

Web Programming and Development (2505411)

Instructor: Dr. Mohamed Salah Hamdi

Office Phone: 44619521

Email: mshamdi@abmmc.edu.qa

Office Hours: Sun/Tue 09:30-10:30

Lectures: Mon/Wed (Thurs: 11:00-12:15)

Prerequisites: Computer Information Systems
(2505223)

Lab Instructor:

Web Programming and Development

Dr. Mohamed Salah Hamdi

PART1:

Introduction

1

Motivation

- What is the WEB?



- The World Wide Web or short WWW is a system of interlinked hypertext documents that are accessed via the Internet.
- With a web browser, a user can view web pages that can contain text, images, videos, and various other media and navigate between them by clicking on hyperlinks that take him from one page to another.

2

Motivation

- The Web was established in 1989 by the British researcher Tim Berners-Lee, who was working at the European Organization for Nuclear Research (CERN) in Switzerland, and was launched in 1992.
- How does the web work?
 - Viewing a web page typically begins with typing the page address (URL: Uniform Resource Locators) into a web browser or by clicking a hyperlink to that page.

3

Motivation

- After that the web browser sends a set of messages (in the background) to fetch and display the page:
 - First, the part of the URL that represents the name of the web server is converted into an IP address using the Internet's publicly distributed database called (DNS: domain name system). An IP address is necessary to connect to a web server and send data packets to it.
 - The web browser then requests the resource by sending an HTTP request to the web server at that specific address.



4

Motivation


- When the resource is a regular webpage, the HTML text is requested first and parsed directly by the web browser, which then makes additional requests for the images and files that are part of that page.
 - When the web browser has obtained all the necessary files from the web server, the web browser displays the page on the screen as told by the page's HTML code. All the images and other resources are combined to make up the page that the user sees on the screen.
- Most web pages contain hyperlinks to other pages and possibly to downloads, drivers and other resources on the web. This useful collection of interlinked resources was called the **Web**.

5

Motivation

- URL: Uniform Resource Locators

`http://www.starbuzzcoffee.com/index.html`


Protocol Domain Path/Webpage

Protocol: Usually either `http://` or `https://`, this tells the web browser to expect a web address to follow. Modern web browsers don't require you to type the protocol; it will fill that in on its own.

Domain: This is the highest-level part of a URL — the website's name — and you can think of it as the computer on which the webpage is stored. In reality, the domain is probably made up of many computers, especially for large domains accessed by many people.

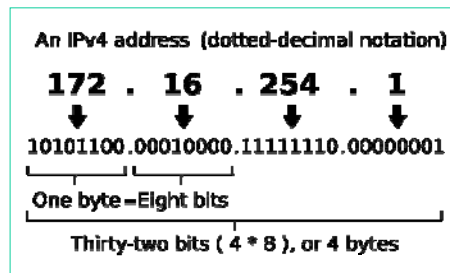
Path: Think of this as the folder structure of the website, so a browser knows which subfolder to find the webpage in.

Webpage: This is the last part of the URL and is the specific page you are requesting. It's generally the actual filename of the page as it is stored on the domain computer.

6

Motivation

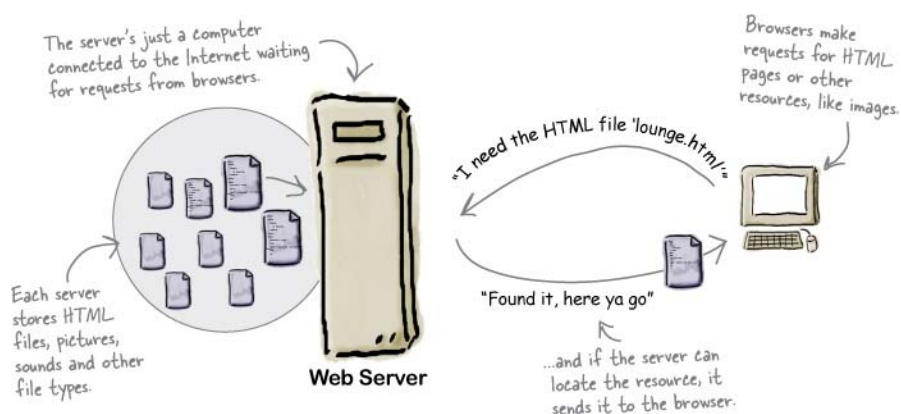
- **IP address (Internet Protocol address):**
 - It is a numerical identification of the devices that participate in a network of computers that uses the Internet protocol to communicate between its nodes.
 - Although IP addresses are stored as binary numbers, they are usually written in a human-readable format, such as: 172.16.254.1



7

Motivation

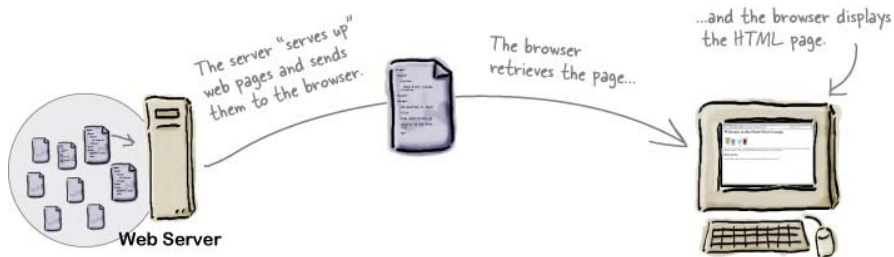
- What does a web server do?



8

Motivation

- What does a web browser do?



9

Motivation

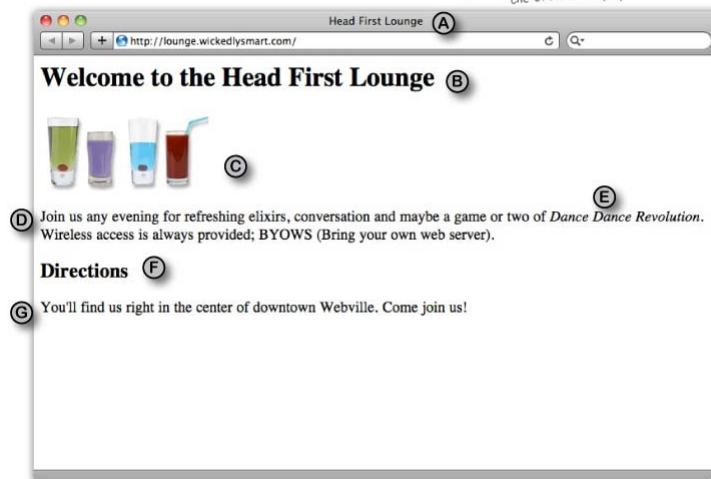
HTML code:

```
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    
    <p>
      Join us any evening for refreshing elixirs,
      conversation and maybe a game or
      two of <em>Dance Dance Revolution</em>.
      Wireless access is always provided;
      BYOWS (Bring your own web server).
    </p>
    <h2>Directions</h2>
    <p>
      You'll find us right in the center of downtown Webville.
      Come join us!
    </p>
  </body>
</html>
```

10

Motivation

- What does a web browser create?



Motivation

- Are **web** and **internet** the same thing?
 - Internet:
 - The word **internet** consists of the prefix **inter** which means “between” and the word **net** which means “network”. So, this name denotes the structure of the Internet as a “**network of networks**” or “**interconnected networks**”. However, a mistake has spread in some media naming it the "international network of information", thinking that the **inter** prefix in the name is an abbreviation of the word "international".

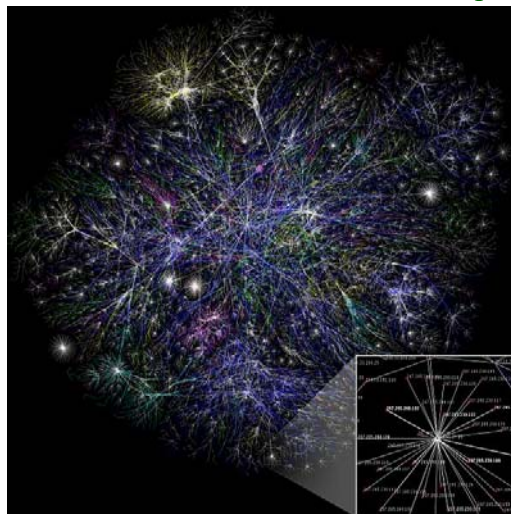
Motivation

- And as indicated by its name, the **internet** is a network between several networks, each of which is managed in isolation from the others in a decentralized manner, and none of them depends on the others for its operation. Different computer and network technologies may be used internally in each of these networks, and what unites them is that these networks communicate with each other through gateways that connect them to a standard common protocol, the **Internet Protocol**.

13

Motivation

- A representation of a network of roads in a small part of the internet:



14

Motivation

- However, in the current era, the vast majority of the networks that make up the Internet use the Internet Protocol internally, due to its technical advantages and the accumulated experience in its operation and maintenance, as well as due to the prevalence of hardware and operating systems that implement and support this protocol.
- The Internet provides **many services** such as the World Wide Web (WWW), communication technologies (chatting), e-mail, and file transfer protocols (FTP).
- So the Web is not the Internet, but rather **a service provided by the Internet!**

15

Motivation

- What's on the web? **Information**
- What do we do with it?
 - **We produce it:** we formulate it, store it and disseminate it
 - This is what we focus more on in the practical part of this course
 - **We use it:** we retrieve it (search for) to make use of it
 - This is what we focus more on in the theoretical part of this course

16

Motivation

- Information retrieval (IR) deals with:
 - the representation of,
 - Storage of,
 - organization of, and
 - access toinformation items.
- Aim of an IR system:
provide the user with easy access to the information in which he is interested.

17

Motivation

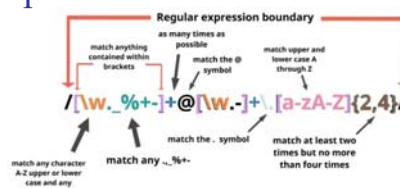
- Characterization of the *user information need* is not a simple problem. Example:
 - Find all docs containing information on college tennis teams which:
 - (1) are maintained by a USA university and
 - (2) participate in the NCAA tournament.
- This description cannot be used directly to request information using the current interfaces of Web search engines:
information need must be transformed into a *query*.
- Most common form of a query:
a set of keywords (or index terms) that summarizes the description of the user information need.

18

Data versus Information Retrieval

- Data retrieval system:
 - Task: which docs contain a set of keywords?
 - Example: *relational database* system.
 - Retrieves all objects which satisfy clearly defined conditions such as those in a *regular expression* or in a relational algebra expression.

Relational database: Is a collection of data items organized as a set of defined tables so that the data can be accessed or regrouped in different ways without having to reorganize the database tables. The relational database was invented in 1970 by E. F. Codd at IBM.



