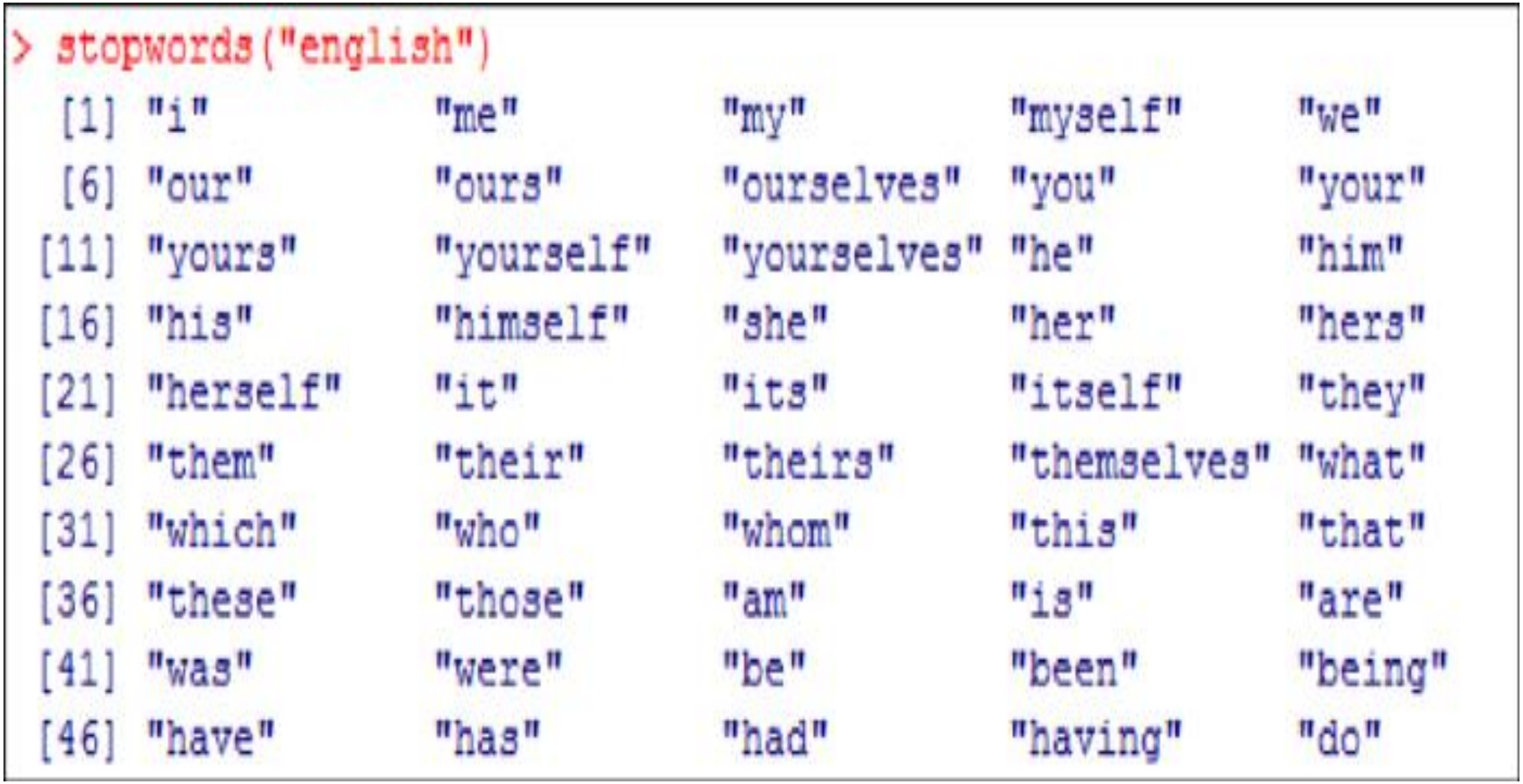
- The two vectors are, again:
- a: [2, 0, 1, 1, 0, 2, 1, 1]
- b: [2, 1, 1, 0, 1, 1, 1, 1]

- The cosine of the angle between them is about **0.822.**
- These vectors are 8-dimensional. A virtue of using cosine similarity is clearly that it converts a question that is beyond human ability to visualise to one that can be. In this case you can think of this as the angle of about **35** degrees which is some 'distance' from zero or perfect agreement.

Cosine Similarity

- □ Sentence1➔ word1 word2 word3 word1 word2 word1
- □ Sentence2→ word1 word5 word2 word3 word5 word1
- □ Consider list of sentences which contains only 10 words.
- □ Compute the Cosine similarity between Sentence 1 and Sentence 2

Stopwords in English



```
> stopwords("english")
 [1] "i"          "me"         "my"         "myself"     "we"
 [6] "our"        "ours"       "ourselves"  "you"        "your"
[11] "yours"      "yourself"   "yourselves" "he"         "him"
[16] "his"        "himself"    "she"        "her"        "hers"
[21] "herself"    "it"         "its"        "itself"     "they"
[26] "them"       "their"      "theirs"     "themselves" "what"
[31] "which"      "who"        "whom"       "this"       "that"
[36] "these"      "those"      "am"         "is"         "are"
[41] "was"        "were"       "be"         "been"       "being"
[46] "have"       "has"        "had"        "having"     "do"
```

Computation (TF)

- □ Need to compute Term Weighting
- □ Term Frequency (tf_{ij})
- □ It may be defined as the number of occurrences of $\mathbf{w}_i$ in $\mathbf{d}_j$. The information that is captured by term frequency is how salient a word is within the given document or in other words we can say that the higher the term frequency the more that word is a good description of the content of that document.

Formula :

tf(t,d) = count of t in d / number of words in d

- TF = {number of times the term appears in the document} /{total number of terms in the document}

- Imagine the term t appears 20 times in a document that contains a total of 100 words. Term Frequency (TF) of t can be calculated as follow:

- **TF=20/100=0.2**

-

Terminology :

- **Terminology :**
- t — term (word)
- d — document (set of words)
- N — count of corpus

- Document Frequency (df_i)
- It may be defined as the total number of documents in the collection in which w_i occurs. It is an indicator of informativeness. Semantically focused words will occur several times in the document unlike the semantically unfocused words.

- ***df(t) = occurrence of t in documents***

- Inverse Document Frequency (idf)
- This is another form of document frequency weighting and often called idf weighting or inverse document frequency weighting. The important point of idf weighting is that the term's scarcity across the collection is a measure of its importance and importance is inversely proportional to frequency of occurrence.
- ***idf(t) = log(N/(df + 1))***

- IDF = log ({number of the documents in the corpus}} /{number of documents in the corpus containing the term})

- Assume a collection of related documents contains 10,000 documents. If 100 documents out of 10,000 documents contain the term t, Inverse Document Frequency (IDF) of t can be calculated as follows
- IDF=log10000/100=2
-

- tf-idf now is a the right measure to evaluate how important a word is to a document in a collection or corpus.

