JEPPIAAR ENGINEERING COLLEGE, CHENNAI 600 109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUB CODE:CS6007

SUB NAME: **INFORMATION RETRIEVAL**

QUESTION BANK

BATCH:2015 - 2019

YEAR/SEMESTER:IV / VII

JEPPIAAR ENGINEERING COLLEGE, CHENNAI 600 109

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2017 – 2018 (ODD SEMESTER)

SYLLABUS

CS6007 INFORMATION RETRIEVAL L T P C

3 0 0 3

UNIT I INTRODUCTION 9

Introduction -History of IR- Components of IR – Issues –Open source Search engine Frameworks – The impact of the web on IR – The role of artificial intelligence (AI) in IR – IR Versus Web Search – Components of a Search engine- Characterizing the web.

UNIT II INFORMATION RETRIEVAL 9

Boolean and vector-space retrieval models- Term weighting – TF-IDF weighting- cosine similarity – Preprocessing – Inverted indices – efficient processing with sparse vectors – Language Model based IR – Probabilistic IR –Latent Semantic Indexing – Relevance feedback and query expansion.

UNIT III WEB SEARCH ENGINE – INTRODUCTION AND CRAWLING 9

Web search overview, web structure, the user, paid placement, search engine optimization/ spam. Web size measurement – search engine optimization/spam – Web Search Architectures – crawling – meta-crawlers- Focused Crawling – web indexes –- Near-duplicate detection – Index Compression – XML retrieval.

UNIT IV WEB SEARCH – LINK ANALYSIS AND SPECIALIZED SEARCH 9

Link Analysis –hubs and authorities – Page Rank and HITS algorithms -Searching and Ranking – Relevance Scoring and ranking for Web – Similarity – Hadoop& Map Reduce – Evaluation – Personalized search – Collaborative filtering and content-based recommendation of documents and products – handling "invisible" Web – Snippet generation, Summarization, Question Answering, Cross- Lingual Retrieval.

UNIT V DOCUMENT TEXT MINING 9

Information filtering; organization and relevance feedback – Text Mining -Text classification and clustering – Categorization algorithms: naive Bayes; decision trees; and nearest neighbor – Clustering algorithms: agglomerative clustering; k-means; expectation maximization (EM).

TOTAL: 45

TEXT BOOKS:

1. C. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval , Cambridge University Press, 2008.
2. Ricardo Baeza -Yates and BerthierRibeiro – Neto, Modern Information Retrieval: The Concepts and Technology behind Search 2nd Edition, ACM Press Books 2011.
3. Bruce Croft, Donald Metzler and Trevor Strohman, Search Engines: Information Retrieval in Practice, 1st Edition Addison Wesley, 2009.
4. Mark Levene, An Introduction to Search Engines and Web Navigation, 2nd Edition Wiley, 2010.

REFERENCES:

1. Stefan Buettcher, Charles L. A. Clarke, Gordon V. Cormack, Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.
2. OphirFrieder "Information Retrieval: Algorithms and Heuristics: The Information Retrieval Series ", 2nd Edition, Springer, 2004.
3. Manu Konchady, "Building Search Applications: Lucene, Ling Pipe", and First Edition, Gate Mustru Publishing, 2008.

UNIT – I

INTRODUCTION

PART – A

QUESTIONS AND ANSWERS

1. **Define information retrieval.(nov/dec 2016)**
   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).

2. **What are the applications of IR?**
   - Indexing
   - Ranked retrieval
   - Web search
   - Query processing

3. **Give the historical view of Information Retrieval.**
   - Boolean model, statistics of language (1950's)
   - Vector space model, probabilistic indexing, relevance feedback (1960's)
   - Probabilistic querying (1970's)
   - Fuzzy set/logic, evidential reasoning (1980's)
   - Regression, neural nets, inference networks, latent semantic indexing, TREC (1990's)

4. **What are the components of IR?(nov/dec 2016)**
   - The document subsystem
   - The indexing subsystem
   - The vocabulary subsystem
   - The searching subsystem
   - The ser-system interface
   - The matching subsystem

5. **How to AI applied in IR systems?(nov/dec 2016)**

   Four main roles investigated

   - Information characterisation
   - Search formulation in information seeking
   - System Integration

- Support functions

6. How to introduce AI into IR systems?

- User simply enters a query, suggests what needs to be done, and the system executes the query to return results.
- First signs of AI. System actually starts suggesting improvements to user.
- Full Automation. User queries are entered and the rest is done by the system.

7. What are the areas of AI for information retrieval?

- Natural language processing
- Knowledge representation
- Machine learning
- Computer Vision
- Reasoning under uncertainty
- Cognitive theory

8. Give the functions of information retrieval system.

- To identify the information(sources) relevant to the areas of interest of the target users community
- To analyze the contents of the sources(documents)
- To represent the contents of the analyzed sources in a way that will be suitable for matching user's queries
- To analyze user's queries and to represent them in a form that will be suitable for matching with the database
- To match the search statement with the stored database
- To retrieve the information that is relevant
- To make necessary adjustments in the system based on feedback form the users.

9. List the issues in information retrieval system.

- Assisting the user in clarifying and analyzing the problem and determining information needs.
- Knowing how people use and process information.
- Assembling a package of information that enables group the user to come closer to a solution of his problem.
- Knowledge representation.
- Procedures for processing knowledge/information.
- The human-computer interface.
- Designing integrated workbench systems.

- Designing user-enhanced information systems.
- System evaluation.

10. What are some open source search frameworks?

- Google Search API
- Apache Lucene
- blekko API
- Carrot2
- Egothor
- Nutch

11. Define relevance.

Relevance appears to be a subjective quality, unique between the individual and a given document supporting the assumption that relevance can only be judged by the information user.Subjectivity and fluidity make it difficult to use as measuring tool for system performance.

12. What is meant by stemming?

Stemming is techniques used to find out the root/stem of a word. Used to improve effectiveness of IR and text mining.Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

13. Define indexing & document indexing.

Association of descriptors (keywords, concepts, metadata) to documents in view of future retrieval.

Document indexing is the process of associating or tagging documents with different "search" terms. Assign to each document (respectively query) a descriptor represented with a set of features, usually weighted keywords, derived from the document (respectively query) content.

14. Discuss the impact of IR on the web.

The impacts of information retrieval on the web are influenced in the following areas.

- Web Document Collection
- Search Engine Optimization
- Variants of Keyword Stuffing
- DNS cloaking: Switch IP address
- Size of the Web
- Sampling URLs
- Random Queries and Searches

15. List Information retrieval models.(nov/dec 2016)

- Boolean model
- Vector space model
- Statistical language model

16. Define web search and web search engine.

Web search is often not informational -- it might be navigational (give me the url of the site I ant to reach) or transactional (show me sites where I can perform a certain transaction, e.g. shop, download a file, or find a map).