Regular Expression: Is a language used in text processing and searching for words and phrases that have a specific structure, such as e-mail addresses, web addresses, etc.

19

Data versus Information Retrieval

- A single erroneous object among a thousand retrieved objects means failure.
- Deals with data that has a *well defined* structure and semantics.

"Well-defined": This word is used to indicate that a concept or object (function, property, relation, ...) is defined in a way that is unambiguous and without contradiction.

20

Data versus Information Retrieval

- Information retrieval system:
 - Task: get information about a subject or topic.
 - Example: Web search engine.
 - Retrieved objects might be inaccurate (errors are unavoidable and tolerated).
 - Deals with natural language text which is:
 - Not always well structured.
 - Can be semantically ambiguous: “Apple” (company vs. fruit).

21

Data versus Information Retrieval

- Information retrieval system:
 - Must ‘interpret’ the contents of the information items (documents) in a collection and **rank** them according to a **degree of relevance** to the user query.
 - Interpretation of a document content involves:
 - Extracting syntactic and semantic information from the document text.
 - Using this information to match the user information need.
 - The notion of **relevance** is very important in IR.
 - Primary goal of an IR system: retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible.

22

Data versus Information Retrieval

Data: It is a set of numbers, symbols and letters stored within the computer in a way that allows the computer to process it. Data is the raw material on which information is based. Information is classified based on the existence of data. This name comes from the Greek name (datum), which is singular, and the plural is data. In fact, the word “data” is used for the plural and singular.

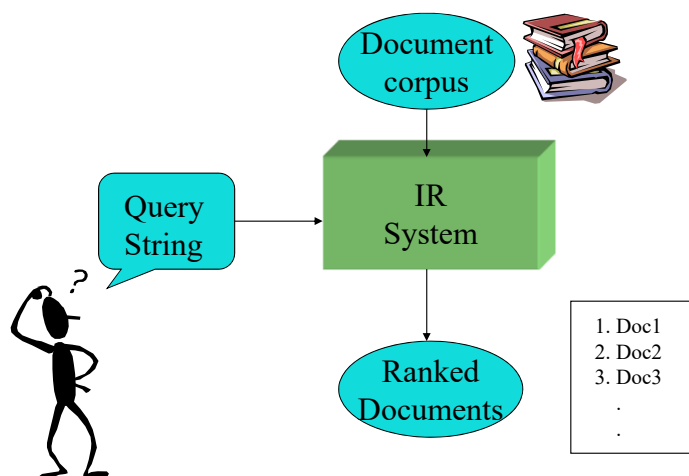
Information: Information is the outcome of processing, modifying, and organizing data in a way that makes it add something to the knowledge of the person who receives it.

Knowledge: It is a certain amount of information that has been extracted, filtered, and prepared in a very special way. Specifically, the information that we call knowledge is information that has been confirmed after examinations have been conducted on it. Common sense knowledge is information that has been established through common sense experience. Scientific knowledge is information that has been established through rules and tests applied by scientists of the appropriate specialty ...

Wisdom: It is the application of knowledge that has been formulated in the form of principles to reach balanced and far-sighted decisions regarding conflicting situations.

23

IR System



24

Relevance

- Relevance is a subjective judgment and may include:
 - Being on the proper subject.
 - Being timely (recent information).
 - Being authoritative (from a trusted source).
 - Satisfying the goals of the user and his/her intended use of the information (*information need*).

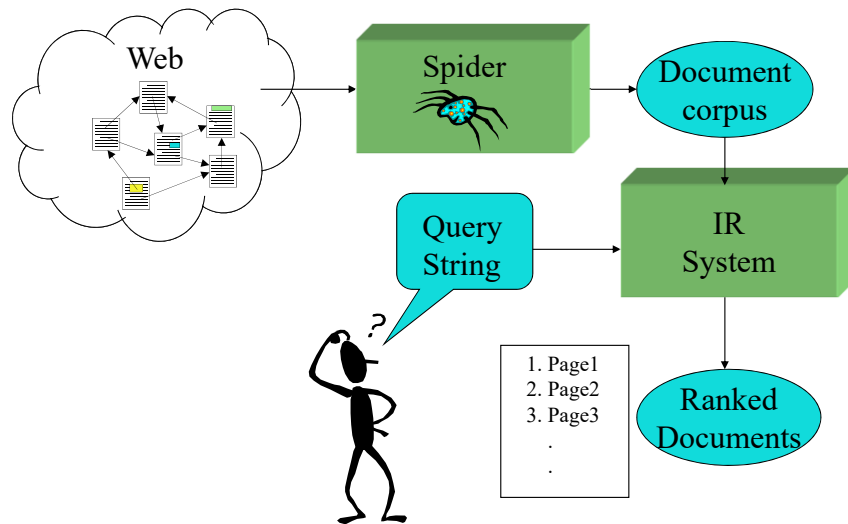
25

Web Search

- Application of IR to HTML documents on the World Wide Web.
- Differences:
 - Must assemble document corpus by spidering the web.
 - Can exploit the structural layout information in HTML (XML).
 - Documents change uncontrollably.
 - Can exploit the link structure of the web.

26

Web Search System



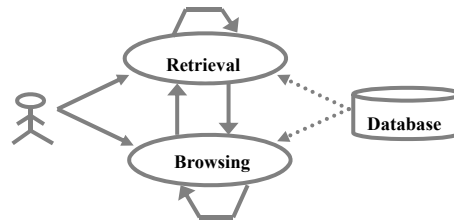
27

Basic Concepts

- The effective retrieval of relevant information is affected by:
 - The **task** the user of the retrieval system might be engaged in.
 - The **logical view of the documents** adopted by the retrieval system.
- User task:
 - Retrieval task: the user has to translate his information need into a query (the task is purposeful).
 - Information retrieval: a set of keywords is used to convey the semantics of the information need.
 - Data retrieval: a query expression (example: SQL query) is used to convey the constraints that must be satisfied by objects in the answer set.
 - Browsing task: a process whose main objectives are not clearly defined in the beginning and whose purpose might change during the interaction with the system.

28

Basic Concepts



- Notice:
 - Classic IR systems: allow information or data retrieval.
 - Hypertext systems: are tuned for providing quick browsing.
 - Modern digital library and Web interfaces: combine both tasks.
 - Example:
query a search engine, browse in the results, state another query, etc.

29

Basic Concepts

- Both retrieval and browsing are '**pulling**' actions:
The user requests (pulls) the information in an interactive manner.
- Alternative: automatic and permanent retrieval
 - Software agents **push** the information towards the user:
 - Dynamically compare newly received items against standing statements of interests of users (profiles) and deliver matching items to user mail files.
 - Asynchronous (background) process.
 - Profile defines all areas of interest (whereas an individual query focuses on specific question).
 - Each item compared against many profiles (whereas each query is compared against many items).
 - Example:
 - Information useful to a user could be extracted periodically from a news service.
 - The IR system *filters* relevant information for later inspection by the user.

30

Basic Concepts

- Logical view of the documents:

How does the retrieval system represent (view) the documents?

- Full text representation:

- Represent a document by its full set of words.
- Complete representation of the document (no information is lost).
- But: very large collections of documents imply higher computational costs

- Representation using a set of index terms (keywords):

- The keywords might be derived automatically from the text of the document or generated by a human specialist.
- Higher-level representation of the document (reduced complexity).
- But: perhaps retrieval of poor quality.

31

Basic Concepts

- Intermediate representations:

- Apply *text operations* to reduce the complexity of the document representation.
- Text operations:
 - Elimination of *stopwords*: articles, connectives, ... (a, and, or, the,).
 - *Stemming*: reduces distinct words to their common grammatical root (“compute”, “computer”, “computation”, “computerisation”, “computational”, ... → comput).
 - Identification of *noun groups*: eliminates adjectives, adverbs, and verbs.
 - Compression.

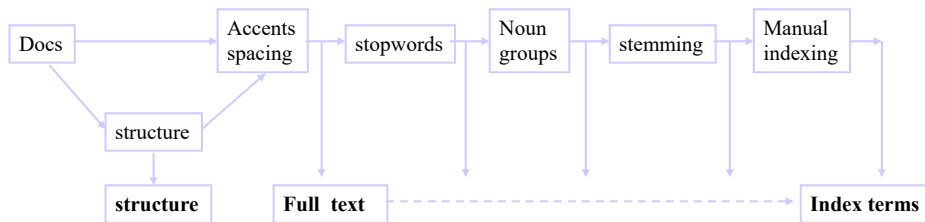
- In addition to adopting an appropriate representation, the retrieval system might also recognize the *internal structure* of the document:

- Example: chapters, sections, subsections, ...
- This information might be quite useful and is required by some systems.

32

Basic Concepts

- Logical view of the documents

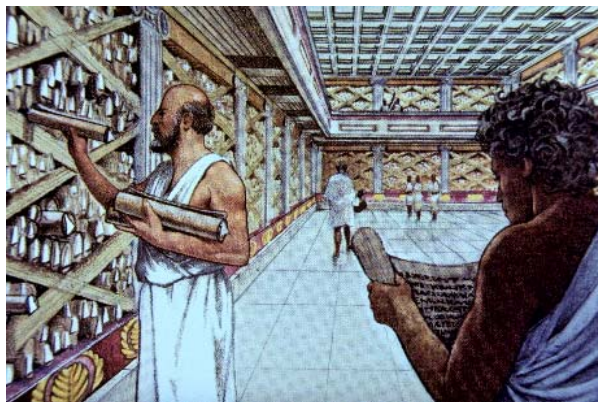


- Document representation viewed as a continuum from unprocessed text to a representation of documents' semantic content.

33

Brief History of IR

- IR grew out of library science and need to categorize, group, and access books and articles.



34

Brief History of IR

- For approximately 4000 years, man has organized information for later retrieval and usage:
 - Example: table of contents of a book.
 - As the amount of information grew beyond a few books, indexing was necessary. An index is:
 - an old and popular data structure for faster information retrieval.
 - a collection of selected words that are associated with pointers to the related information (or documents).
 - at the core of every modern information retrieval system.

35

Brief History of IR

- Libraries were among the first institutions to adopt IR systems for retrieving information:
 - First generation:
 - Automation of previous technologies (such as card catalogs).
 - Searches based on author name and title.
 - Second generation:
 - Increased search functionality was added: searching by subject headings, by keywords, and some more complex query facilities.
 - Third generation (now):
 - focus on improved graphical interfaces, electronic forms, hypertext features,

36

Brief History of IR

- IR as a Computer Science field (80s & early 90s):

Research focused on:

- Document classification and categorization
- Systems architecture
- User interfaces and visualization
- ...



Despite its maturity, the area was still seen as of narrow interest mainly to librarians and information experts!