- ***tf-idf(t, d) = tf(t, d) * log(N/(df + 1))***

- Using these two quantities, we can calculate TF-IDF score of the term t for the document.

- $\text{TF-IDF}=0.2*2=0.4$

Question

□ Consider a document containing 100 words wherein the word cat appears 3 times.

□ Calculate the Term Frequency?

- The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03.

- Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, Compute the inverse document frequency (i.e.,

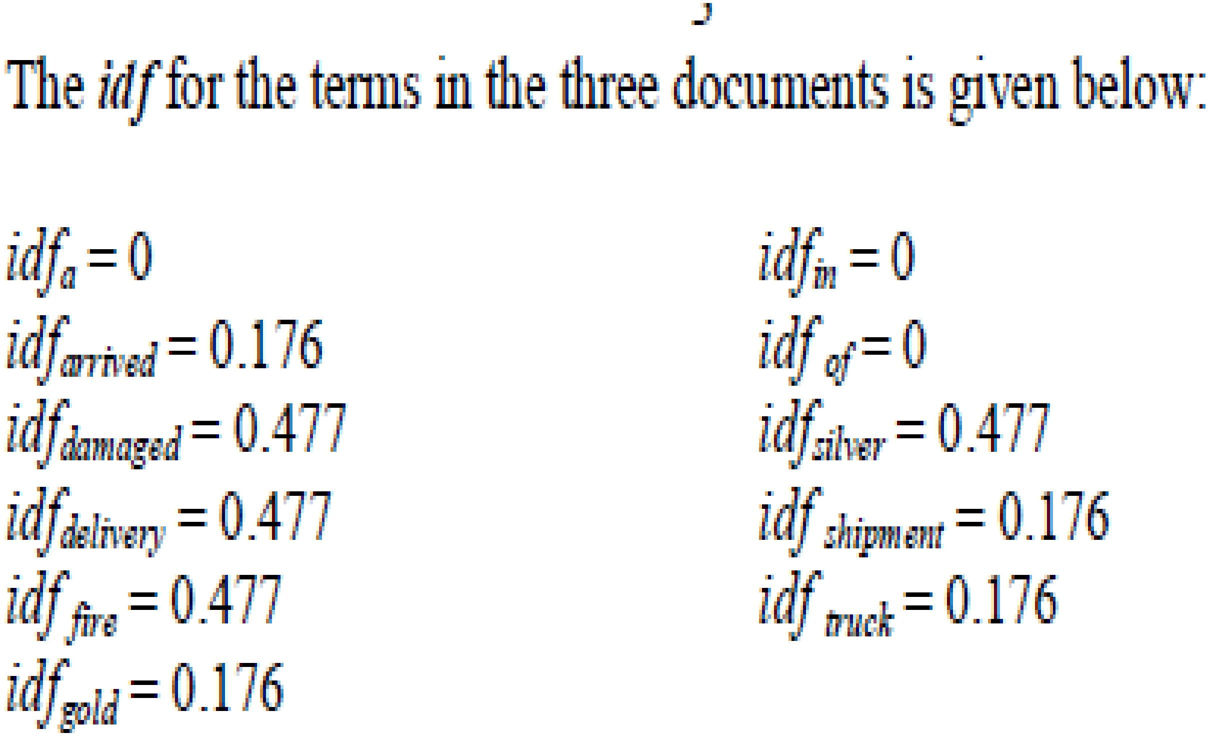
- Idf is calculated as log(10,000,000 / 1,000) = 4.

- Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

Calculating similarity: Similarity Coefficent Using Inner Product

- Consider a case insensitive query and document collection with a query **Q and a** document collection consisting of the following three documents:
- Q: *"gold silver truck"*
- *D1: "Shipment of gold damaged in a fire"*
- *D2: "Delivery of silver arrived in a silver truck"*
- *D3: "Shipment of gold arrived in a truck"*

Calculating Idf for each term



The *idf* for the terms in the three documents is given below:

$idf_{a} = 0$

$idf_{arrived} = 0.176$

$idf_{damaged} = 0.477$

$idf_{delivery} = 0.477$

$idf_{fire} = 0.477$

$idf_{gold} = 0.176$

$idf_{in} = 0$

$idf_{of} = 0$

$idf_{silver} = 0.477$

$idf_{shipment} = 0.176$

$idf_{truck} = 0.176$

Vector generation

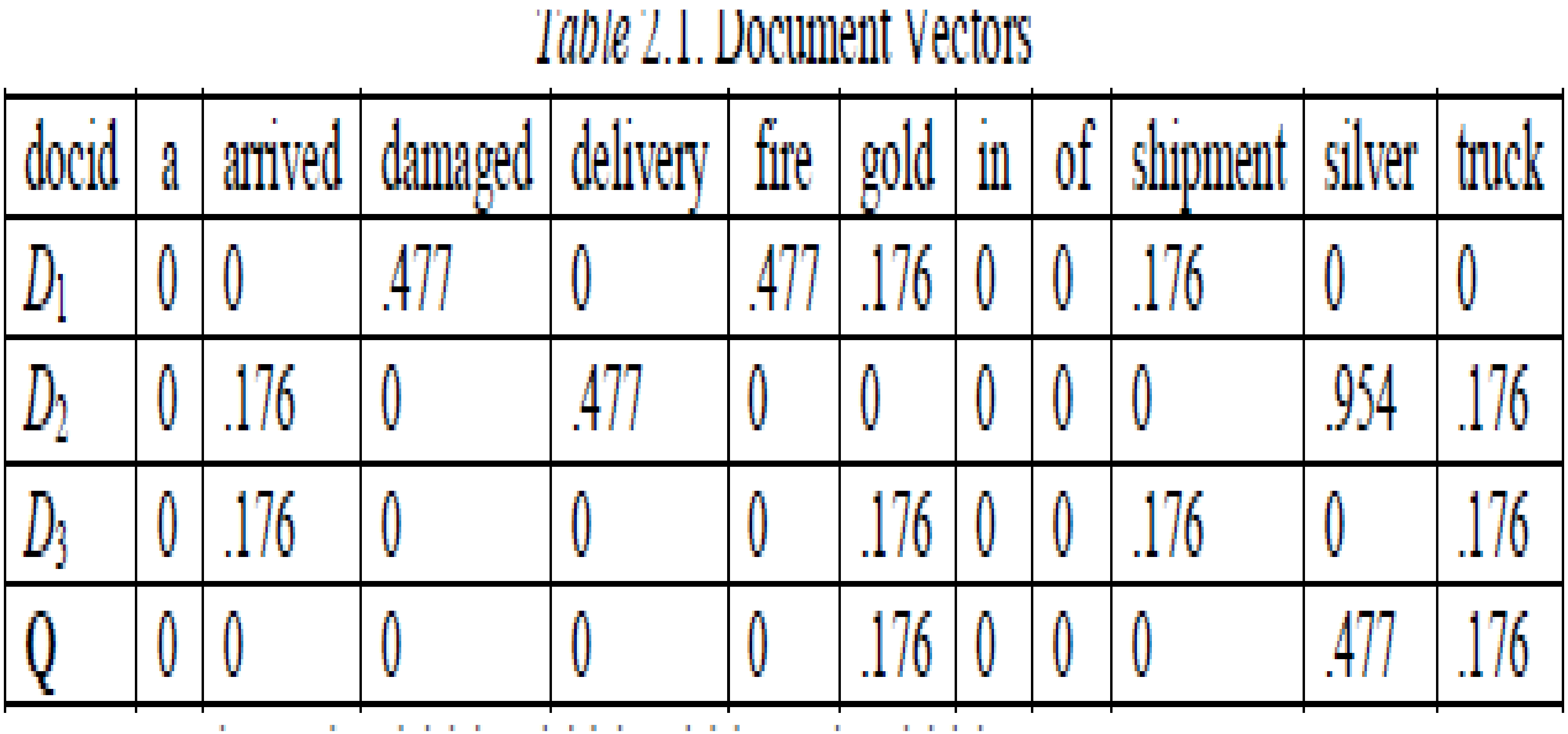
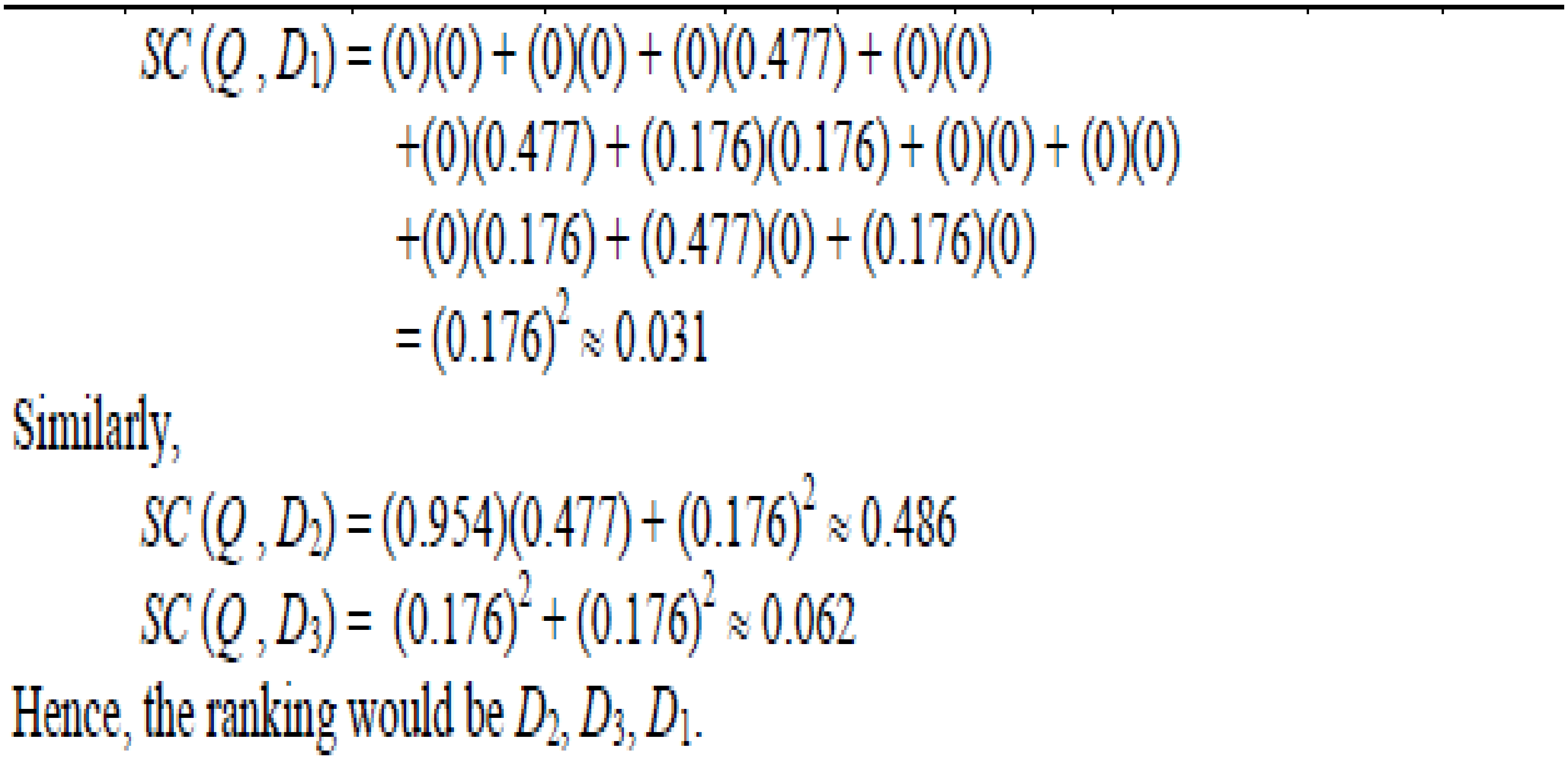


Table 2.1. Document Vectors

docid	a	arrived	damaged	delivery	fire	gold	in	of	shipment	silver	truck
D_1	0	0	.477	0	.477	.176	0	0	.176	0	0
D_2	0	.176	0	.477	0	0	0	0	0	.954	.176
D_3	0	.176	0	0	0	.176	0	0	.176	0	.176
Q	0	0	0	0	0	.176	0	0	0	.477	.176

Similarity Calculation



$$SC(Q, D_1) = (0)(0) + (0)(0) + (0)(0.477) + (0)(0)$$
$$+(0)(0.477) + (0.176)(0.176) + (0)(0) + (0)(0)$$
$$+(0)(0.176) + (0.477)(0) + (0.176)(0)$$
$$= (0.176)^2 \approx 0.031$$

Similarly,

$$SC(Q, D_2) = (0.954)(0.477) + (0.176)^2 \approx 0.486$$
$$SC(Q, D_3) = (0.176)^2 + (0.176)^2 \approx 0.062$$

Hence, the ranking would be D_2, D_3, D_1.