Web search engines crawl the Web, downloading and indexing pages in order to allow full-text search. There are many general purpose search engines; unfortunately none of them come close to indexing the entire Web. There are also thousands of specialized search services that index specific content or specific sites.

17. What are the components of search engine?

Generally there are three basic components of a search engine as listed below:

1. Web Crawler
2. Database
3. Search Interfaces

18. Define web crawler.

This is the part of the search engine which combs through the pages on the internet and gathers the information for the search engine. It is also known as spider or bots**.** It is a software component that traverses the web to gather information.

19. What are search engine processes?

Indexing Process

- Text acquisition
- Text transformation
- Index creation

Query Process

- User interaction
- Ranking
- Evaluation

20. How to characterize the web?

Web can be characterized by three forms

- Search engines -AltaVista
- Web directories -Yahoo
- Hyperlink search-Web Glimpse

21. What are the challenges of web?

- Distributed data
- Volatile data
- Large volume
- Unstructured and redundant data
- Data quality
- Heterogeneous data

PART – B

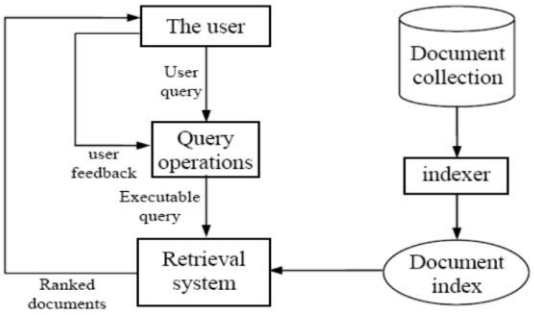
QUESTIONS AND ANSWERS

1. Write about history of Information Retrieval.

- Early keyword-based engines ca. 1995-1997
  - Altavista, Excite, Infoseek, Inktomi, Lycos
- 1998+: Link-based ranking pioneered by Google
  - Blew away all early engines save Inktomi
- 2005+: Google gains search share, dominating in Europe and very strong in North America
  - 2009: Yahoo! and Microsoft propose combined paid search offering

2. **Explain the Information Retrieval. (nov/dec 2016)**

IR helps users find information that matches their information needs expressed as queries.
Historically, IR is about document retrieval, emphasizing document as the basic unit.
– Finding documents relevant to user queries.



Architecture of Information Retrieval

IR Queries

• Keyword queries
• Boolean queries (using AND, OR, NOT)
• Phrase queries
• Proximity queries
• Full document queries
• Natural language questions

IR Models

– Boolean model
– Vector space model
– Statistical language model

3. Discuss the influence of AI in Information Retrieval.

Areas of AI for IR

- Natural language processing
- Knowledge representation
  - ❖ Expert systems
  - ❖ Ex: Logical formalisms, conceptual graphs, etc
- Machine learning
  - ❖ Short term: over a single session
  - ❖ Long term: over multiple searches by multiple users
- Computer Vision
  - ❖ Ex: OCR
- Reasoning under uncertainty
  - ❖ Ex: Dempster-Shafer, Bayesian networks, probability theory, etc
- Cognitive theory
  - ❖ Ex: User modelling

AI applied to IR

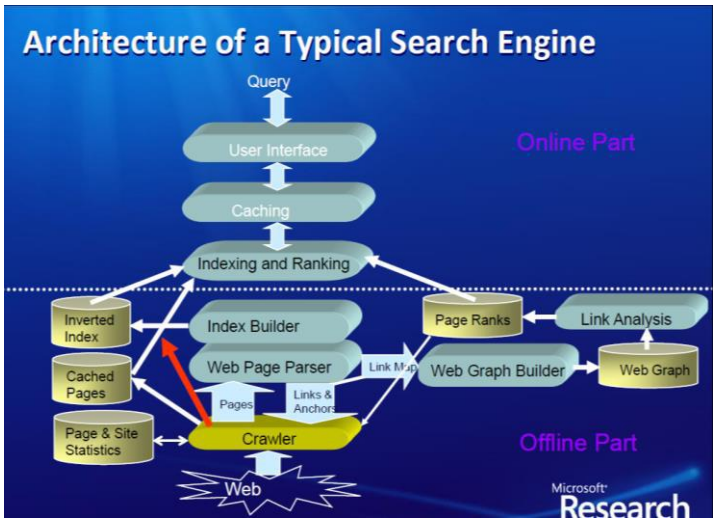
- Four main roles investigated
  - ❖ Information characterisation
  - ❖ Search formulation in information seeking
  - ❖ System Integration
  - ❖ Support functions
- AI has a very valuable contribution to make
  - ❖ Specialised systems where domain is controlled, well-integrated and understood
  - ❖ Support functions
  - ❖ Case-based reasoning and dialogue functions
  - ❖ Integrated functions

4. Explain in detail about Search Engine.

Search Engine is in the field of IR .Searching authors, titles and subjects in library card catalogs or computers. Document classification and categorization, user interfaces, data visualization, filtering

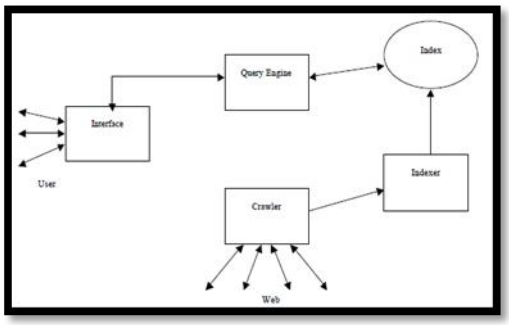
Types of Search Engines

- Search by Keywords (e.g. AltaVista,
- Excite, Google, and Northern Light)
- Search by categories (e.g. Yahoo!)
- Specialize in other languages (e.g.
- Chinese Yahoo! and Yahoo! Japan)
- Interview simulation (e.g. Ask Jeeves!)