37

Brief History of IR

- Advent of the Web changed this perception:

- Search engines on the Web use indexes which are very similar to those used by librarians a century ago.

– But now:

- Cheaper access to information.
- Greater access to information (networks allow distant and quick access).
- Publishing freedom (you can post whatever information you judge useful)



38

Brief History of IR

- Result: *the Web is a highly interactive medium*
people can exchange information in a convenient and low cost fashion at the time of their preference.
- Despite the convenience of the service, many question need to be addressed:
 - Which techniques will allow retrieval of higher quality information in the dynamic world of the Web?
 - Which techniques will yield faster indexes and smaller query response times?
 - How will a better understanding of the user behavior affect the design and deployment of new information retrieval strategies?
- IR is seen as key to finding the solutions!

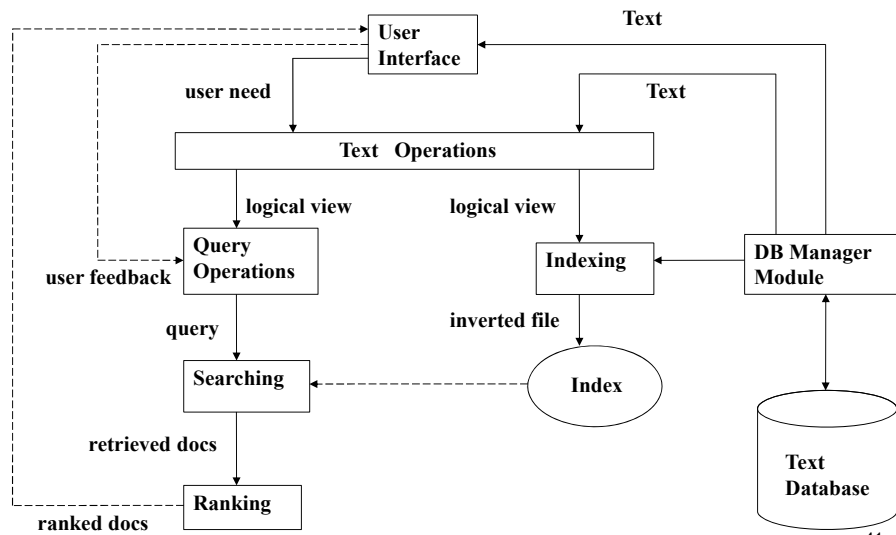
39

The IR Problem

- Two different views of the IR problem:
 - Computer-centered view:
 - Building efficient indexes.
 - Processing user queries with high performance.
 - Developing ranking algorithms which improve the 'quality' of the answer set.
 - Human-centered view:
 - Studying the behavior of the user.
 - Understanding his main needs.
 - Determining how such understanding affects the organization and operation of the retrieval system.
- In this course, we focus mainly on the computer-centered view.

40

IR System Architecture



41

IR System Components

- **Text Operations:** form index words (tokens).
 - Stopword removal
 - Stemming, ...
- **Indexing:**
 - Indexing results in records like
Doc12: napoleon, france, revolution, emperor
or (weighted terms)
Doc12: napoleon-8, france-6, revolution-4, emperor-7
 - To find all documents about Napoleon would involve looking at every document's index record (possibly 1,000s or millions).
 - (Assume that Doc12 references another file (documents file) which contains other details about the document.)

42

IR System Components

Or *inverted index*:

- Instead the information is *inverted* :
napoleon : doc12, doc56, doc87, doc99
or (weighted)
napoloen : doc12-8, doc56-3, doc87-5, doc99-2
- inverted file contains one record per index term
- inverted file is organized so that a given index term can be found quickly.
- **Searching:** retrieves documents that contain a given query token from the inverted index.
- **Ranking:** scores all retrieved documents according to a relevance metric.

43

IR System Components

- **User Interface:** manages interaction with the user:
 - Query input and document output.
 - Relevance feedback.
 - Visualization of results.
- **Query Operations:** transform the query to improve retrieval:
 - Query expansion using a thesaurus (book of words and their synonyms).
 - Query transformation using relevance feedback from the user.

44

The Retrieval Process

1. The database manager (using the DB Manager Module) defines the text database by specifying:
 - The documents to be used.
 - The operations to be performed on the text (to generate the logical view of the documents).
 - The text model (the text structure and what elements can be retrieved).
2. The database manager (using the DB Manager Module) builds an index of the text database:
 - An index allows fast searching.
 - Most popular index structure: *inverted file* (keywords, occurrences).

45

The Retrieval Process

3. The retrieval process can now be initiated:
 - The user specifies a *user need*.
 - The user need is parsed.
 - The user need is transformed (by the same text operations applied to the text) into a logical view.
 - **Query operations** are applied to generate the actual *query* (system representation for the user need).
 - The query is processed to obtain the *retrieved documents*.
 - The retrieved documents are ranked and sent to the user.
 - The user might initiate a *user feedback* (specify documents to be of interest).
 - The system uses the user feedback to change the query formulation.

46

The Retrieval Process

Notice:

- The time and storage spent in steps 1 and 2 are amortized by querying the retrieval system many times.
- Current IR systems like Web search engines and Web browsers are bad in allowing the user to declare his information need:
 - The user is required to provide a direct representation for the *query* that the system will execute (not user need!).
 - Most users have no knowledge of text and query operations
 - ➔ poorly formulated queries
 - ➔ poor retrieval.

47

Course Outline

- Chapter 2: Modeling
 - In the keyword-based approach to IR:
 - The user specifies his information need by providing sets of keywords.
 - The IR system retrieves the documents which best approximate the user query.
 - The IR system also attempts to rank the retrieved documents using some measure of relevance.
 - The *ranking task* is very important for satisfying the user information need and is the main goal of *modeling* in IR.
 - In this chapter:
 - We discuss important IR models.
 - We introduce many of the fundamental concepts in IR needed in the remaining part of the course.

48

Course Outline

- Chapter 3: Retrieval evaluation
 - We discuss how to evaluate the performance of retrieval algorithms in terms of “the relevance of the documents retrieved”.
- Chapter 4: Query languages
 - In traditional IR, queries are expressed as a set of keywords:
 - This approach is simple and easy to implement.
 - But: more elaborate queries cannot be formulated.
Example: queries that refer to both the structure and content of a document.
 - In this chapter: we discuss more sophisticated query languages.

49

Course Outline

- Chapter 5: Query operations
 - In retrieval based on keywords:
 - The user query might be composed of too few terms
→ query context is poorly characterized
→ retrieval of low quality.
 - This is frequently the case in the Web.
 - In this chapter: we discuss how to deal with this problem through transformations in the query such as query expansion and user relevance feedback.
- Chapter 6: Text languages
 - We discuss text languages used to describe a document content and its structure like HTML.

50

Course Outline

- Chapter 7: Text operations
 - In retrieval based on keywords:
 - The set of keywords generated for a given document might fail to summarize its semantic content properly
 - ➔ Retrieval of low quality.
 - In this chapter: we discuss how to deal with this problem through transformations in the text such as:
 - Identification of noun groups to be used as keywords.
 - Stemming.
 - Use of a thesaurus.
 - ...

51

Course Outline

- Chapter 8: Indexing and searching
 - We discuss indexing and searching techniques that can be used to speed up the task of matching documents to queries.
- Chapter 10: User interfaces and visualization
 - We discuss:
 - User interfaces for assisting the user to form his query.
 - Current approaches for visualization of large data sets of retrieved documents.

52

Web Programming and Development

Dr. Mohamed Salah Hamdi

PART2:

Modeling

1

Introduction

- IR systems usually adopt index terms to process queries.
- Index term:
 - a keyword or group of selected words
 - any word (more general)
- Retrieval based on index terms:
 - Idea: express semantics of the documents and the user information need through a set of keywords.

2

Introduction

- Common **Pre**processing Steps:
 - Strip unwanted characters/markup (e.g. HTML tags, punctuation, numbers, etc.).
 - Break into tokens (keywords) on whitespace.
 - Stem tokens to “root” words
 - connecting, connection, connections → connect
 - Remove common stopwords (e.g. a, the, it, etc.).
 - Detect common phrases (possibly using a domain specific dictionary).
 - Build index.

3

Introduction

- Indexing results in records like
Doc12: napoleon, france, revolution, emperor
or (weighted terms)
Doc12: napoleon-8, france-6, revolution-4, emperor-7
- To find all documents about Napoleon would involve looking at every document’s index record (possibly 1,000s or millions).
- (Assume that Doc12 references another file which contains other details about the document. This file is called: *documents file*)

4

Introduction

- Instead the information is ***inverted*** (*inverted index*):
napoleon : doc12, doc56, doc87, doc99
or (weighted)
napoleon : doc12-8, doc56-3, doc87-5, doc99-2
- inverted file contains one record per index term
- inverted file is organised so that a given index term can be found quickly.

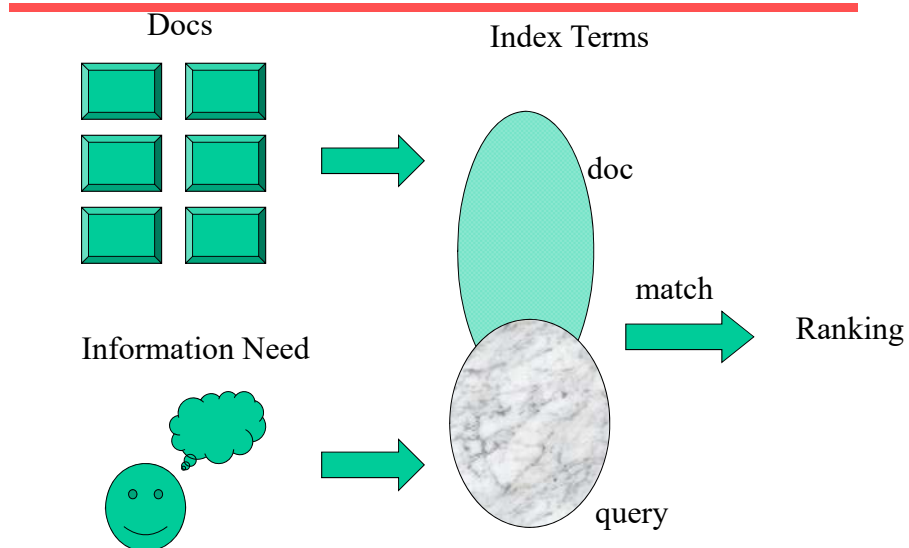
5

Introduction

- To find documents satisfying the query
napoleon and revolution
 - use the inverted file to find the set of documents indexed by *napoleon* → doc12, doc56, doc87, doc99
 - similarly find the set of documents indexed by *revolution* → doc12, doc20, doc45, doc99
 - **intersect** these sets → doc12, doc99
 - go to the documents file to retrieve title, authors, location for these documents.
- For **or**, union the sets
 - e.g. Napoleon **or** Waterloo
- For **not**, use set complement
 - e.g. **not** Waterloo

6

Introduction



Introduction

- Retrieval based on keywords is simple, but a lot of semantics in a *document* or *user information need* is lost:
 - ➔ Matching of user request with documents at index term level is quite imprecise
- Since most users have no training in query formation, problem is even worse
 - ➔ Poor retrieval results: many documents are irrelevant!