Question

□ Consider the following documents:

□ D1: Shipment of gold damaged in a fire

□ D2: Delivery of silver arrived in a silver truck

□ D3: Shipment of gold arrived in a truck

□ and the following set of terms: T = {fire, gold,silver,truck}.

□ Compute, using the boolean model, what documents satisfy the query

□ (fire OR gold) AND (truck OR NOT silver)

Compute TF and IDF

- first_sentence : “Data Science is the emerging job of the 21st century”.
- second_sentence : “machine learning is the key for data science”.

Question

- □ Given the following query: **"new new times"**, we calculate the *tf-idf* vector for the query, and compute the score of each document in C relative to this query, using the cosine similarity measure.
- □ D1: :new york times"
- □ D2: new york post:
- □ D3: "los angeles times"
- □ **Q: "new new times"**

Question

- Calculate the tf and idf score use log Base 2 for Idf

TF- Term Frequency Score

- “Tf score”

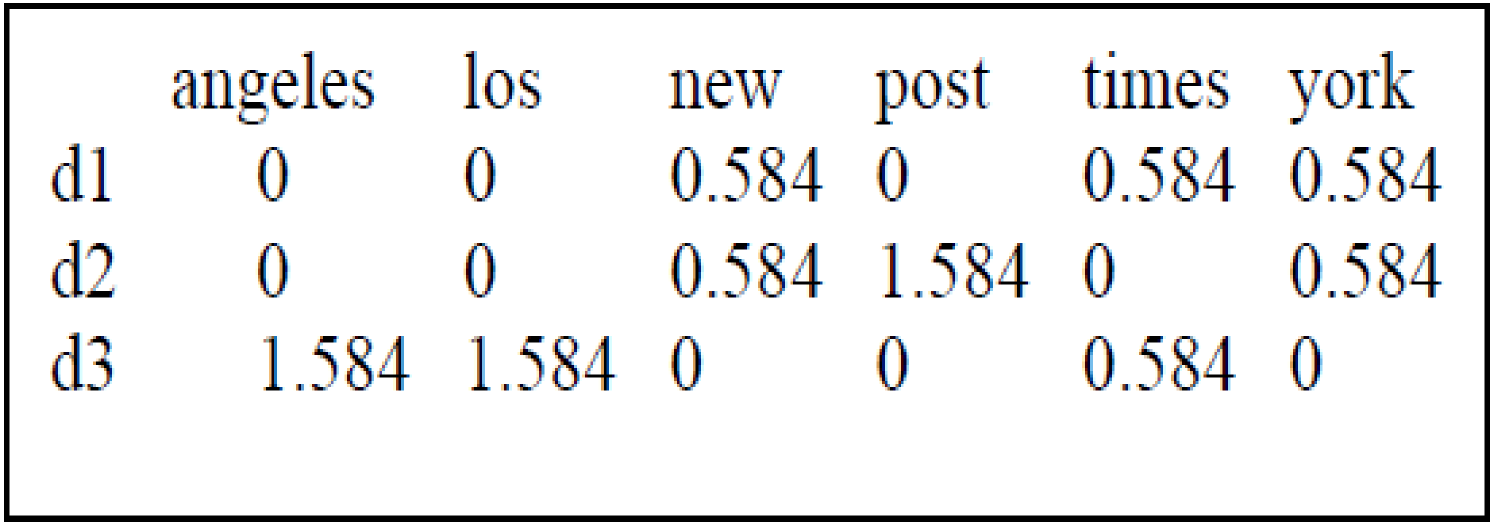
	angeles	los	new	post	times	york
D1	0	0	1	0	1	1
D2	0	0	1	1	0	1
D3	1	1	0	0	1	0

Idf score: log base2

angles log2(3/1)=1.584

- □ los log2 (3/1)=1.584
- □ new log2 (3/2)=0.584
- □ post log2 (3/1)=1.584
- □ times log2 (3/2)=0.584
- □ york /log2 (3/2)=0.584

Idf score: log base2



	angeles	los	new	post	times	york
d1	0	0	0.584	0	0.584	0.584
d2	0	0	0.584	1.584	0	0.584
d3	1.584	1.584	0	0	0.584	0

□ **Query 0 0 2*.584 0 .584 0**

Compute similarity coefficient between

- Sim(d1,q) =
- Sim(d2,q) =
- Sim(d3,q) =

Questions

- □ What is meant by cosine similarity?
- □ **What is a good cosine similarity?**
- □ **Can cosine similarity be greater than 1?**
- □ Name 10 similarity measures techniques?
- □ **What is the range of similarity measure?**
- □ **Which is better Jaccard or cosine similarity?**
- □ **Use of Cosine similarity?**