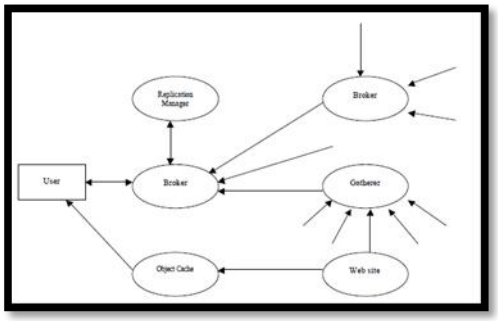


Search Engine Architectures

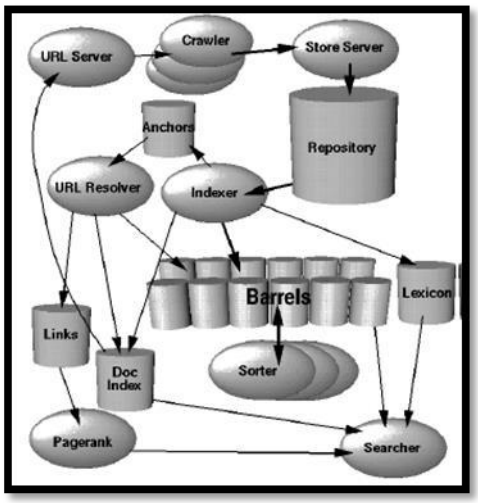
- AltaVista
- Harvest
- Google



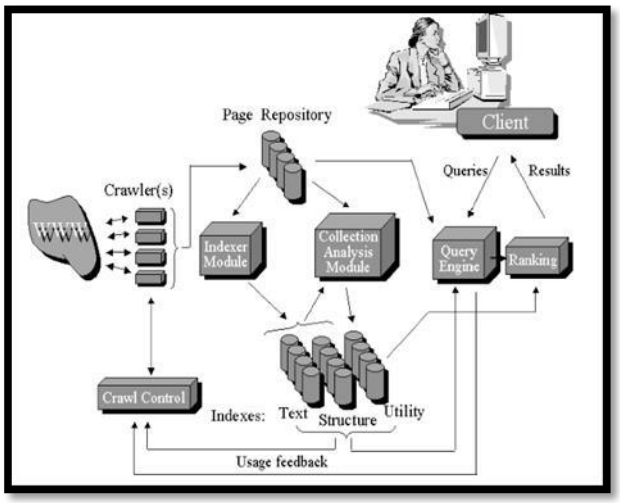
AltaVista Architecture



Harvest Architecture



Google Architecture



Modern Search Engine

5. Discuss web information retrieval system.

Web Search Engine Evolution

- Web Search 1.0 –Traditional Text Retrieval
- Web Search 2.0 –Page-level Relevance Ranking
- Next Generation Web Search

Web Analysis and Its Relationship to IR

- Goals of Web analysis:
  - ❖ Improve and personalize search results relevance
  - ❖ Identify trends
- Classify Web analysis:
  - ❖ **Web content analysis**
  - ❖ **Web structure analysis**
  - ❖ **Web usage analysis**
- Searching the Web
- Analyzing the Link Structure of Web Pages
- Web Content Analysis

Trends in Information Retrieval

Faceted search

- ❖ Allows users to explore by filtering available information
- ❖ **Facet :**Defines properties or characteristics of a class of objects

- **Social search**
  - New phenomenon facilitated by recent Web technologies: **collaborative social search**, **guided participation**

UNIT II

INFORMATION RETRIEVAL

PART – A

QUESTIONS AND ANSWERS

1.What are the three classic models in information retrieval system?

1.Boolean model

2.Vector Space model

3.Probabilistic model

2.What is the basis for boolean model?

Simple model based on set theory and Boolean algebra

- Documents are sets of terms
- Queries are specified as Boolean expressions on terms

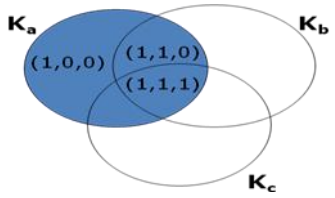
3.How can we represent the queries in boolean model?

Queries specified as boolean expressions

- Precise semantics
- Neat formalism
- q = ka ∧(kb ∨¬kc)

4.Definition of boolean model?

Index term weight variables all are binary



- $w_{ij} \varepsilon \{0,1\}$
- Query $q = k_a \wedge (k_b \vee \neg k_c)$
- $sim(q_i,d_j) =$ 1 , i.e. doc's are relevant

0, otherwise i.e. doc's are not relevant

$q = k_a \wedge (k_b \vee \neg k_c)$, can be written as disjunctive normal form,

$vec(q_{dnf}) = (1,1,1)\ v\ (1,1,0)\ v\ (1,0,0)$

5. What are the advantages of Boolean model?

- Clean Formalism
- Easy to implement
- Intuitive concept
- Still it is dominant model for document database systems

6.What are the disadvantages of Boolean model?

Exact matching may retrieve too few or too many documents

- Difficult to rank output, some documents are more important than others
- Hard to translate a query into a Boolean expression
- All terms are equally weighted
- More like data retrieval than information retrieval
- No notion for partial matching

7.Define the Vector Model

This model recognizes that the Use of binary weights is too limiting and proposes a framework in which partial matching is possible.

- Non-binary weights provide consideration for partial matches
- These term weights are used to compute a degree of similarity**between a query and each document**
- Ranked set of documents provides for better matching
  - $w_{i,j} >= 0$ associated with the pair (ki,dj)
  - $vec(d_j) = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$
  - $W_{i,q} >= 0$ associated with the pair (k_i,q)
  - $vec(q) = (w_{1,q}, w_{2,q}, ..., w_{t,q})$ $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$
  - t- total no. Of index terms in the collection

- Sim(d_j, q) =

$$\frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| x \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{i,j} \, x \, w_{i,q})}{\sqrt{\sum_{i=1}^{t} w_{i,j}^{2} \, x \sum_{j=1}^{t} w_{i,q}^{2}}}$$

8. What are the advantages of**Vector Model?**

- ☐ Simple model based on linear algebra
- ☐ Term weights not binary
- ☐ Allows computing a continuous degree of similarity between queries and documents
- ☐ Allows ranking documents according to their possible relevance
- ☐ Allows partial matching
- ☐ Allows efficient implementation for large document collections

8. What are the disadvantages ofVector Model?

- Index terms are assumed to be mutually independent
- Search keywords must precisely match document terms
- Long documents are poorly represented
- The order in which the terms appear in the document is lost in the vector space representation
- Weighting is intuitive, but not very formal

9. What are the Parameters in calculating a weight for a document term or query term?

Term Frequency (tf): Term Frequency is the number of times a term i appears in document j (tfij)

– Document Frequency (df): Number of documents a term i appears in, (dfi).