Introduction

- One central problem in IR is therefore deciding which documents are *relevant* and which are not:
 - A *ranking algorithm* is needed to order the retrieved documents according to their *relevance*.
 - The ranking algorithm is the core of the IR system.
 - The ranking algorithm depends on the *notion of document relevance*.
 - The notion of document relevance depends on the *IR model* adopted by the IR system.

9

Introduction

- What is a retrieval model?
 - Model is an idealization or abstraction of an actual process (here, retrieval).
 - Mathematical models are used to study the properties of the process, draw conclusions, make predictions.
 - Conclusions derived from a model depend on whether the model is a good approximation of the actual situation.

10

IR Models

- A retrieval model specifies the details of:
 - Document representation
 - Query representation
 - Retrieval function
- Determines a notion of relevance.
- Notion of relevance can be **binary** or **continuous** (i.e. *ranked retrieval*).

11

IR Models

- Formal Characterization of IR Models:
An IR model is a quadruple
 - $[\mathbf{D}, \mathbf{Q}, F, R(q_i, d_j)]$
 - Where
 - \mathbf{D} is a set of logical views of documents
 - \mathbf{Q} is a set of logical views of queries
 - F is a framework for modelling documents, queries and their relationships
 - $R(q_i, d_j)$ is a ranking function which rates document d_j according to query q_i

12

IR Models

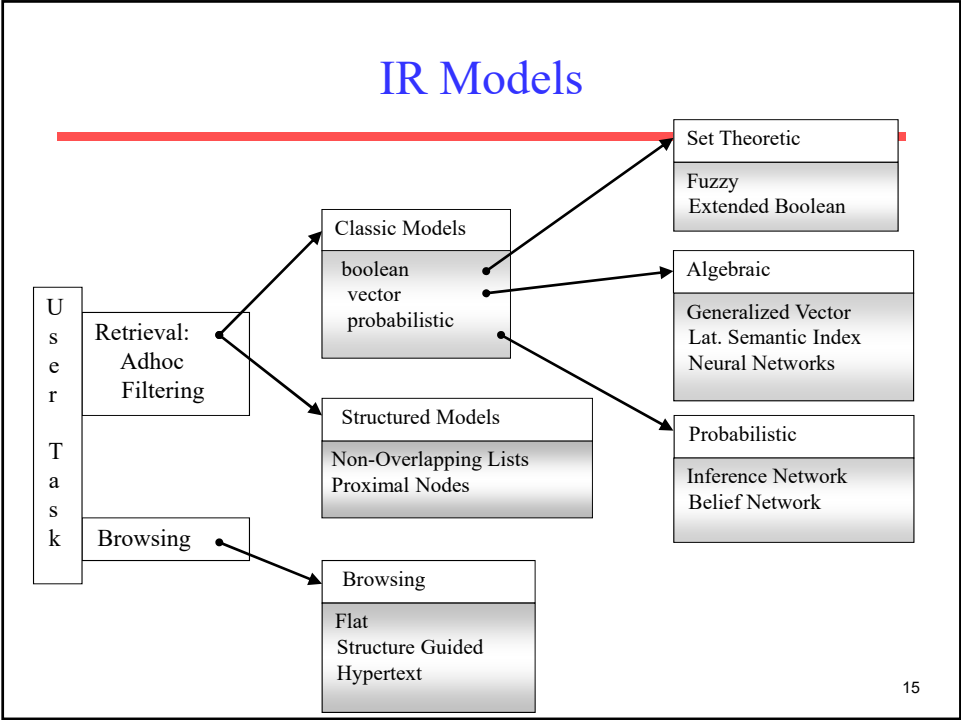
- There are 3 classic IR models:
 - Boolean model:
 - Documents and queries are represented as **sets** of index terms.
 - The model is set theoretic: standard operations on sets are applied to determine the ranking.
 - Vector model:
 - Documents and queries are represented as **vectors** in a t-dimensional space.
 - The model is algebraic: standard linear algebra operations on vectors are applied to determine the ranking.
 - Probabilistic model:
 - Documents and queries are represented based on **probability** theory.
 - The model is probabilistic: standard probability operations are applied to determine the ranking.

13

IR Models

- There are also many extensions of these classic models.
- Other model dimensions:
 - Logical View of Documents
 - Index terms
 - Full text
 - Full text + Structure (e.g. hypertext)
 - User Task
 - Retrieval
 - Browsing

14



IR Models

- The IR model, the logical view of the docs, and the retrieval task are distinct aspects of the system.

LOGICAL VIEW OF DOCUMENTS

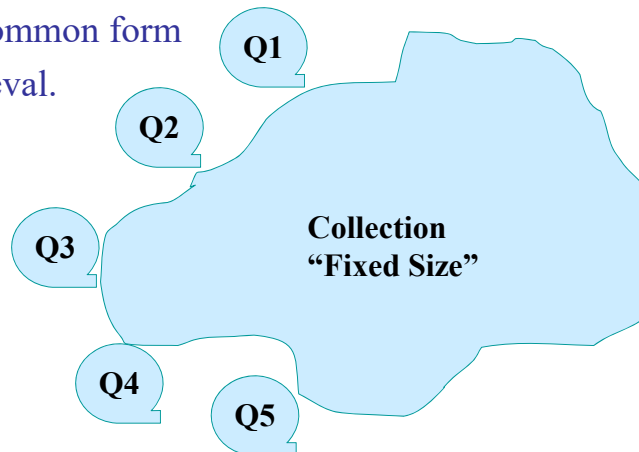
		Index Terms	Full Text	Full Text + Structure
USER TASK	Retrieval	Classic Set Theoretic Algebraic Probabilistic	Classic Set Theoretic Algebraic Probabilistic	Structured
	Browsing	Flat	Flat Hypertext	Structure Guided Hypertext

16

Retrieval Tasks

- **Ad hoc retrieval:**

- Fixed document corpus, varied queries.
- Most common form of retrieval.



17

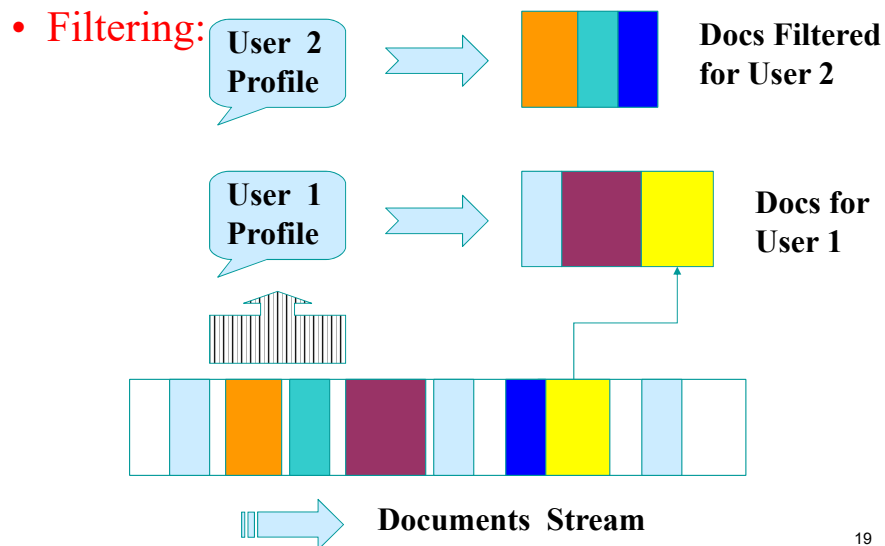
Retrieval Tasks

- **Filtering:**

- Fixed query, continuous document stream.
- Applications: stock market, news articles, ...
- A ***user profile*** that describes the user preferences is needed.
- The user profile is compared to incoming docs in order to determine their suitability for the user.
- No ranking of the filtered documents is provided.

18

Retrieval Tasks



Retrieval Tasks

- Constructing a user profile that truly reflects the user's preferences is crucial.
- Approaches for constructing user profiles:
 - Require the user to provide a set of keywords that describe his profile:
 - If the user is not familiar with the service that generates the upcoming documents, it will be difficult for him to provide appropriate keywords.
 - Letting the user familiarize himself with the vocabulary of the upcoming documents may be tedious and time consuming.

20

Retrieval Tasks

- Build the user profile dynamically based on relevance feedback:
 - At the beginning, the user provides a set of keywords that describe an *initial profile*.
 - The system uses this profile to select documents and shows them to the user.
 - The user provides a feedback (which documents are relevant and which are not).
 - The system uses this feedback to adjust the user profile.
 - The user profile changes continually, but stabilizes hopefully after a while.
- **Routing**: Same as filtering but continuously supply ranked lists rather than binary filtering.

21

Classic IR Models - Basic Concepts

- Each document is represented by a set of representative *keywords* or *index terms*.
- An *index term* is a document word useful for remembering the document's main themes.
- Index terms summarize the document contents.
- Usually, index terms are nouns because nouns have meaning by themselves.
- However, search engines assume that all words are index terms (full text representation).
- We discuss the problem of generating index terms later.

22

Classic IR Models - Basic Concepts

- Not all terms are equally useful for representing the document contents:
consider a collection of 100000 documents
 - A word which appears in each of the 100000 documents is completely useless as an index term because it does not tell us anything about which documents the user may be interested in.
 - A word which appears in just 5 documents is quite useful because it narrows down considerably the space of the documents which may be of interest to the user.
- Each index term is assigned a numerical *weight* that quantifies the *importance* of the index term for describing the document semantic contents.

23

Classic IR Models - Basic Concepts

- Notation:
 - k_i : an index term
 - d_j : a document
 - $w_{ij} \geq 0$: a weight associated with the pair (k_i, d_j)
 $w_{ij} = 0$ means index term does not appear in the text of document d_j .

24

Classic IR Models - Basic Concepts

- t : is the total number of index terms in the system.
- $K = \{k_1, k_2, \dots, k_t\}$: is the set of all index terms in the system.
- $\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj})$: is a weighted vector associated with the document d_j (document vector).
- $g_i(\vec{d}_j) = w_{ij}$: is a function which returns the weight associated with index term k_i in vector \vec{d}_j .