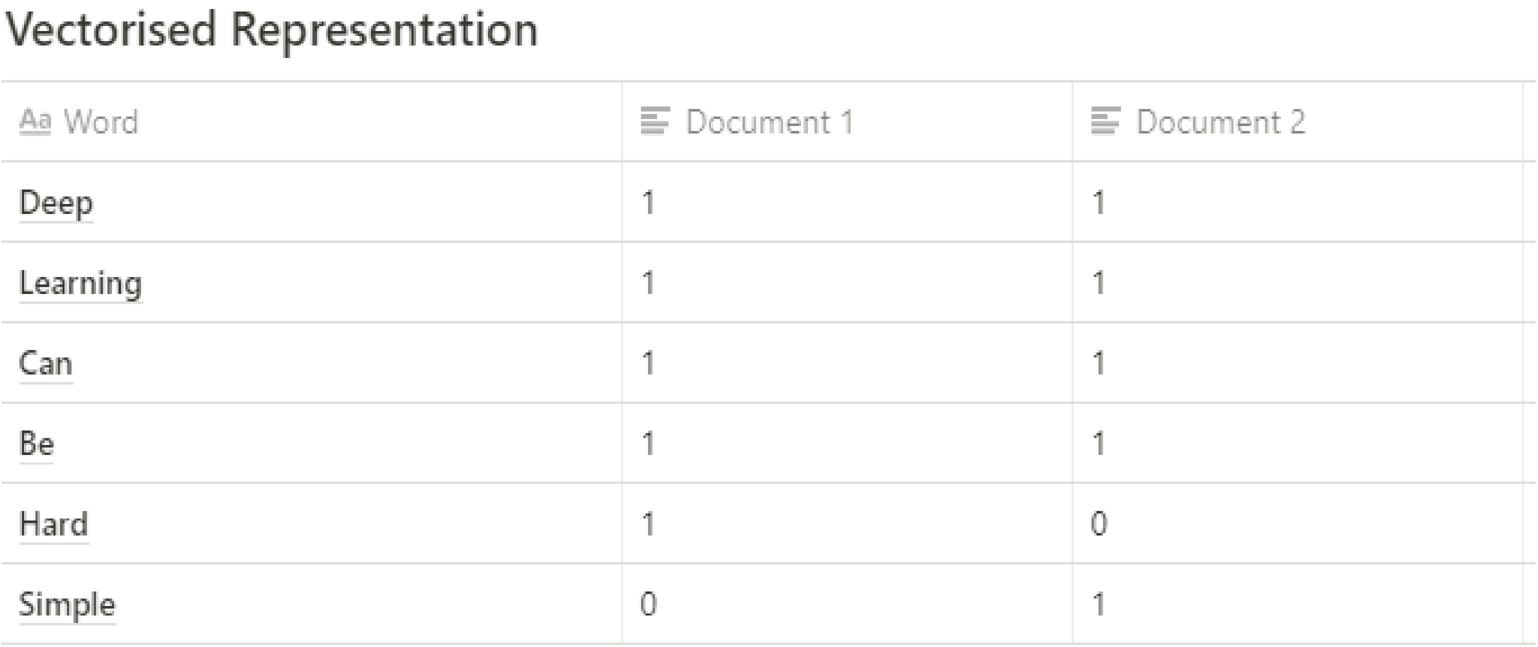
Question

- *Compute the cosine similarity between below two documents*

- *Document 1: Deep Learning can be hard*
- *Document 2: Deep Learning can be simple*

Solution

- □ First obtain Vectors:



Vectorised Representation

Word	Document 1	Document 2
Deep	1	1
Learning	1	1
Can	1	1
Be	1	1
Hard	1	0
Simple	0	1

Solution

- *Document 1: [1, 1, 1, 1, 1, 0] let's refer to this as A*
- *Document 2: [1, 1, 1, 1, 0, 1] let's refer to this as B*

Question

- Recommend a query processing order for
  - (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
- given the following postings list sizes:

Term Postings	size
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
Trees	316812

- **SOLUTION. Using the conservative estimate of the length of unioned**
- postings lists, the recommended order is**: (kaleidoscope OR eyes) (300,321)**
- **AND (tangerine OR trees) (363,465) AND (marmalade OR skies) (379,571)**
- **Time for processing :**

- (i) (tangerine OR trees) = O(46653+316812) = O(363465)
- (ii) (marmalade OR skies) = O(107913+271658) = O(379571)
- (iii) (kaleidoscope OR eyes) = O(46653+87009) = O(300321)
- **Order of processing: a. Process (i), (ii), (iii) in any order as first 3 steps (total**
- **time for these steps is O(363465+379571+300321) in any case)**
- b. Merge (i) AND (iii) = (iv):
- 363465 +300321 =**663786 (iv)**
- In case of AND operator, the complexity of merging postings list depends on the length of the shorter postings list. Therefore, the more short the smaller postings list, the lesser the time spent.
- The reason for choosing (i) instead of (ii) is that the output list (iv) is more probable to be shorter if (i) is chosen.
- c. Merge (iv) AND (ii):
- **663786 +** 379571 = **1043357**
- This is the only merging operation left.
- Merge 1 & 2➔ 363465 + 379571 = **743036** (Large value as compared to merging 1 & 3)
- Merge 2 & 3➔ 379571 300321 = **679892** (Large value as compared to merging 1 & 3)

Multimedia data

- Data combining several different media, such as text, images, sound and video.

Query

- A *query is a request for documents* pertaining to some topic.
- The expression of the user information need in the input language provided by the information system.

Repository

- A physical or digital place where objects are stored for a period of time, from which individual objects can be obtained if they are requested and their terms and conditions are satisfied.

IR Motivation

- Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the user.
- The emphasis is on the retrieval of information as opposed to the retrieval of data.
- An *Information Retrieval (IR) System* attempts to find

  relevant documents to respond to a user’s request.

Measuring Effectiveness

- An algorithm is deemed incorrect if it does not have a “right” answer.
- A heuristic tries to guess something close to the right answer. Heuristics are measured on “how close” they come to a right answer.
- IR techniques are essentially heuristics because we do not know the right answer.
- So we have to measure how *close to the* right answer we can come.

What is a Document?

- Examples:
  - web pages, email, books, news stories, scholarly papers, text messages, Word™, Powerpoint™, PDF etc.
- Common properties
  - Significant text content
  - Some structure (e.g., title, author, date for papers; subject, sender, destination for email)

Documents vs. Database Records

- Database records (or *tuples* in relational databases) are typically made up of well-defined fields (or *attributes*)
  - e.g., bank records with account numbers, balances, names, addresses, social security numbers, dates of birth, etc.
- Easy to compare fields with well-defined semantics to queries in order to find matches
- Text is more difficult

Documents vs. Records

- Example bank database query
  - *Find records with balance > $50,000 in branches located in Amherst, MA.*
  - Matches easily found by comparison with field values of records
- Example search engine query
  - *bank scandals in western group*
  - This text must be compared to the text of entire news stories

Information versus Data Retrieval

- Data retrieval, in the context of an IR system, consists mainly of determining which documents of a collection contain the keywords in the user query which, most frequently, is not enough to satisfy the user information need.
- In fact, the user of an IR system is concerned more with retrieving *information about a* subject than with retrieving data which satisfies a given query.

Information versus Data Retrieval …

- A data retrieval language aims at retrieving all objects which satisfy clearly defined conditions such as those in a regular expression or in a relational algebra expression.
- Data retrieval language works in context to conditions specified by the user and retrieves the data accordingly.

Information versus Data Retrieval …

- The main reason for this difference is that information retrieval usually deals with natural language text which is not always well structured and could be semantically ambiguous.
- On the other hand, a data retrieval system (such as a relational database) deals with data that has a well defined structure and semantics.
-

Information versus Data Retrieval

…

- Data retrieval provide a solution to the user of a data base system . It does not solve the problem of retrieving info. about a subject or topic.
- The IR system somehow interpret the contents of the information items(documents) in a collection and rank them according to a degree of relevance to the user query.
- The primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving a few non relevant documents as possible.
- Data retrieval system aims to retrieve objects which satisfies defined conditions, thus a single erroneous object among a thousand retrieved objects means total failure