– Inverse Document Frequency (idf): A discriminating measure for a term i in collection, i.e., how discriminating term i is. (idf i) = log10(n / dfi), where n is the number of document

10. How can you calculate tf and idf in vector model?

- The normalized frequency (term factor) $f_{i,j}$is,

$f_{i,j} = freq_{i,j} / max_l freq_{l,j}$; if k_i not appear in d_j then $f_{i,j} = 0$;

- Inverse document frequency (idf) is

  $idf_i = log(N/n_i)$ or $idf_i = log(D/df_i)$

  Where

- N- total no.of documents in the collection
- n_i – no.of documents in which the index terms k_i appears
- $freq_{i,j}$ – frequency of the term k_i in the document d_j
- max_l – maximum over all terms frequencies

11.How do you calculate the term weighting in document and Query term weight ?**(nov/dec 2016)**

- Trem weighting is,
- $w_{i,j} = tf * idf_i$ i.e. $wij = f_{i,j} * log(N/n_i)$
- Query term weight is,
- $w_{i,q} = (0.5 + 0.5 * freq_{i,q} / max_l freq_{l,q}) * log(N/n_i)$

12.Write the cosine similarity function for vector space model:-

Cosine $\theta = \frac{Q.D}{|Q|.|D|}$

$$\therefore \text{Sim}(Q,D_i) = \frac{\sum_i w_{Q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \sqrt{\sum_i w_{i,j}^2}}$$

13.Define Probabilistic model or Binary Independence Retrieval :-

The Objectiveof Probabilistic model is to capture the IR problem using a probabilistic framework

Given a user query, there is an ideal answer set

- Querying as specification of the properties of this ideal answer set
- Definition
- Weight variables all are binary, i.e. $w_{i,j} \in \{0,1\}$ and $w_{i,q} \in \{0,1\}$
- q- a query is a subset of index terms

- R – set of doc's known (initial guess) to be relevant
- R – the complement of R, i.e. the set of non-relevant doc's
- $P(R|d_j)$ – probability of d_j relevant to q
- $P(\overline{R}|d_j)$ - probability of d_j non-relevant to q
- $sim(d_j,q) = P(R|d_j) / P(\overline{R}|d_j)$

14.What are the Fundamental assumptions for probabilistic principle?

- q- user query,d_j – doc in the collections
- Model assumes, relevance depends on the query and the doc representation only
  - R – ideal answer set, relevant to the query
  - R - ideal answer set, non-relevant to the query
  - Similarity to the query ratio is, i.e. probabilistic ranking computed as
  - Ratio = P(d_j relevant-to q) / P(d_j non-relevant-to q)
  - The rank minimizes the probability of the erroneous judgment

15. How can you find similarity between doc and query in probabilistic principleUsing Bayes' rule?

$sim(d_j,q) = P(d_j|R) \times P(R) / P(d_j|\overline{R}) \times P(\overline{R})$

where

$P(d_j|R)$ - probability of randomly selecting the document d_j from the set R of relevant documents

- $P(R)$ - probability of randomly selecting the document from the entire collection is relevant

$P(\overline{R})$

The meaning of $P(d_j|R)$ and $P(R)$ are analogous and complementary

- Since $P(R)$ and $P(\overline{R})$ are same for all doc's in the collection, then we write,

$$sim(d_j,q) \sim P(d_j|R) / P(d_j|\overline{R})$$

16.Write the advantages and disadvantages of probabilistic model:

- Advantages

- Doc's are ranked in decreasing order of their probability of relevant

- Disadvantages
  - Need to guess the initial separation of doc's into relevant and non-relevant sets
  - All weights are binary
  - The adoption of the independence assumption for index terms
  - need to guess initial estimates for P(ki | R)
  - method does not take into account tf and idf factors

17. Why theClassic IR might lead to poor retrieval?
    - The user information need is more related to concepts and ideas than to index terms but in classic IR
    - Unrelated documents might be included in the answer set
    - Relevant documents that do not contain at least one index term are not retrieved
    - Reasoning: retrieval based on index terms is vague and noisy

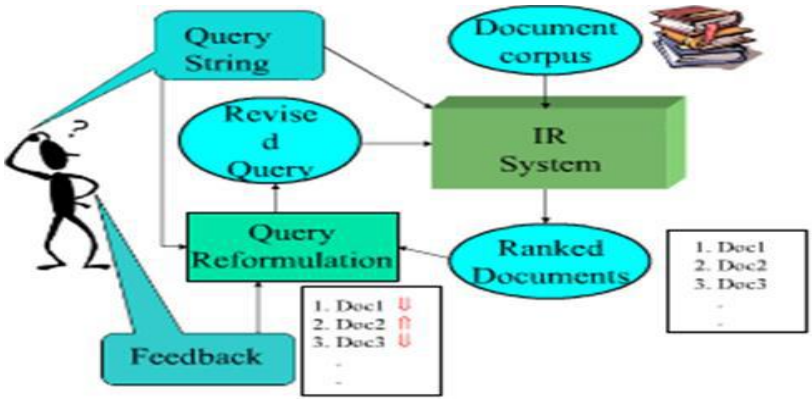
18. Definitions Latent Semantic Indexing Model:-
    - Let t be the total number of index terms
    - Let N be the number of documents
    - Let vec(M) = M_{ij} be a term-document matrix with t -rows and N -columns
    - To each element of this matrix is assigned a weight w_{ij} associated with the pair $[k_i,d_j]$
    - The weight w_{ij} can be based on a tf-idf weighting scheme

19. Write the advantages of Latent Semantic Indexing Model?
    - Latent semantic indexing provides an interesting conceptualization of the IR problem
    - It an efficient indexing scheme for the documents in te collection
    - It provides,
      - Elimination of noise
      - Removal of redundancy

20. Define Relevance feedback model:-**(nov/dec 2016)**

After initial retrieval results are presented allow the user to provide feedback on the relevance of one or more of the retrieved documents. use this feedback information to reformulate the query and produce new results based on reformulated query. Thus allows more interactive multi pass process.

21.Draw the flow diagram for relevance feedback query processing model:**(nov/dec 2016)**



22. Write the types of queries:

There are 4 type of queries such as Structured queries,Pattern matching queries,Boolean queries,Context Queries

23.Give short notes for User Relevance Feedback:

It is the most popular query formulation strategy. In a relevance feed backcycle ,the user presented with a list of the retrieved documents .Then examine them, marks those which are relevant

Only to 10 (or 20) ranked documents are examined

- Selecting important terms, or expression, attached to the documents
- Enhancing the important of these terms in a new query formulation
  - The new query will be
  - 1.Moved towards the relevant documents,2.Away from the non-relevant ones.

24.What are the two basic approaches in User Relevance Feedback for query processing?

1)Query expansion- Expand queries with the vector model

2)Term reweighting –

i)Reweight query terms with the probabilistic model

ii)Reweight query terms with a variant of the probabilistic model

25. What are the Advantages of User Relevance Feedback method?

- It shields the user from the details of the query reformulation process because all the user has to provide is a relevance judgement on documents
- It breaks down the whole searching task into a sequence of small steps which are easier to grasp
- It provides a controlled process designed to emphasize some terms (relevant ones) and de-emphasize others (non-relevant ones)

26.What are the three classic and similar ways to calculate the modified query q_m?