25

Classic IR Models - Basic Concepts

- Note: the index term weights might be:
 - Mutually independent:
 - Knowing the weight w_{ij} associated with the pair (k_i, d_j) tells us nothing about the weight w_{lj} associated with the pair (k_l, d_j) .
 - This simplifies the task of computing index term weights and allows fast ranking computation.
 - Correlated:
 - Suppose that the terms “*computer*” and “*network*” are used to index a given document that covers the area of computer networks:
 - The appearance of one of these two words attracts the appearance of the other → the two words are correlated → their weights are correlated.
 - There is no evidence that index term correlations are advantageous for ranking purposes.
- In the following, we assume mutual independence.

26

The Boolean Model

- This model considers index terms as either present or absent in a document. Thus:
 - $w_{ij} \in \{0,1\}$
 - Each document is represented with a binary weighted vector \vec{d}_j
- A query q is composed of index terms linked by: **not**, **and**, **or**. Thus:
 - q is a conventional Boolean expression.
 - q can be represented in disjunctive normal form (DNF):
a disjunction of conjunctive vectors
 - Example: $q = k_a \wedge (k_b \vee \neg k_c)$

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

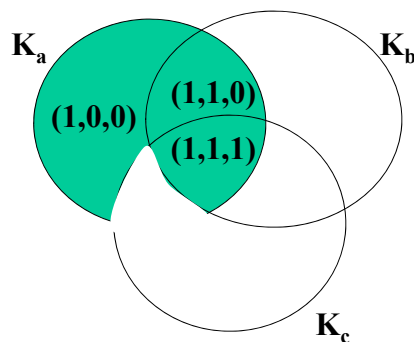
$$= (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c) \vee$$

$$(k_a \wedge \neg k_b \wedge \neg k_c) \vee (k_a \wedge \neg k_b \wedge k_c)$$
 - $\vec{q}_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$

27

The Boolean Model

- \vec{q}_{dnf} is called **query vector**.
- Each of its components \vec{q}_{cc} (example: $\vec{q}_{cc} = (1,1,0)$):
 - Is a binary weighted vector associated with the tuple (k_a, k_b, k_c) .
 - Is called a **conjunctive component** of \vec{q}_{dnf} .
- The figure illustrates the three conjunctive components of the query:



28

The Boolean Model

- Matching queries and documents:

- Let d_j be a document with document vector \vec{d}_j .
- Let q be a query with query vector \vec{q}_{dnf} .
- Let \vec{q}_{cc} be any of the conjunctive components of \vec{q}_{dnf} .
- The **similarity** of document d_j to query q is defined as:

$$\text{sim}(q, d_j) = \begin{cases} 1 & \text{if } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 & \text{otherwise} \end{cases}$$

- If $\text{sim}(q, d_j) = 1$, the Boolean model **predicts** that the document is relevant to the query (there is a query component that matches exactly the document vector).
- If $\text{sim}(q, d_j) = 0$, the Boolean model **predicts** that the document is not relevant to the query.

29

The Boolean Model

- Example:

- Terms: k_1, k_2, k_3 .

- Documents:

- $d_1 = \{k_2\}$
- $d_2 = \{k_1, k_2, k_3\}$
- $d_3 = \{\}$
- $d_4 = \{k_2, k_3\}$
- $d_5 = \{k_1\}$

- Query: $q = k_1 \wedge (k_2 \vee \neg k_3)$

$$\vec{q}_{dnf} = (1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0)$$

- Similarities:

- $\text{sim}(q, d_1) = 0$: d_1 includes the index term k_2 , but is considered non-relevant to q (**no** notion of **partial match** in Boolean model!).
- $\text{sim}(q, d_2) = 1, \text{sim}(q, d_3) = 0, \text{sim}(q, d_4) = 0, \text{sim}(q, d_5) = 1$
- Answer: $\{d_2, d_5\}$

	\vec{d}_1	\vec{d}_2	\vec{d}_3	\vec{d}_4	\vec{d}_5
k_1	0	1	0	0	1
K_2	1	1	0	1	0
K_3	0	1	0	1	0

30

The Boolean Model

- Advantages:
 - Clean formalism behind the model:
 - Simple retrieval model based on set theory and Boolean algebra.
 - Queries are specified as Boolean expressions → precise semantics.
 - Easy to understand by a common user of an IR system for simple queries:
 - Was adopted by many of the early commercial bibliographic systems.
 - Is still the dominant model with commercial document databases.
 - Reasonably efficient implementations possible for normal queries:
 - Implementation of the Boolean model using *inverted index*: see slide 6.
- and* = intersection of document sets
or = union of document sets
not = complement of document sets

31

The Boolean Model

- Drawbacks:
 - No notion of partial match (no grading scale):
 - A document is either relevant or non-relevant.
 - This model is much more a data retrieval model than an IR model.
 - Exact match → bad retrieval performance (either too few or too many documents).
 - Information need has to be translated into a Boolean expression:
 - Difficult to express complex user requests.
 - The Boolean queries formulated by the users are most often too simplistic.

32

The Vector Model

- Use of binary weights is too limiting: no partial matching between query and document.
- The vector model proposes the following solution:
 - Assign *non-binary weights* to index terms in queries and documents.
 - Use the weights to compute the *degree of similarity* between each document and the user query.
 - Retrieve only documents that have a degree of similarity greater than a certain threshold.
 - In this way, it is possible to take into account documents that match the query terms only partially.

33

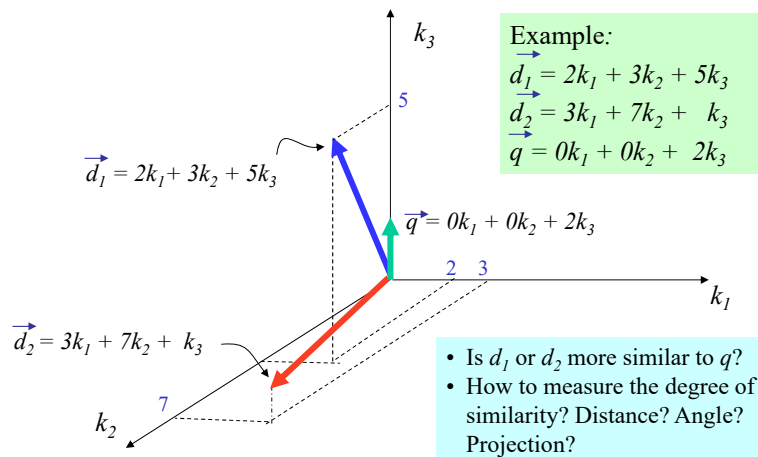
The Vector Model

- Notation:
 - k_i : index term
 - q : query
 - d_j : document
 - t : total number of index terms in the system
 - $w_{ij} \geq 0$: weight associated with the pair (k_i, d_j)
 - $\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj})$: document vector for document d_j
 - $w_{iq} \geq 0$: weight associated with the pair (k_i, q)
 - $\vec{q} = (w_{1q}, w_{2q}, \dots, w_{tq})$: query vector for query q

34

The Vector Model

- Queries and documents are represented as t -dimensional vectors:



35

The Vector Model

- A collection of N documents can be represented in the vector space model 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.

$$\begin{pmatrix} & k_1 & k_2 & \dots & k_t \\ d_1 & w_{11} & w_{21} & \dots & w_{t1} \\ d_2 & w_{12} & w_{22} & \dots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ d_N & w_{1N} & w_{2N} & \dots & w_{tN} \end{pmatrix}$$

36

The Vector Model

- Similarity measure:
 - A **similarity measure** is a function that computes the *degree of similarity* between two vectors.
 - Using a similarity measure between the query and each document:
 - It is possible to rank the retrieved documents in the order of presumed relevance.
 - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.
 - A document may be retrieved even it matches the query only partially.

37

The Vector Model

Similarity Measure - Inner Product

- Similarity between vectors for the document d_j and query q can be computed as the vector inner product:

$$\text{sim}(d_j, q) = \vec{d}_j \cdot \vec{q} = \sum_{i=1}^t w_{ij} \cdot w_{iq}$$

where w_{ij} is the weight of term k_i in document d_j and w_{iq} is the weight of term k_i in the query

- For binary vectors, the inner product is the number of matched query terms in the document (size of intersection).
- For weighted term vectors, it is the sum of the products of the weights of the matched terms.

38

The Vector Model

Properties of Inner Product

- The inner product is unbounded.
- Favors long documents with a large number of unique terms.
- Measures how many terms matched but not how many terms are *not* matched.

39

The Vector Model

Inner Product -- Examples

Binary: 

$$d = 1, 1, 1, 0, 1, 1, 0$$

$$q = 1, 0, 1, 0, 0, 1, 1$$

Size of vector = size of vocabulary = 7

0 means corresponding term not found in document or query

$$\text{sim}(d, q) = 3$$

Weighted:

$$d_1 = 2k_1 + 3k_2 + 5k_3 \quad d_2 = 3k_1 + 7k_2 + 1k_3$$

$$q = 0k_1 + 0k_2 + 2k_3$$

$$\text{sim}(d_1, q) = 2*0 + 3*0 + 5*2 = 10$$

$$\text{sim}(d_2, q) = 3*0 + 7*0 + 1*2 = 2$$

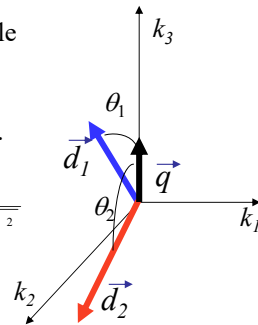
40

The Vector Model

Similarity Measure - Cosine

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.
- $w_{ij} \geq 0$ and $w_{iq} \geq 0$: similarity is between 0 and 1.

$$\text{CosSim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^I (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^I w_{ij}^2} \cdot \sqrt{\sum_{i=1}^I w_{iq}^2}}$$



$$\begin{aligned} d_1 &= 2k_1 + 3k_2 + 5k_3 & \text{CosSim}(d_1, q) &= 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81 \\ d_2 &= 3k_1 + 7k_2 + 1k_3 & \text{CosSim}(d_2, q) &= 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13 \\ q &= 0k_1 + 0k_2 + 2k_3 \end{aligned}$$

d_1 is 6 times better than d_2 using cosine similarity but only 5 times better using inner product.

41

The Vector Model

How index term weights are obtained?

- There are many term-weighting techniques.
- Here we discuss only one based on **clustering**.
- Clustering problem:
 - Given a collection C of objects and a **vague** description of a set A.
 - Separate the objects in C into two sets:
 - One containing objects related to the set A.
 - One containing objects **not** related to the set A.
 - Vague description means:
 - We do not have complete information for deciding which objects are in A and which are not.
 - Example: Set A of cars having a price **comparable** to that of a Lexus 400.
 - The exact meaning of comparable is not clear
 - ➔ no precise description of A.