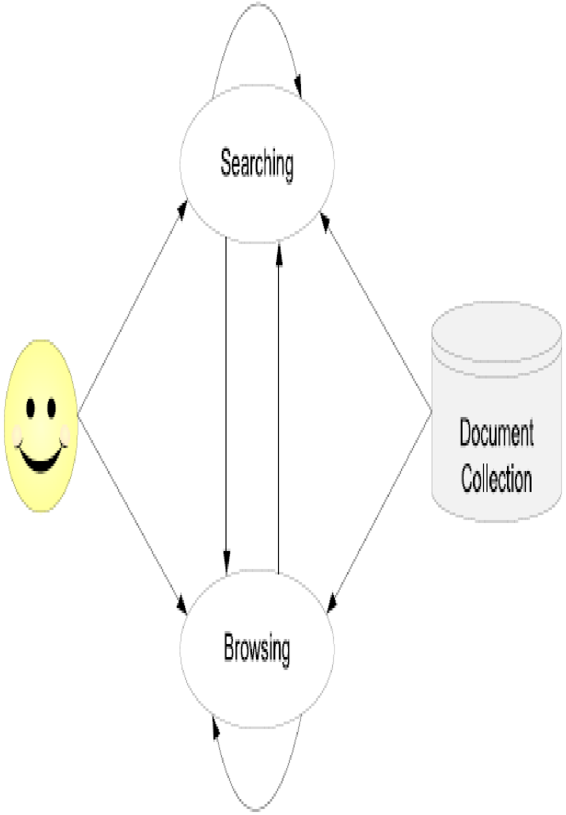
The IR Problem…

The IR Problem

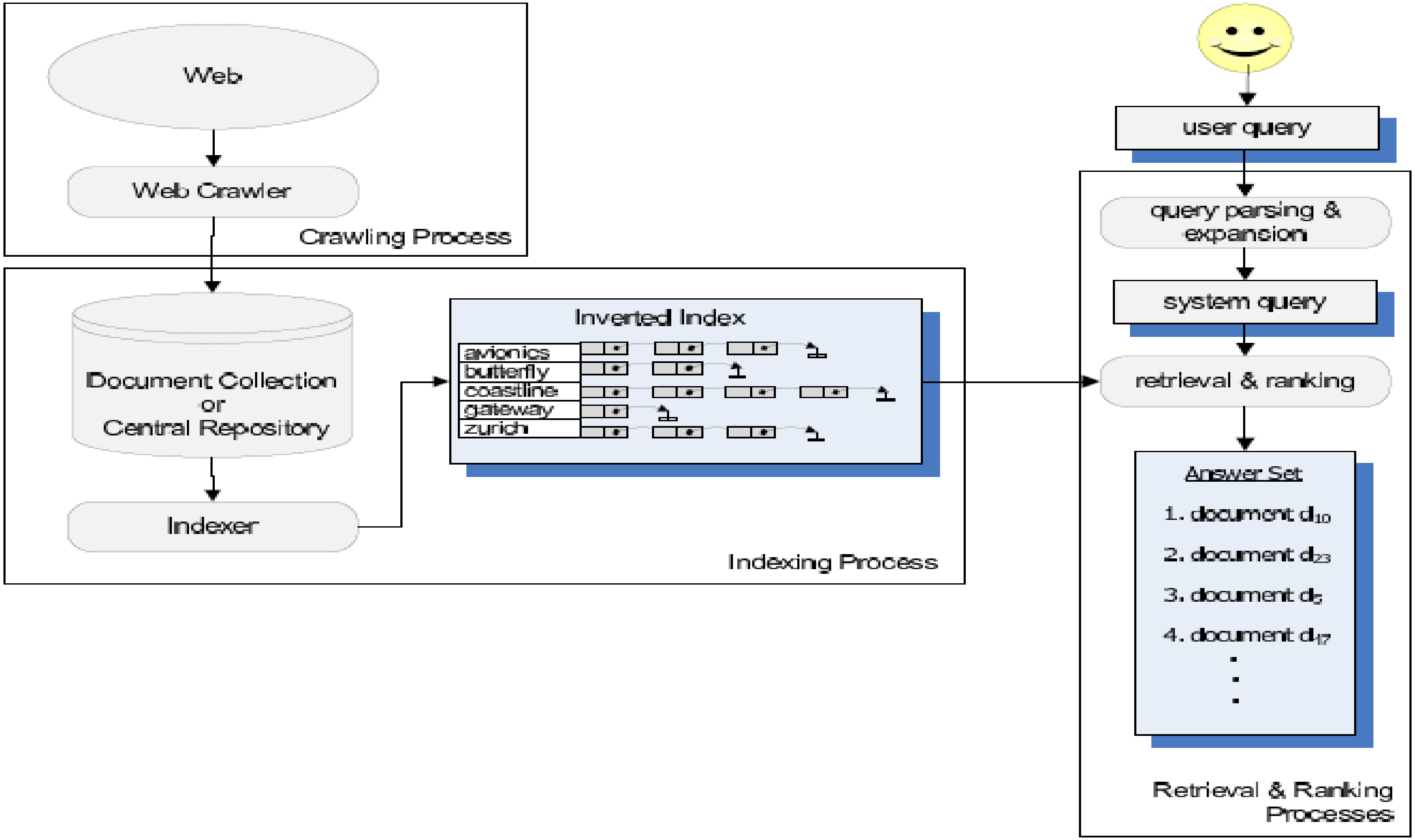
- *The key goal of an IR system is to retrieve all the items that are relevant to a user query, while retrieving as few nonrelevant items as possible*
- The notion of relevance is of central importance in IR

Searching Vs Browsing

- **Consider a user who seeks information on a topic of their interest**
- This user first translates their information need into a query, which requires specifying the words that compose the query
- In this case, we say that the user is *searching or querying for* information of their interest
- **Consider now a user who has an interest that is either poorly defined or characteristic broad**
- For instance, the user has an interest in car racing and wants to browse documents on Formula 1 and Formula Indy
- In this case, we say that the user is *browsing or navigating the* documents of the collection