Standard_Rochio : $$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

Ide_Regular : $$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

Ide_Dec_Hi : $$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{\text{non-relevant}}(\vec{d}_j)$$

27.What are the advantages and disadvantages of query processing?

Advantages :

It is simple:

1)The fact that the modified term weights are computed directly from the set of retrieved documents

2)It gives good results:

Observed experimentally and are due to the fact that the modified query vector does reflect a portion of the intended query semantics

Disadvantages

No optimality

PART – B

QUESTIONS AND ANSWERS

1. Explain in detail about vector-space retrieval models with an example:-
   - Use of binary weights is too limiting
   - Non-binary weights provide consideration for partial matches
   - These term weights are used to compute a degree of similarity between a query and each document
   - Ranked set of documents provides for better matching
   - Define:
     - $w_{i,j} >= 0$ associated with the pair (ki,dj)
     - $vec(d_j) = (w_{1,j}, w_{2,j}, ..., w_{t,j})$
     - $W_{i,q} >= 0$ associated with the pair (k_i,q)
     - $vec(q) = (w_{1,q}, w_{2,q}, ..., w_{t,q})$
     - t- total no. Of index terms in the collection
   - Use of binary weights is too limiting
   - Non-binary weights provide consideration for partial matches
   - These term weights are used to compute a degree of similarity between a query and each document
   - Ranked set of documents provides for better matching
   - Define:
     - $w_{i,j} >= 0$ associated with the pair (ki,dj)
     - $vec(d_j) = (w_{1,j}, w_{2,j}, ..., w_{t,j})$
     - $W_{i,q} >= 0$ associated with the pair (k_i,q)
     - $vec(q) = (w_{1,q}, w_{2,q}, ..., w_{t,q})$
     - t- total no. Of index terms in the collection
   - Definition
     - N- total no.of documents in the collection
     - n_i – no.of documents in which the index terms $k_i$appears
     - $freq_{i,j}$ – frequency of the term k_i in the document d_j
     - max_l– maximum over all terms frequencies
     - The normalized frequency (term factor) $f_{i,j}$is,
     - **$f_{i,j} = freq_{i,j} / max_l freq_{l,j}$**; if k_i not appear in d_j then $f_{i,j} = 0$;
     - Inverse document frequency (idf)
     - **$idf_i = log(N/n_i)$ or $idf_i = log(D/df_i)$**
     - Trem weighting is,
     - **$w_{i,j} = tf * idf_i$ i.e. $wij = f_{i,j} * log(N/n_i)$**
     - Query term weight is,
     - **$w_{i,q} = (0.5 + 0.5 * freq_{i,q} / max_l freq_{l,q}) * log(N/n_i)$**

2. Explain about Boolean model for IR:
    - Simple model based on set theory and Boolean algebra
        - Documents are sets of terms
        - Queries are Boolean expressions on terms
    - Historically the most common model
        - Library OPACs
        - Dialog system
        - Many web search engines, too
    - Queries specified as boolean expressions
        - Precise semantics
3. Neat formalism
4. q = ka ∧(kb ∨¬kc)

- Terms are either present or absent. Thus, wij ε {0,1}
- There are three conectives used: and, or, not

☐ **D:** set of words (indexing terms) present in a document
  ■ each term is either present (1) or absent (0)

☐ **Q:** A Boolean expression
  ■ terms are index terms
  ■ operators are AND, OR, and NOT

☐ **F:** Boolean algebra over sets of terms and sets of documents

☐ **R:** a document is predicted as relevant to a query expression if it satisfies the query expression

((text∨ information) ∧ retrieval ∧¬theory)

☐ Each query term specifies a set of documents containing the term

☐ AND (∧): the intersection of two sets

☐ OR (∨): the union of two sets

☐ NOT (¬): set inverse, or really set difference

☐ Definition
  ■ Index term weight variables all are binary
  ■ w_{ij}ε {0,1}
  ■ Query $q = k_a \wedge (k_b \vee \neg k_c)$
  ■ $sim(q_i,d_j)$ = 1 , i.e. doc's are relevant
    0, otherwise i.e. doc's are not relevant

☐ $q = k_a \wedge (k_b \vee \neg k_c)$, can be written as
  disjunctive normal form,
  $vec(q_{dnf})$= (1,1,1) v (1,1,0) v (1,0,0)

3.Explain about Probabilistic IR:

- Assuming independence index terms,
  - $sim(d_j,q) \sim \frac{[\Pi P(k_i \mid R)] * [\Pi P(\neg k_i \mid R)]}{[\Pi P(k_i \mid \neg R)] * [\Pi P(\neg k_i \mid \neg R)]}$
- $P(k_i \mid R)$: probability that the index term k_i is present in a document randomly selected from the set R of relevant documents
- Taking logaritms, recalling that $P(k_i|R)+P(\neg k_i|R) = 1$
  - $sim(d_j,q) \sim \log \frac{[\Pi P(k_i \mid R)] * [\Pi P(\neg k_i \mid R)]}{[\Pi P(k_i \mid \neg R)] * [\Pi P(\neg k_i \mid \neg R)]}$

$$sim(d_j,q) \sim \sum_{i=1}^{t} w_{i,q} * w_{i,j} * \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \neg R)}{P(k_i \mid \neg R)}\right)$$

- Which is a key expression for ranking computation in the probabilistic model
  - =>Improving the Initial Ranking

4. Explain about Inverted indices, efficient processing with sparse vectors

5. Explain about Latent Semantic Indexing method:

- Definitions
  - Let t be the total number of index terms
  - Let N be the number of documents
  - Let vec(M) = M_{ij} be a term-document matrix with t -rows and N -columns
  - To each element M_{ij} of this matrix is assigned a weight w_{ij}associated with the pair $[k_i,d_j]$
  - The weight w_{ij} can be based on a tf-idf weighting scheme, like Vector model
- The matrix vec(M) can be decomposed into 3 matrices (singular value decomposition) as follows:
  - $(M_{ij}) = (K) (S) (D)^t$
  - (K) is the matrix of eigenvectors derived from the term-term correlation matrix given by $(M)(M)^t$
  - $(D)^t$ is the matrix of eigenvectors derived from the transpose of the doc-doc matrix given by $(M)^t(M)$
  - (S) is an r x r diagonal matrix of singular values

    Where, r = min(t,N) that is, the rank of (M_{ij})

- In the matrix (S), select only the s largest singular values
  - Keep the corresponding columns in (K) and $(D)^t$ i.e. The remaining singular values of the S ae deleted.
  - The resultant matrix is called $(M)_s$ and is given by

    $$M_s = K_s S_s D_s^t$$

    where s, $s < r$, is the dimensionality of a reduced concept space
    - The parameter, s should be
      - large enough to allow fitting all the structure in the real data
      - small enough to allow filter out the non-relevant representational details (i.e. based on index-term representation)