42

The Vector Model

- Other clustering algorithms may produce more than 2 classes:
 - Example: five classes for cancer patient
terminal, advanced, metastasis, diagnosed, healthy
 - Again, the class description are imprecise.
 - Clustering problem: decide to which of these classes a new patient should be assigned.
- Here we only consider the clustering problem with two classes:
 - Documents predicted to be **relevant** to the user query.
 - Documents predicted to be **not relevant** to the user query.

43

The Vector Model

- The IR problem viewed as a clustering problem:
 - We think of the documents as a collection C of objects.
 - We think of the query as a vague specification of a set A of Objects.
- Two main issues have to be solved:
 - Determine the features that better describe the objects in the set A
→ quantify the *intra-cluster* similarity.
 - Determine the features that better distinguish the objects in the set A from the remaining objects in the collection C .
→ quantify the *inter-cluster* dissimilarity.
- Successful clustering algorithms try to balance these two effects.

44

The Vector Model

- In the vector model:
 - Intra-cluster similarity is quantified by measuring the raw frequency of a term k_i inside a document d_j :
 - This term frequency is called *tf factor*.
 - It provides a measure of how well the term describes the document contents.
 - Inter-cluster dissimilarity is quantified by measuring the inverse of the frequency of a term k_i among the documents in the collection:
 - This factor is called *inverse document frequency* or *idf factor*.
 - Motivation for using this factor:
terms that appear in many documents are not very helpful for distinguishing a relevant document from a non-relevant one.

45

The Vector Model

- Term-weighting schemes for IR try to balance these two effects.
- Notation:
 - N : total number of documents in the system.
 - n_i : the number of documents in which the index term k_i appears (document frequency of k_i).
 - $freq_{ij}$: the number of times the term k_i appears in document d_j (raw frequency of term k_i in document d_j).

46

The Vector Model

- tf factor:

$$tf_{ij} = (freq_{ij}) / (\max_l freq_{lj})$$

- This is the normalized frequency of term k_i in document d_j .
- The max is computed on all terms which are mentioned in the text of document d_j .
- If term k_i does not appear in document d_j then $tf_{ij} = 0$.

- idf factor:

$$idf_i = \log(N/n_i)$$

- This is the inverse document frequency for term k_i .
- the \log is used to make the values of tf and idf comparable.

47

The Vector Model

- tf-idf weighting:

$$w_{ij} = tf_{ij} * idf_i$$

- This is a typical combined term importance indicator.
- 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.

48

The Vector Model

- 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: $tf = 3/3$; $idf = \log(10000/50) = 5.3$; $tf-idf = 5.3$

B: $tf = 2/3$; $idf = \log(10000/1300) = 2.0$; $tf-idf = 1.3$

C: $tf = 1/3$; $idf = \log(10000/250) = 3.7$; $tf-idf = 1.2$

49

The Vector Model

Query vector:

- For the query term weights, a suggestion is

$$w_{iq} = (0.5 + [0.5 * freq_{i,q} / \max_l(freq_{l,q})]) * \log(N/n_i)$$

- $freq_{i,q}$ is the raw frequency of the term k_i in the text of the query q .
- The max is computed on all terms which are mentioned in the text of the query q .

- An alternative is for the user to supply weights for the given query terms.

50

The Vector Model

Naïve Implementation:

Convert all documents in collection D to tf-idf weighted vectors, \vec{d}_j , for keyword vocabulary $V = K$ (set of all keywords).

Convert query to a tf-idf-weighted vector \vec{q} .

For each \vec{d}_j in D do

 Compute score $s_j = \text{CosSim}(\vec{d}_j, \vec{q})$

Sort documents by decreasing score.

Present top ranked documents to the user.

Time complexity: $O(|V| \cdot |D|)$ Bad for large V & D !

$|V| = 10,000$; $|D| = 100,000$; $|V| \cdot |D| = 1,000,000,000$

51

The Vector Model

- Comments on vector model:

- Simple, mathematically based approach.
- Considers both local (*tf*) and global (*idf*) word occurrence frequencies.
- Provides partial matching and ranked results.
- Allows efficient implementation for large document collections.
- Is a good ranking strategy with general collections
- Is usually as good as any known ranking alternatives.

52

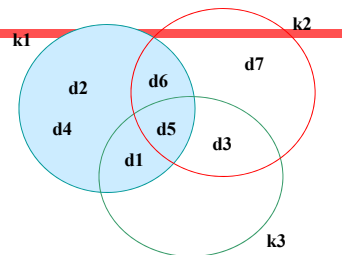
The Vector Model

- Problems with vector model:
 - Missing semantic information (e.g. word sense).
 - Missing syntactic information (e.g. phrase structure, word order, proximity information).
 - Assumes independence of index terms (??); not clear that this is bad though
 - 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.

53

The Vector Model

- Example I:

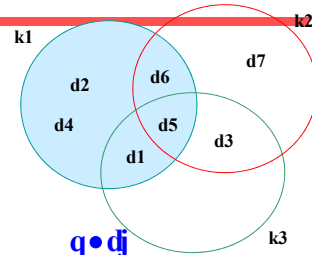


	k1	k2	k3	$q \bullet d_j$	$ d_j $	$\text{Sim}(d_j, q)$
d1	1	0	1	2	1.41	0.82
d2	1	0	0	1	1	0.58
d3	0	1	1	2	1.41	0.82
d4	1	0	0	1	1	0.58
d5	1	1	1	3	1.73	1
d6	1	1	0	2	1.41	0.82
d7	0	1	0	1	1	0.58
q	1	1	1	$ q $	1.73	

54

The Vector Model

- Example II:

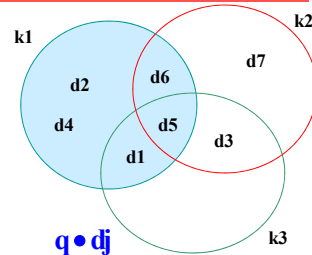


	k1	k2	k3	$q \bullet d_j$
d1	1	0	1	4
d2	1	0	0	1
d3	0	1	1	5
d4	1	0	0	1
d5	1	1	1	6
d6	1	1	0	3
d7	0	1	0	2
q	1	2	3	

55

The Vector Model

- Example III:



	k1	k2	k3	$q \bullet d_j$
d1	2	0	1	5
d2	1	0	0	1
d3	0	1	3	11
d4	2	0	0	2
d5	1	2	4	17
d6	1	2	0	5
d7	0	5	0	10
q	1	2	3	

56

The Probabilistic Model

Brief review of probabilities:

- Coin toss
 - Probability of heads is $\frac{1}{2}$, probability of tails is $\frac{1}{2}$
 - $P(\text{heads}) + P(\text{tails}) = 1$
- Fair die
 - Probability of throwing any number is $\frac{1}{6}$
- Bag of marbles (2 red, 3 green, 4 blue)
 - Probability of drawing a red marble is $\frac{2}{9}$
 - Probability of not drawing blue is: $\frac{2}{9} + \frac{3}{9} = \frac{5}{9} = 1 - \frac{4}{9}$

57

The Probabilistic Model

- Addition and multiplication rules
 - The probability of the union of two independent events is the sum of their individual probabilities:
 - $P(A \text{ or } B) = P(A) + P(B)$
 - The probability of the intersection of two independent events is the product of their probabilities:
 - $P(A \text{ and } B) = P(A) * P(B)$
 - Example: Ali and Omar play tennis each week
 - So far this year Ali has won 12 times and Omar 18
 - What is the chance of Ali winning tomorrow? $\frac{12}{30}$
 - Ali also plays with Mohamed and wins 1 of 4 games.
 - What is the chance of Ali winning both games this week? $\frac{12}{30} * \frac{1}{4}$
 - What is the chance of Ali winning one of the games this week?
 $\frac{12}{30} + \frac{1}{4}$

58

The Probabilistic Model

- Conditional Probability
 - Draw 2 cards from a deck of 52 cards
 - What is the probability that the second is the 7♠ ?
 - 1/52 – in the absence of other info
 - If the first card is not the 7♠ then the probability is 1/51, otherwise it is 0.

We call the probability of event J given the occurrence of event H the conditional probability of event J given the occurrence of H.

This conditional probability is represented by the ratio between the probability of the two events intersecting (i.e., their occurrence together) to the probability of event H occurring, i.e.

$$P[J | H] = \frac{P[J \cap H]}{P[H]}$$

If the value of the conditional probability of event J given the occurrence of H does not change from the original unconditional value of the event (that is, the probability is always the same whether H occurs or does not occur), then we say that these two events are independent.

59

The Probabilistic Model

- Conditional Probability
 - Say I want to go jogging and I want to ring the friends most likely to come along:
 - Data from the last 20 weekends:
 - Ali 9
 - Omar 10
 - Salah 12
 - Salah is most likely to come out: $12/20 = 0.6$ probability

60

The Probabilistic Model

- However, if we have data on temperature levels and high temperature is expected...

	Low	Med	High	Tot
Ali	2	4	3	9
Omar	2	4	4	10
Salah	6	5	1	12
Tot	7	8	5	20

$P[A] = 9/20$
 $P[A | H] = 3/5$
 $P[O | H] = 4/5$
 $P[S | H] = 1/5$

Number of weekends in which temperature was high (out of 20)

$P[A|H]$ reads as "probability of A given H"

61

The Probabilistic Model

- Given two events J and H: $P[J | H] = \frac{P[J \cap H]}{P[H]}$
- Probability of J given H is the same as probability of J *iff*

J and H independent:

$$P[J | H] = P[J] \iff P[J] = \frac{P[J \cap H]}{P[H]} \iff P[J \cap H] = P[J] * P[H]$$

	Low	Med	High	Tot
Ali	2	4	3	9
Omar	2	4	4	10
Salah	6	5	1	12
Tot	7	8	5	20

$P[A] = 9/20$
 $P[A | H] = 3/5$
 $P[H] = 5/20$
 $P[A \cap H] = P[A | H] * P[H]$
 $= 3/5 * 5/20 = 3/20$
 $P[A] * P[H] = 9/20 * 5/20$
 $= 45/400 \neq P[A \cap H]$
→ A and H are correlated

The Probabilistic Model

– Given two events J and H: $P[J | H] = \frac{P[J \cap H]}{P[H]}$

– Probability of J given H is the same as probability of J *iff*

J and H independent: $P[J | H] = P[J] \iff P[J] = \frac{P[J \cap H]}{P[H]} \iff P[J \cap H] = P[J] * P[H]$

	Low	Med	High	Tot
Ali	2	4	3	9
Omar	2	4	4	10
Salah	6	5	1	12
Tot	7	8	5	20

$P[A] = 9/30$
 $P[A | H] = 3/10$
 $P[H] = 10/30$
 $P[A \cap H] = P[A | H] * P[H]$
 $= 3/10 * 10/30 = 3/30 = 1/10$
 $P[A] * P[H] = 9/30 * 10/30$
 $= 9/90 = 1/10 = P[A \cap H]$
→ A and H are independent