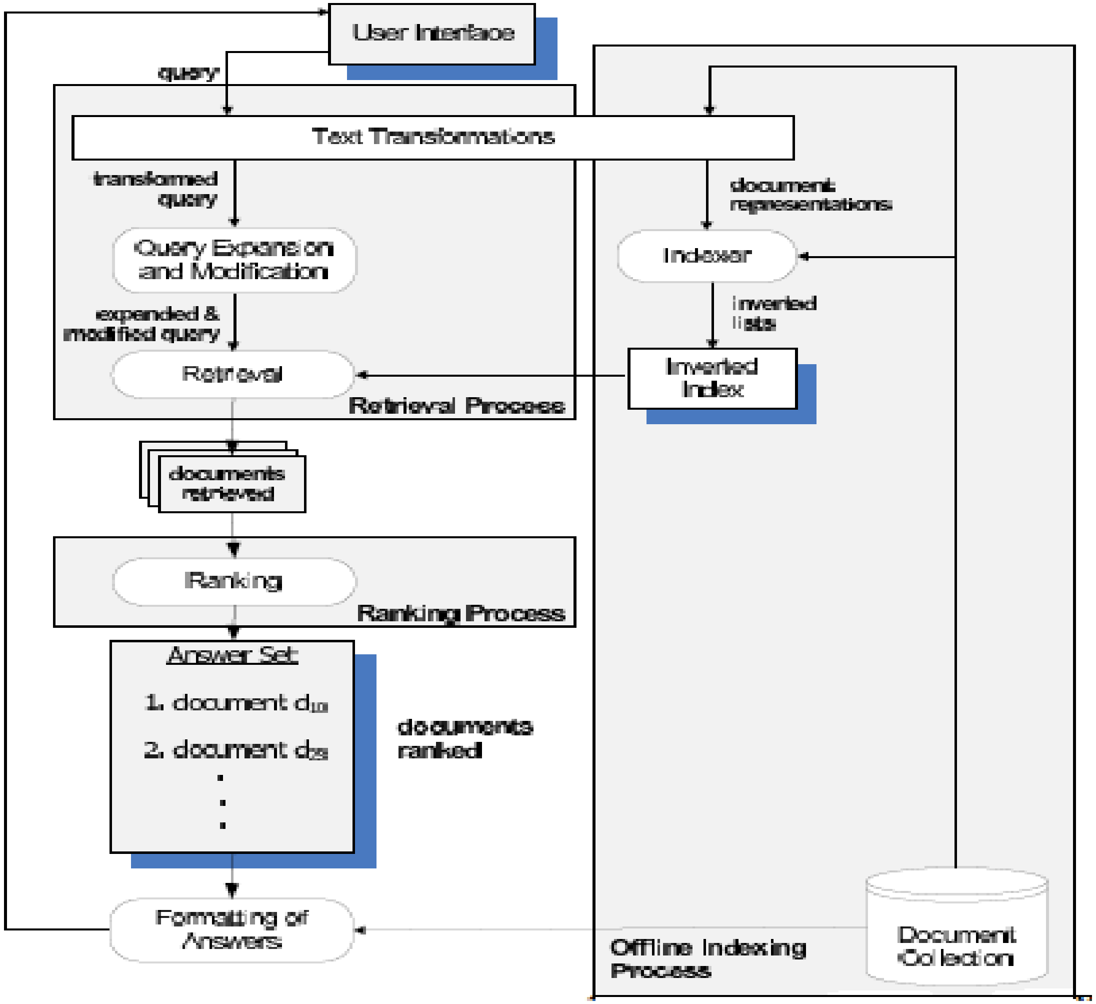


Architecture of the IR System



Retrieval and Ranking Processes

- The processes of *indexing, retrieval, and ranking*



Introduction

- The Web
  - very large, public, unstructured but ubiquitous repository
  - need for efficient tools to manage, retrieve, and filter information
  - search engines have become a central tool in the Web
- Two characteristics make retrieval of relevant information from the Web a really hard task
  - the large and distributed volume of data available
  - the fast pace of change

Introduction

- There are basically three different forms of searching the Web. Two of them are well known and are frequently used.
- The first is to use search engines that index a portion of the Web documents as a full-text database.
- The second is to use Web directories, which classify selected Web documents by subject.
- The third and not yet fully available, is to search the Web exploiting its hyperlink structure.
- **Three different forms**
  - **Search engines**
    - AltaVista
  - **Web directories**
    - Yahoo
  - **Hyperlink search**
    - WebGlimpse

How the Web Changed Search

- Web search is today the most prominent application of IR and its techniques—the ranking and indexing components of any search engine are fundamentally IR pieces of technology
- **The *first major impact of the Web on search is related* to the characteristics of the document collection itself**
  - The Web is composed of pages distributed over millions of sites and connected through hyperlinks
  - This requires collecting all documents and storing copies of them in a central repository, prior to indexing
  - This new phase in the IR process, introduced by the Web, is called *crawling*

How the Web Changed Search…

- **The *second major impact of the Web on search is* related to:**
  - The size of the collection
  - The volume of user queries submitted on a daily basis
  - As a consequence, performance and scalability have become critical characteristics of the IR system
- **The *third major impact: in a very large collection,* predicting relevance is much harder than before**
  - Fortunately, the Web also includes new sources of evidence
  - Ex: hyperlinks and user clicks in documents in the answer set

Scalability is the ability of a system, network, or process, to handle growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth

How the Web Changed Search…

- **The *fourth major impact derives from the fact that the* Web is also a medium to do business**
  - Search problem has been extended beyond the seeking of text information to also encompass other user needs
  - Ex: the price of a book, the phone number of a hotel, the link for downloading a software

Challenges of searching the web

- Main challenges posed by Web are of two types
  - **data-centric: related to the data itself**
  - **interaction-centric: related to the users and their interactions**

  The problems related to the data are:
  - **Distributed Data**
  - **High Percentage of Volatile Data**
  - **Large Volume**
  - **Unstructured and Redundant Data**
  - **Quality of Data**
  - **Heterogeneous Data**

Challenges of Searching the web

- ***Distributed data:*** due to intrinsic nature of web , data spawns over many computers and platforms. These computers are interconnected with no predefined topology and the available bandwidth and reliability on the network interconnections varies widely
- ***High percentage of volatile data:*** due to internet dynamics new computers and data can be added or removed easily.(it is estimated that 40% of the web changes every month[B. Kahle "Archiving the Internet"]

Challenges of Searching the web..

- ***Large Volume :*** the exponential growth of the web poses scaling issues that are difficult to cope with.
- ***Unstructured and redundant data:*** most people say that the Web is a distributed hypertext. However, this is not exactly so. Any hypertext has a conceptual model behind it, which organizes and adds consistency to the data and the hyperlinks.
- That is hardly true in the Web, even for individual documents. In addition, each HTML page is not well structured and some people use the term *semi-structured data. Moreover, much Web* data is repeated (mirrored or copied) or very similar.

Challenges of Searching the web..

- ***Quality of data:*** The web can be considered as new publishing medium with minimum editorial process. So, the data can be false, invalid , poorly written or typically with errors from different sources (like typographical, grammatical mistakes etc.)

- ***Heterogeneous data :*** Web deals with multiple media types and hence with multiple formats and having data in different languages having different alphabets .

Challenges…

- **interaction-centric: related to the users and their interactions**
- The second class of problems are those faced by the user during the interaction with the retrieval system.
- There are basically two problems:
- (1) how to specify a query and
- (2) how to interpret the answer provided by the system.
- Without taking into account the semantic content of a document, it is not easy to precisely specify a query, unless it is very simple …Further, even if the user is able to pose the query, the answer might be a thousand Web pages.
- **User key challenge**
  - to conceive a good query
- **System key challenge**
  - to do a fast search and return relevant answers, even to poorly formulated queries

Practical Issues in the Web

- **Security**
- Commercial transitions over the Internet are not yet a completely safe procedure
- **Privacy**
- Frequently, people are willing to exchange information as long as it does not become public
- **Copyright and patent rights**
- It is far from clear how the wide spread of data on the Web affects copyright and patent laws in the various countries

Web Characteristics

- **The amount of information on the Web is huge,** and easily accessible.
- **The coverage of Web information is very wide and diverse.**--One can find information about almost anything.
- **Information/data of almost all types exist on the Web**, e.g.,structured tables, texts, multimedia data, etc.