6.Give brief notes about user Relevance feedback method and how it is used in query expansion:It is the most popular query formulation strategy

In a relevance feedback cycle,

- The user presented with a list of the retrieved documents
- Then examine them, marks those which are relevant
- Only to 10 (or 20) ranked documents are examined
- Selecting important terms, or expression, attached to the documents
- Enhancing the important of these terms in a new query formulation
- The new query will be
  - Moved towards the relevant documents
  - Away from the non-relevant ones

- Two basic approaches are,
  - Query expansion
  - Term reweighting

7. Write the advantages and disadvantages for classic models which are used in IR and discriminate their techniques:

a. Boolean model ,vector model , Probabilistic IR advantage and disadvantages

b. Techniques

8. Write the formal characterization of IR Models:

- Ranking algorithms are at the core of IR systems
- A ranking algorithm operates according to basic premises regarding notation of the relevance
- We should state clearly what exactly an IR Model is
  - "An IR Model is a quadruple [D, Q, f ,$R(q_i,d_j)$]"
  - Where,
    - D – a set composed of logical views for the documents in the collection
    - Q – a set composed of logical views for the user information needs – queries
    - f – a framework for modeling doc representations, queries and their relationships
    - $R(q_i,d_j)$ – a ranking function, q_i € Q and d_j € D, ranking based on q_i
- To build the model
  - To represent the document and user information need
  - From these to form a framework in which they can be modeled
  - This framework used for constructing ranking function

9. Sort and rank the documents in descending order according to the similarity values:

Suppose we query an IR system for the query "**gold silver truck**"

The database collection consists of three documents (D = 3) with the following content,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"

Answer:

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$											
Query, Q: "gold silver truck" D_1: "Shipment of gold damaged in a fire" D_2: "Delivery of silver arrived in a silver truck" D_3: "Shipment of gold arrived in a truck" D = 3; IDF = log(D/df_i)											
		Counts, tf_i						Weights, $w_i = tf_i * IDF_i$			
Terms	Q	D_1	D_2	D_3	df_i	D/df_i	IDF_i	Q	D_1	D_2	D_3
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761

$$|D_1| = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.5173} = 0.7192$$

$$|D_2| = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2} = \sqrt{1.2001} = 1.0955$$

$$|D_3| = \sqrt{0.1761^2 + 0.1761^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.1240} = 0.3522$$

$$\therefore |D_i| = \sqrt{\sum_i w_{i,j}^2}$$

$$|Q| = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = \sqrt{0.2896} = 0.5382$$

$$\therefore |Q| = \sqrt{\sum_i w_{Q,j}^2}$$

$$Q \bullet D_1 = 0.1761 * 0.1761 = 0.0310$$

$$Q \bullet D_2 = 0.4771 * 0.9542 + 0.1761 * 0.1761 = 0.4862$$

$$Q \bullet D_3 = 0.1761 * 0.1761 + 0.1761 * 0.1761 = 0.0620$$

$$\therefore Q \bullet D_i = \sum_i w_{Q,j} w_{i,j}$$

$$\text{Cosine}\,\theta_{D_1} = \frac{Q \bullet D_1}{|Q| * |D_1|} = \frac{0.0310}{0.5382 * 0.7192} = 0.0801$$

$$\text{Cosine}\,\theta_{D_2} = \frac{Q \bullet D_2}{|Q| * |D_2|} = \frac{0.4862}{0.5382 * 1.0955} = 0.8246$$

$$\text{Cosine}\,\theta_{D_3} = \frac{Q \bullet D_3}{|Q| * |D_3|} = \frac{0.0620}{0.5382 * 0.3522} = 0.3271$$

$$\therefore \text{Cosine}\,\theta_{D_i} = \text{Sim}(Q, D_i)$$

$$\therefore \text{Sim}(Q, D_i) = \frac{\sum_i W_{Q,j} W_{i,j}}{\sqrt{\sum_j W_{Q,j}^2} \sqrt{\sum_i W_{i,j}^2}}$$

- Finally we sort and rank the documents in descending order according to the similarity values

- Rank 1: Doc 2 = 0.8246
  Rank 2: Doc 3 = 0.3271
  Rank 3: Doc 1 = 0.0801

UNIT-III

WEB SEARCH-LINK ANALYSIS AND SPECIALIZED SEARCH

PART – A

QUESTIONS AND ANSWERS

1. Define web search engine?

A **web search engine** is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as search engine results pages (SERPs).

2.What are the Practical Issues in the Web?

Security Commercial transactions 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 Scanning, optical character recognition (OCR), and cross-language retrieval

3. What are the Main challenges posed by Web?

data-centric: related to the data itself

- distributed data
- high percentage of volatile data
- large volume of data
- unstructured and redundant data
- quality of data
- heterogeneous data

interaction-centric: related to the users and their interactions

- expressing a query
- interpreting results

User key challenge

- to conceive a good query

System key challenge

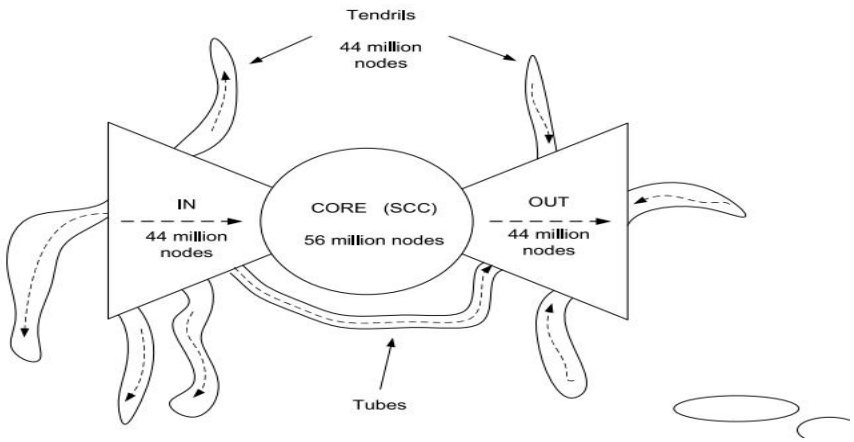
- to do a fast search and return relevant answers, even to poorl y formulated queries
- Structure of the Web Graph

4. How Web can be viewed as a graph?

- where the nodes represent individual pages

- the edges represent links between pages

5. Draw bow-tie structure of the Web?



6.Define power law?