The Probabilistic Model

• Bayes' rule: $P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$

	Low	Med	High	Tot
Ali	2	4	3	9
Omar	2	4	4	10
Salah	6	5	1	12
Tot	7	8	5	20

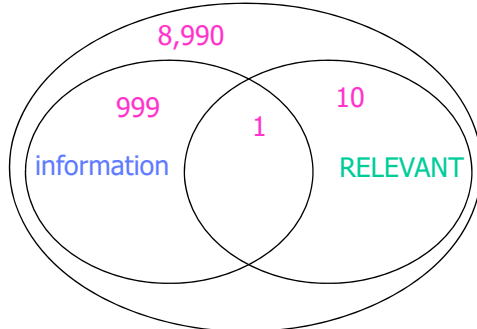
$P[H] = 5/20$
 $P[A] = 9/20$
 $P[O] = 10/20$
 $P[S] = 12/20$
 $P[A | H] = 3/5$
 $P[O | H] = 4/5$
 $P[S | H] = 1/5$

$P[H | A] = 3/9$ $(P[A | H] * P[H]) / P[A] = (3/5 * 5/20) / (9/20) = 3/9$
 $P[H | O] = 4/10$ $(P[O | H] * P[H]) / P[O] = (4/5 * 5/20) / (10/20) = 4/10$
 $P[H | S] = 1/12$ $(P[S | H] * P[H]) / P[S] = (1/5 * 5/20) / (12/20) = 1/12$

64

The Probabilistic Model

- Ex. of Probabilities in IR:



total #docs = 10,000
 #relevant docs = 11
 #docs containing "information" = 1000
 #relevant docs containing "information" = 1
 #docs containing "computer" = 2000
 #relevant docs containing "computer" = 2

- Probability that index term "information" is present in a doc randomly selected from the entire collection:
 $P(\text{"information"}) = 1000/10000 = 1/10$

- Probability that index terms "information" and "computer" are present in a doc randomly selected from the entire collection (no inf available about set of docs containing "computer" → assume independence):
 $P(\text{"information" and "computer"}) = 1000/10000 * 2000/10000 = 2/100 = 1/50$

- Probability that "information" is present in a doc given it is relevant: $P(\text{"information"} | R) = 1/11$

- Probability that "information" and "computer" are present in a doc given it is relevant (assume independence):
 $P(\text{"information", "computer"} | R) = P(\text{"information"} | R) * P(\text{"computer"} | R) = 1/11 * 2/11 = 2/121$

- Probability of relevance given a doc containing or not containing "information":
 $P(R | \text{"information"}) = 1/1000$
 $P(R | \text{not "information"}) = 10/9000$

- Probability of relevance: $P(R) = 11/10000$

- Probability of non-relevance: $P(\text{not } R) = 9989/10000$

65

The Probabilistic Model

- Idea behind probabilistic model:

- Given a user query, there is a set of documents which contains exactly the relevant documents (*ideal answer set*).
- Given the description of this ideal answer set, it would be easy to retrieve its documents.
 → IR problem = a process of specifying the properties of an ideal answer set (clustering).

- Problem:

- we do not know exactly the properties of the ideal answer set.
- All we know is a set of index terms (query).

66

The Probabilistic Model

- Solution:
 - Provide an *initial guess* of the properties of the ideal answer set.
 - Use these properties to retrieve a first set of documents.
 - The user takes a look at the retrieved documents and decides which ones are relevant and which ones are not (only the first few documents need to be examined).
 - Use this information to refine the properties of the ideal answer set.
 - By repeating the last 3 steps many times, it is expected to obtain properties that are closer to the real ones.

67

The Probabilistic Model

- The probabilistic model attempts to realize this solution in probabilistic terms:
 - It tries to estimate the probability that the user will find the document d_j relevant for query q .
 - It assumes that this probability depends only on the document and query representation.
 - It assumes that there is an *ideal answer set* R that maximizes the overall probability of relevance to the user.
 - Measure of similarity between document d_j and query q :

$$Sim(q, d_j) = \frac{P(d_j \text{ relevant to } q)}{P(d_j \text{ not relevant to } q)}$$

- This quotient is called the odds of the document d_j being relevant.
- Many studies showed that taking the odds of relevance as the rank minimizes the probability of an erroneous judgment (i.e., deciding that d_j is relevant when it is not and vice versa).

68

The Probabilistic Model

- Notation:

- $w_{ij} \in \{0, 1\}$: Index terms weights are binary.
- $w_{iq} \in \{0, 1\}$: Query q is a subset of the index terms.
- R : the set of documents known (or initially guessed) to be relevant to q .
- \bar{R} : complement of R (the set of non-relevant documents to q).
- $P(R|\vec{d}_j)$ = probability that document d_j is relevant to q .
= probability of relevance given a particular document d_j .
= the probability that a document is relevant given its content.
- $P(\bar{R}|\vec{d}_j)$ = probability that document d_j is non-relevant to q .

69

The Probabilistic Model

- Similarity:

$$Sim(q, d_j) = \frac{P(R|\vec{d}_j)}{P(\bar{R}|\vec{d}_j)}$$

\swarrow prob that d_j is relevant to query q \swarrow prob that d_j is not relevant to query q

- Bayes' Rule:

$$P(R|\vec{d}_j) = \frac{P(\vec{d}_j | R) \times P(R)}{P(\vec{d}_j)} \quad P(\bar{R}|\vec{d}_j) = \frac{P(\vec{d}_j | \bar{R}) \times P(\bar{R})}{P(\vec{d}_j)}$$

70

The Probabilistic Model

- Applying Bayes' Rule: $\frac{P(\vec{d}_j | R) \times P(R)}{P(\vec{d}_j | \bar{R}) \times P(\bar{R})}$

$$Sim(d_j, q) = \frac{P(\vec{d}_j | R) \times P(R)}{P(\vec{d}_j | \bar{R}) \times P(\bar{R})}$$

prob of randomly selecting doc d_j from the set R of relevant docs
 prob that a doc randomly selected from entire collection is relevant
 prob of randomly selecting doc d_j from the set \bar{R}
 prob that a doc randomly selected from entire collection is non-relevant

71

The Probabilistic Model

- Since $P(R)$ and $P(\bar{R})$ are the same for all the doc in the collection, $\frac{P(R)}{P(\bar{R})}$ is constant:

$$Sim(d_j, q) \approx \frac{P(\vec{d}_j | R)}{P(\vec{d}_j | \bar{R})}$$

prob that index term k_i is present in a doc randomly selected from R
 prob that index term k_i is not present in a doc randomly selected from R

Assuming independence of index terms

$$Sim(d_j, q) \approx \frac{\left[\prod_{g_i(\vec{d}_j)=1} P(k_i | R) \right] \times \left[\prod_{g_i(\vec{d}_j)=0} P(\bar{k}_i | R) \right]}{\left[\prod_{g_i(\vec{d}_j)=1} P(k_i | \bar{R}) \right] \times \left[\prod_{g_i(\vec{d}_j)=0} P(\bar{k}_i | \bar{R}) \right]}$$

72

The Probabilistic Model

$$Sim(d_j, q) \approx \left[\prod_{g_i(\vec{d}_j)=1} \frac{P(k_i | R)}{P(k_i | \bar{R})} \right] \times \left[\prod_{g_i(\vec{d}_j)=0} \frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right]$$

- For $g_i(\vec{q}) = 0$ ($k_i \notin q$), we assume that $P(k_i | R) = P(k_i | \bar{R})$
 $\rightarrow P(\bar{k}_i | R) = P(\bar{k}_i | \bar{R}) \rightarrow$ the corresponding ratios will be equal to 1:

$$Sim(d_j, q) \approx \left[\prod_{g_i(\vec{d}_j)=1, g_i(\vec{q})=1} \frac{P(k_i | R)}{P(k_i | \bar{R})} \right] \times \left[\prod_{g_i(\vec{d}_j)=0, g_i(\vec{q})=1} \frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right]$$

73

The Probabilistic Model

- We can take the logarithm of this expression:

$$Sim(d_j, q) \approx \log \left(\left[\prod_{g_i(\vec{d}_j)=1, g_i(\vec{q})=1} \frac{P(k_i | R)}{P(k_i | \bar{R})} \right] \times \left[\prod_{g_i(\vec{d}_j)=0, g_i(\vec{q})=1} \frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right] \right)$$

$$\log(a \cdot b) = \log(a) + \log(b)$$

$$Sim(d_j, q) \approx \sum_{g_i(\vec{d}_j)=1, g_i(\vec{q})=1} \log \left(\frac{P(k_i | R)}{P(k_i | \bar{R})} \right) + \sum_{g_i(\vec{d}_j)=0, g_i(\vec{q})=1} \log \left(\frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right)$$

74

The Probabilistic Model

- First sum:
 - Ranges over keywords present in d_j and q .
 - If we extend it to range over all keywords, we have to subtract the expressions that correspond to keywords not in d_j ($w_{ij} = 0$) or keywords not q ($w_{iq} = 0$).
 - This can be obtained by multiplying with $w_{ij} * w_{iq}$ (if one of these weights is 0 the expression will evaluate to 0).
- Second sum:
 - Ranges over keywords not present in d_j but present in q .
 - We can extend it to range over all keywords not present in d_j by multiplying with w_{iq} .
 - If we extend the result to range over all keywords, we have to subtract the expressions that correspond to keywords present in d_j ($w_{ij} = 1$).
 - The subtracted expression is a sum similar to the first sum and can be extended to range over all keywords in the same way.