Web Characteristics

- **Much of the Web information is semi-structured** due to the nested structure of HTML code.
- **Much of the Web information is linked.** There are hyperlinks among pages within a site, and across different sites.
- **Much of the Web information is redundant.** The same piece of information or its variants may appear in many pages.

Web Characteristics

- **The Web is noisy.** A Web page typically contains a mixture of many kinds of information, e.g., main contents,advertisements, navigation panels, copyright notices, etc.
- **The Web consists of surface Web and deep Web.**
- **Surface Web:** pages that can be browsed using a browser.
- **Deep Web:** databases that can only be accessed through parameterized query interfaces.
- **The Web is also about services.** Many Web sites and pages

enable people to perform operations with input parameters,

i.e., they provide services.

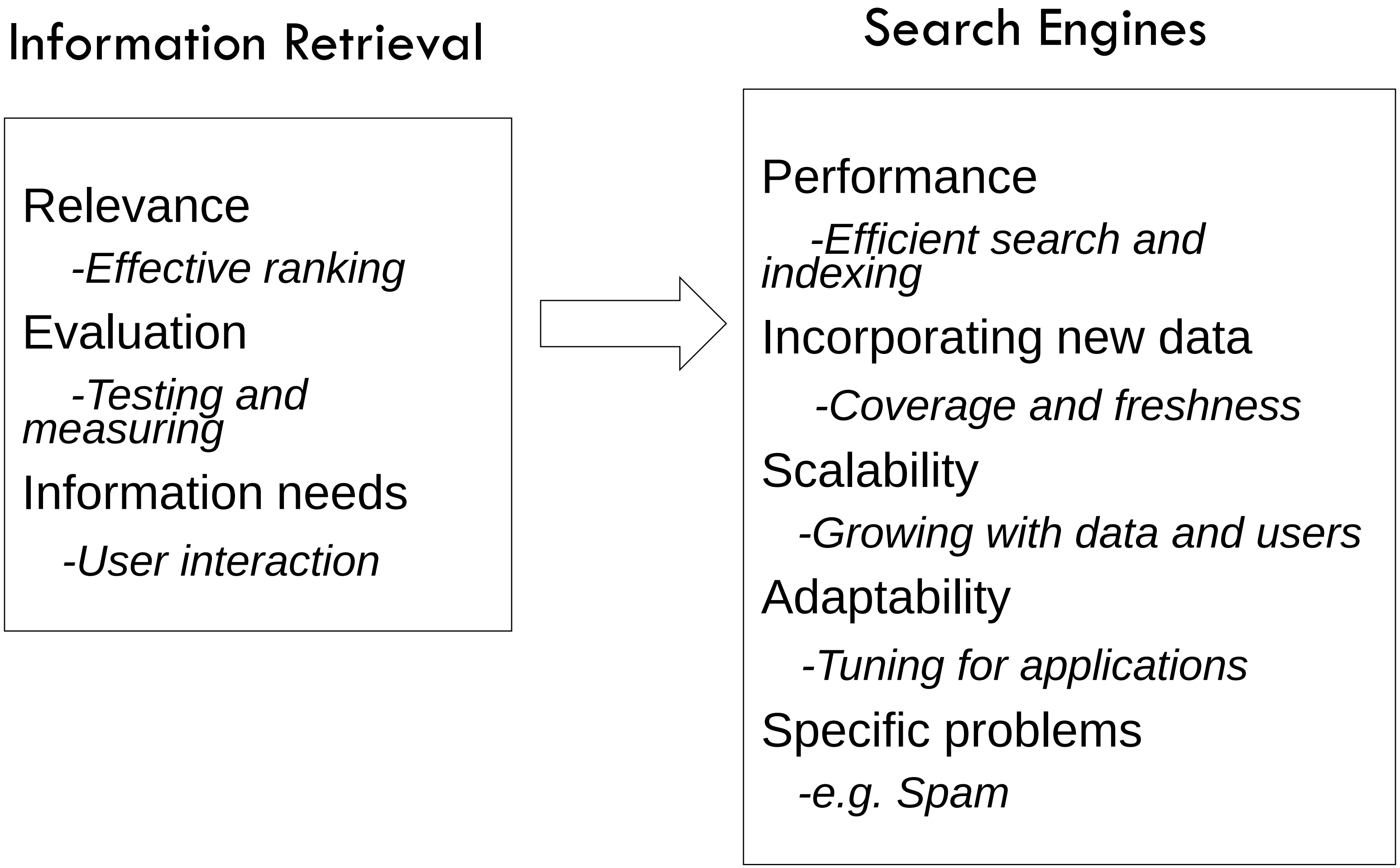
Web Characteristics

- **The Web is dynamic.** Information on the Web changes constantly. Keeping up with the changes and monitoring the changes are important issues.
- **Above all, the Web is a virtual society.** It is not only about data, information and services, but also about interactions among people, organizations and automatic systems, i.e., communities.

IR and Search Engines

- A search engine is the practical application of information retrieval techniques to large scale text collections
- Web search engines are best-known examples, but many others
  - *Open source* search engines are important for research and development
    - e.g., Lucene, Lemur/Indri, *Galago*

IR and Search Engines



Information Retrieval

Relevance

-Effective ranking

Evaluation

-Testing and measuring

Information needs

-User interaction

Search Engines

Performance

-Efficient search and indexing

Incorporating new data

-Coverage and freshness

Scalability

-Growing with data and users

Adaptability

-Tuning for applications

Specific problems

-e.g. Spam

Search Engine Issues

- Performance
  - Measuring and improving the efficiency of search
    - e.g., reducing *response time*, increasing *query throughput*, increasing *indexing speed*
  - *Indexes* are data structures designed to improve search efficiency
    - designing and implementing them are major issues for search engines

Search Engine Issues

- Dynamic data
  - The “collection” for most real applications is constantly changing in terms of updates, additions, deletions
    - e.g., web pages
  - Acquiring or “crawling” the documents is a major task
    - Typical measures are *coverage* (how much has been indexed) and *freshness* (how recently was it indexed)
  - Updating the indexes while processing queries is also a design issue

Search Engine Issues

- Scalability
  - Making everything work with millions of users every day, and many terabytes of documents
  - Distributed processing is essential
- Adaptability
  - Changing and tuning search engine components such as ranking algorithm, indexing strategy, interface for different applications