Power Law: function that is invariant to scale changes

$$f(x) = \frac{a}{x^{\alpha}} \quad \text{with} \quad \alpha > 0$$

Depending on value of α, moments of distribution will be finite or not

$\alpha \leq 2$: average and all higher-order moments are infinite

$2 < \alpha \leq 3$: mean exists, but variance and higher-order moments are infinite

7. Define Logarithmic Normal distribution

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(\ln x - \mu)^2/2\sigma^2}$$

where

- x is the document size
- average size: $\mu = 9.357$ (in a sample)
- standard deviation: $\sigma = 1.318$ (in a sample)

8. Define Pareto distribution

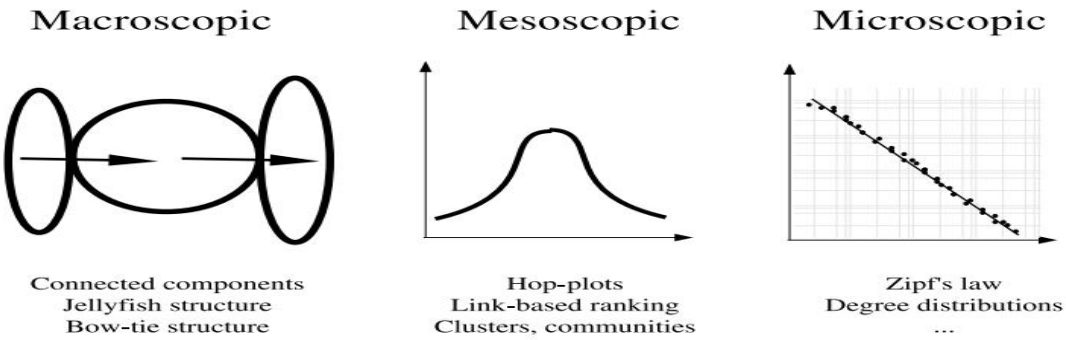
$$p(x) = \frac{\alpha k^{\alpha}}{x^{1+\alpha}}$$

where

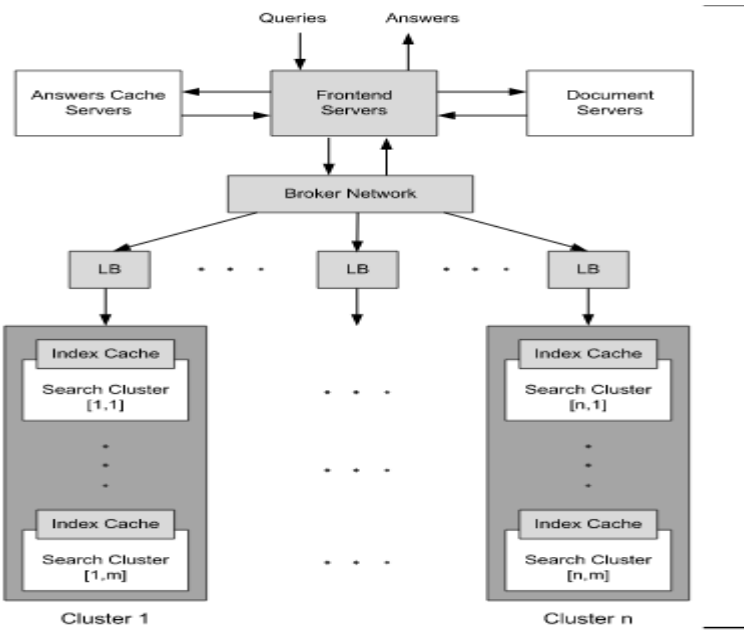
- x is measured in bytes
- k and α are parameters of the distribution

9. What are the levels of link analysis?

- microscopic level: related to the statistical properties of links and individual nodes
- mesoscopic level: related to the properties of areas or regions of the Web
- macroscopic level: related to the structure of the Web at large



10. Draw the Cluster-based Architecture?



11. Define a Web Crawler?

A Web Crawler is a software for downloading pages from the Web.

12. What are the Cycle of a Web crawling process?

- The crawler start downloading a set of seed pages, that are parsed and scanned for new links

- The links to pages that have not yet been downloaded are added to a central queue for download later
- Next, the crawler selects a new page for download and the process is repeated until a stop criterion is met.

13. List the Applications of a Web Crawler?

- create an index covering broad topics (general Web search)
- create an index covering specific topics (vertical Web search)
- archive content (Web archival)
- analyze Web sites for extracting aggregate statistics (Web characterization)
- keep copies or replicate Web sites (Web mirroring)
- Web site analysis

14. What are the Types of Web search based on crawling?(nov/dec 2016)

- General Web search: done by large search engines
- Vertical Web search: the set of target pages is delimited by a topic, a country or a language

15. What is the Main problem of focused crawling?

To predict the relevance of a page before downloading the page

16. What are the basic rules for Web crawler operation are?

- A Web crawler must identify itself as such, and must not pretend to be a regular Web user
- A Web crawler must obey the robots exclusion protocol (robots.txt)
- A Web crawler must keep a low bandwidth usage in a given Web site.

17. What are the Indexing Issues?

- Availability and speed
  - Most search engines will cache the page being referenced.
- Multiple search terms
  - OR: separate searches concatenated
  - AND: intersection of searches computed.
  - Regular expressions not typically handled.
- Parsing
  - Must be able to handle malformed HTML, partial documents

18. Why compression need?

- Use less disk space (saves money)

- Keep more stuff in memory (increases speed)
- Increase speed of transferring data from disk to memory (increases speed)
    - [read compressed data and decompress] is faster than [read uncompressed data]
- Premise: Decompression algorithms are fast
    - True of the decompression algorithms we use
- In most cases, retrieval system runs faster on compressed postings lists than on uncompressed postings lists.

19. Define Lossless vs. lossy compression

- Lossless compression: All information is preserved.
    - What we mostly do in IR.
- Lossy compression: Discard some information

PART – B

QUESTIONS AND ANSWERS

1.Briefly explain web search architectures?

Search Engine refers to a huge database of internet resources such as web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web.User can search for any information by passing query in form of keywords or phrase. It then searches for relevant information in its database and return to the user.

Search Engine Components

Generally there are three basic components of a search engine as listed below:

1. Web Crawler
2. Database
3. Search Interfaces

Web crawler

It is also known as **spider** or **bots.** It is a software component that traverses the web to gather information.

Database

All the information on the web is stored in database. It consists of huge web resources.

Search Interfaces

This component is an interface between user and the database. It helps the user to search through the database.

Search Engine Working

Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search. Following are the steps that are performed by the search engine:

- The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
- It then uses software to search for the information in the database. This software component is known as web crawler.

- Once web crawler finds the pages, the search engine then shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.