75

The Probabilistic Model

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \log \left(\frac{P(k_i | R)}{P(k_i | \bar{R})} \right) + \sum_{i=1}^t w_{iq} \times \log \left(\frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right) - \sum_{i=1}^t w_{ij} \times w_{iq} \times \log \left(\frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right)$$

This expression can be removed because it does not depend doc d_j

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \left[\log \left(\frac{P(k_i | R)}{P(k_i | \bar{R})} \right) - \log \left(\frac{P(\bar{k}_i | R)}{P(\bar{k}_i | \bar{R})} \right) \right]$$

76

The Probabilistic Model

$$\log(a) - \log(b) = \log(a/b)$$

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \left[\log \left(\frac{P(k_i | R)}{P(k_i | \bar{R})} \times \frac{P(\bar{k}_i | \bar{R})}{P(\bar{k}_i | R)} \right) \right]$$

$$(a/b) \times (c/d) = (a/d) \times (c/b)$$

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \left[\log \left(\frac{P(k_i | R)}{P(\bar{k}_i | R)} \times \frac{P(\bar{k}_i | \bar{R})}{P(k_i | \bar{R})} \right) \right]$$

$$\log(a \times b) = \log(a) + \log(b)$$

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \left[\log \left(\frac{P(k_i | R)}{P(\bar{k}_i | R)} \right) + \log \left(\frac{P(\bar{k}_i | \bar{R})}{P(k_i | \bar{R})} \right) \right]$$

77

The Probabilistic Model

- Final result:

$$Sim(d_j, q) \approx \sum_{i=1}^t w_{ij} \times w_{iq} \times \left[\log \left(\frac{P(k_i | R)}{1 - P(k_i | R)} \right) + \log \left(\frac{1 - P(k_i | \bar{R})}{P(k_i | \bar{R})} \right) \right]$$

$$P(\bar{k}_i | R) = 1 - P(k_i | R)$$

$$P(\bar{k}_i | \bar{R}) = 1 - P(k_i | \bar{R})$$

78

The Probabilistic Model

- How to compute the probabilities $P(k_i | R)$ and $P(k_i | \bar{R})$?
- Simplifying assumptions at the beginning (initial guess):
 - $P(k_i | R)$ is constant for all index terms k_i :

$$P(k_i | R) = 0.5$$

- \bar{R} can be approximated by the whole collection of docs:

$$P(k_i | \bar{R}) = \frac{n_i}{N}$$

n_i ← Number of docs that contain k_i
 N ← Total number of docs

- Use this initial guess to rank the documents and retrieve an initial subset of documents (example: the top r ranked documents; r a threshold).

79

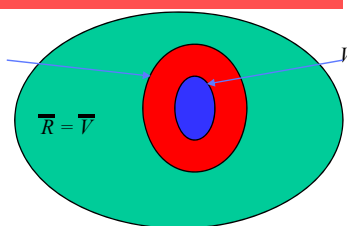
The Probabilistic Model

- Improving the initial guess:
 - Let V be a subset of the docs initially retrieved (example: all of them).
 - R can be approximated by V .
 - \bar{R} can be approximated by \bar{V} .
 - Let V_i be the subset of V consisting of documents containing the index term k_i .

$$P(k_i | R) = \frac{|V_i|}{|V|}$$

$$P(k_i | \bar{R}) = \frac{n_i - |V_i|}{N - |V|}$$

- Repeat recursively.



80

The Probabilistic Model

- Result:

- We are able to improve our guesses for $P(k_i | R)$ and $P(k_i | \bar{R})$ without any assistance from a human (contrary to the original idea).
- However, we can also use assistance from the user for definition of V as originally conceived.

- Notice:

The previous formulas for $P(k_i | R)$ and $P(k_i | \bar{R})$ pose problems for small values of $|V|$ and $|V_i|$ (example: $|V|=1$ and $|V_i|=0 \rightarrow \log(0)!$)

81

The Probabilistic Model

- Solution 1: add a constant adjustment factor

$$P(k_i | R) = \frac{|V_i| + 0.5}{|V| + 1} \quad P(k_i | \bar{R}) = \frac{n_i - |V_i| + 0.5}{N - |V| + 1}$$

- Solution 2: use n_i/N as adjustment factor

$$P(k_i | R) = \frac{|V_i| + \frac{n_i}{N}}{|V| + 1} \quad P(k_i | \bar{R}) = \frac{n_i - |V_i| + \frac{n_i}{N}}{N - |V| + 1}$$

82

The Probabilistic Model

- Example:

- Terms: $K=(k_1, k_2, k_3)$
- Query vector: $q = (1, 0, 1)$
- Documents:
 - $d_1 = (0, 1, 1)$
 - $d_2 = (1, 1, 1)$
 - $d_3 = (0, 0, 0)$
 - $d_4 = (1, 0, 1)$

- Initial ranking:

- $\text{sim}(d_1, q) =$

$$0*1*\log(0.5/0.5) + \log(0.5/0.5) + 1*0*\log(0.5/0.5) + \log(0.5/0.5) + 1*1*\log(0.5/0.5) + \log(0.25/0.75) = \log(1/3) = -1.099$$
- $\text{sim}(d_2, q) =$

$$1*1*\log(0.5/0.5) + \log(0.5/0.5) + 1*0*\log(0.5/0.5) + \log(0.5/0.5) + 1*1*\log(0.5/0.5) + \log(0.25/0.75) = \log(1/3) = -1.099$$
- $\text{sim}(d_3, q) =$

$$0*1*\log(0.5/0.5) + \log(0.5/0.5) + 0*0*\log(0.5/0.5) + \log(0.5/0.5) + 0*1*\log(0.5/0.5) + \log(0.25/0.75) = 0$$
- $\text{sim}(d_4, q) =$

$$1*1*\log(0.5/0.5) + \log(0.5/0.5) + 0*0*\log(0.5/0.5) + \log(0.5/0.5) + 1*1*\log(0.5/0.5) + \log(0.25/0.75) = \log(1/3) = -1.099$$

$$N = 4, n_1 = 2, n_2 = 2, n_3 = 3.$$

$$\begin{aligned} P(k_1|R) &= 0.5 \\ P(k_2|R) &= 0.5 \\ P(k_3|R) &= 0.5 \end{aligned}$$

$$\begin{aligned} P(k_1|\bar{R}) &= 2/4 = 0.5 \\ P(k_2|\bar{R}) &= 2/4 = 0.5 \\ P(k_3|\bar{R}) &= 3/4 = 0.75 \end{aligned}$$

83

The Probabilistic Model

- Example:

- Terms: $K=(k_1, k_2, k_3)$
- Query vector: $q = (1, 0, 1)$
- Documents:
 - $d_1 = (0, 1, 1)$
 - $d_2 = (1, 1, 1)$
 - $d_3 = (0, 0, 0)$
 - $d_4 = (1, 0, 1)$

- improving ranking:

- $\text{sim}(d_1, q) =$

$$0*1*\log(2.5/1.5) + \log(0.75/0.25) + 1*0*\log(2.5/1.5) + \log(0.75/0.25) + 1*1*\log(3.5/0.5) + \log(0.75/0.25) = \log(7*3) = \log(21) = 3.045$$
- $\text{sim}(d_2, q) =$

$$1*1*\log(2.5/1.5) + \log(0.75/0.25) + 1*0*\log(2.5/1.5) + \log(0.75/0.25) + 1*1*\log(3.5/0.5) + \log(0.75/0.25) = \log(5) + \log(21) = 4.654$$
- $\text{sim}(d_3, q) =$

$$0*1*\log(2.5/1.5) + \log(0.75/0.25) + 0*0*\log(2.5/1.5) + \log(0.75/0.25) + 0*1*\log(3.5/0.5) + \log(0.75/0.25) = 0$$
- $\text{sim}(d_4, q) =$

$$1*1*\log(2.5/1.5) + \log(0.75/0.25) + 0*0*\log(2.5/1.5) + \log(0.75/0.25) + 1*1*\log(3.5/0.5) + \log(0.75/0.25) = \log(5) + \log(21) = 4.654$$

Use solution 1:

$$\begin{aligned} P(k_1|R) &= (2+0.5)/(3+1) = 2.5/4 \\ P(k_2|R) &= (2+0.5)/(3+1) = 2.5/4 \\ P(k_3|R) &= (3+0.5)/(3+1) = 3.5/4 \\ P(k_1|\bar{R}) &= (2-2+0.5)/(4-3+1) = 0.25 \\ P(k_2|\bar{R}) &= (2-2+0.5)/(4-3+1) = 0.25 \\ P(k_3|\bar{R}) &= (3-3+0.5)/(4-3+1) = 0.25 \end{aligned}$$

84

The Probabilistic Model

- Example:

- Terms: $K=(k_1, k_2, k_3)$

- Query vector: $q = (1, 0, 1)$

- Documents:

- $d_1 = (0, 1, 1)$

- $d_2 = (1, 1, 1)$

- $d_3 = (0, 0, 0)$

- $d_4 = (1, 0, 1)$

- $N = 4, n_1 = 2, n_2 = 2, n_3 = 3.$

- suppose $V = \{d_2, d_4\}$

- $V_1 = \{d_2, d_4\}$

- $V_2 = \{d_2\}$

- $V_3 = \{d_2, d_4\}$

- improving ranking:

- $\text{sim}(d_1, q) =$

$$0 * 1 * [\log(2.5/0.5) + \log(2.5/0.5)] + \\ 1 * 0 * [\log(0.5/0.5) + \log(0.5/0.5)] + \\ 1 * 1 * [\log(2.5/0.5) + \log(0.5/0.5)] = \log(5) = 1.609$$

- $\text{sim}(d_2, q) =$

$$1 * 1 * [\log(2.5/0.5) + \log(2.5/0.5)] + \\ 1 * 0 * [\log(0.5/0.5) + \log(0.5/0.5)] + \\ 1 * 1 * [\log(2.5/0.5) + \log(0.5/0.5)] = \log(25) + \log(5) = 4.828$$

- $\text{sim}(d_3, q) =$

$$0 * 1 * [\log(2.5/0.5) + \log(2.5/0.5)] + \\ 0 * 0 * [\log(0.5/0.5) + \log(0.5/0.5)] + \\ 0 * 1 * [\log(2.5/0.5) + \log(0.5/0.5)] = 0$$

- $\text{sim}(d_4, q) =$

$$1 * 1 * [\log(2.5/0.5) + \log(2.5/0.5)] + \\ 0 * 0 * [\log(0.5/0.5) + \log(0.5/0.5)] + \\ 1 * 1 * [\log(2.5/0.5) + \log(0.5/0.5)] = \log(25) + \log(5) = 4.828$$

Use solution 1:

- $P(k_1|R) = (2+0.5)/(2+1) = 2.5/3$

- $P(k_2|R) = (1+0.5)/(2+1) = 0.5$

- $P(k_3|R) = (2+0.5)/(2+1) = 2.5/3$

- $P(k_1|\overline{R}) = (2-2+0.5)/(4-2+1) = 0.5/3$

- $P(k_2|\overline{R}) = (2-1+0.5)/(4-2+1) = 0.5$

- $P(k_3|\overline{R}) = (3-2+0.5)/(4-2+1) = 0.5$

85

The Probabilistic Model

- Advantages:

- Docs ranked in decreasing order of probability of relevance.

- Disadvantages

- Need to guess the initial separation of documents into relevant and non-relevant sets.

- Does not take into account frequency of term occurrence (binary weights).

- Assumes that terms are independent (however, it is not clear that this is bad).

86

Brief Comparison of Classic Models

- Boolean model does not provide for partial matches and is considered to be the weakest classic model
- Salton and Buckley did a series of experiments that indicate that, in general, the vector model outperforms the probabilistic model with general collections
- This seems also to be the view of the research community

87