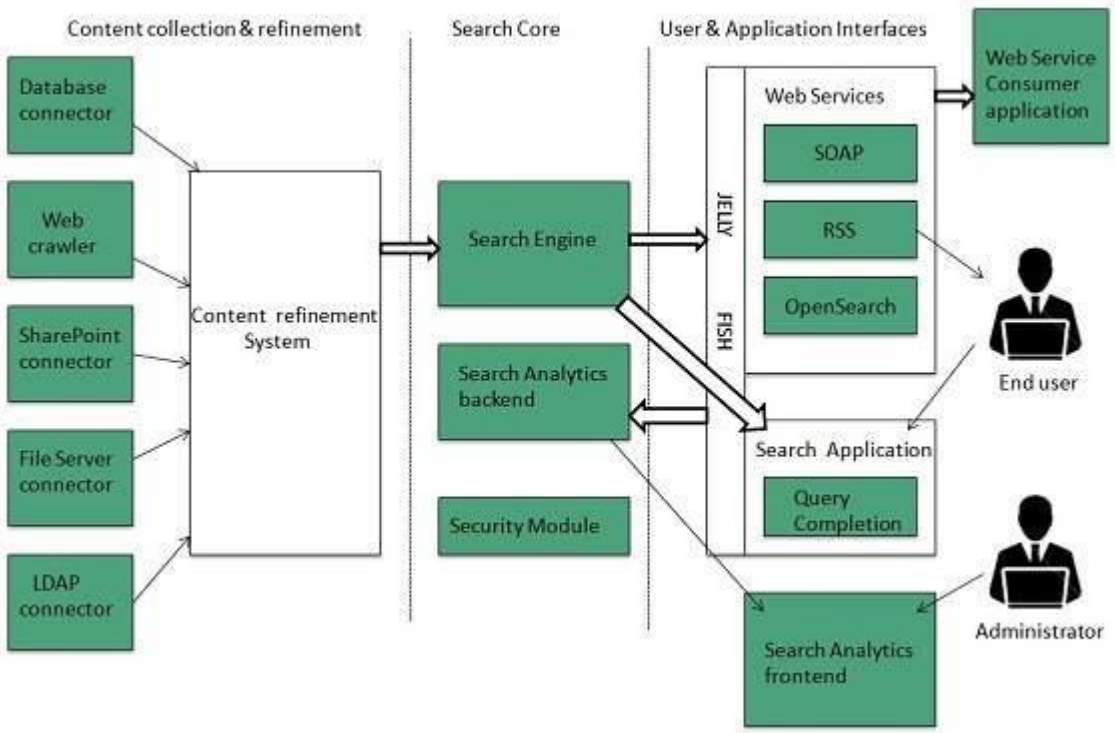
These search criteria may vary from one search engine to the other. The retrieved information is ranked according to various factors such as frequency of keywords, relevancy of information, links etc.

- User can click on any of the search results to open it.

Architecture

The search engine architecture comprises of the three basic layers listed below:

- Content collection and refinement.
- Search core
- User and application interfaces



Search Engine Processing

Indexing Process

Indexing process comprises of the following three tasks:

- Text acquisition
- Text transformation
- Index creation

Text acquisition

It identifies and stores documents for indexing.

Text Transformation

It transforms document into index terms or features.

Index Creation

It takes index terms created by text transformations and create data structures to suport fast searching.

Query Process

Query process comprises of the following three tasks:

- User interaction
- Ranking
- Evaluation

User interaction

It supporst creation and refinement of user query and displays the results.

Ranking

It uses query and indexes to create ranked list of documents.

Evaluation

It monitors and measures the effectiveness and efficiency. It is done offline.

Examples

Following are the several search engines available today:

Search Engine	Description
Google	It was originally called **BackRub.** It is the most popular search engine globally.
Bing	It was launched in 2009 by **Microsoft.** It is the latest web-based search engine that

	also delivers Yahoo's results.
Ask	It was launched in 1996 and was originally known as **Ask Jeeves.** It includes support for match, dictionary, and conversation question.
AltaVista	It was launched by **Digital Equipment Corporation** in 1995. Since 2003, it is powered by Yahoo technology.
AOL.Search	It is powered by Google.
LYCOS	It is top 5 internet portal and 13th largest online property according to Media Matrix.
Alexa	It is subsidiary of Amazon and used for providing website traffic information.

2. Explain crawling and types of crawling?(nov/dec 2016)

Crawler

- Identifies and acquires documents for search engine
- Many types – web, enterprise, desktop
- Web crawlers follow *links* to find documents
  - Must efficiently find huge numbers of web pages (*coverage*) and keep them up-to-date (*freshness*)
  - Single site crawlers for *site search*
  - *Topical* or *focused* crawlers for verticalsearch
- *Document* crawlers for enterprise and desktop search
  - Follow links and scan directories

Web crawlers

- Starts with a set of *seeds,* which are a set of URLs given to it asparameters
- Seeds are added to a URL request queue
- Crawler starts fetching pages from the request queue
- Downloaded pages are parsed to find link tags that might contain other useful URLs to fetch
- New URLs added to the crawler's request queue, or *frontier*
  Continue until no more new URLs or disk full
  Explain each types in details.

3.Explain XML retrieval?(nov/dec 2016)

Document-oriented XML retrieval

- Document vs. data- centric XML retrieval (recall)
- Focused retrieval
- Structured documents

- Structured document (text) retrieval
- XML query languages
- XML element retrieval
- (A bit about) user aspects

Explain the above in details.

4.Describe index compression techniques?

Uncompressed indexes are large

- ❑ It might be useful for some modern devices to support information retrieval techniques that would not be able to do with uncompressed indexes

Types of Compression

- ■ Lossy
  - ❑ Compression that involves the removal of data.
- ■ Loseless

Compression that involves no removal of data.

- ■ A lossy compression scheme
  - ❑ Static Index Pruning
- ■ Loseless compression
  - ❑ Elias Codes
  - ❑ n-s encoding
  - ❑ Golomb encoding
  - ❑ Variable Byte Encoding (vByte)
  - ❑ Fixed Binary Codewords
  - ❑ CPSS-Tree

5.Write a detail note on how to measure size of web?

How fast does it index

- Number of documents/hour
- (Average document size)

- How fast does it search
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries
- Uncluttered UI
- Is it free?
- All of the preceding criteria are *measurable*: we can quantify speed/size
  - we can make expressiveness precise
- The key measure: user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness
  Issue: who is the user we are trying to make happy?
  - Depends on the setting
- Web engine:
  - User finds what s/he wants and returns to the engine
    - Can measure rate of return users
  - User completes task – search as a means, not end
  - See Russell http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf
- eCommerce site: user finds what s/he wants and buys
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?

6.Explain all in details.Write a note on search engine optimization/spam?

Motives

- Commercial, political, religious, lobbies
- Promotion funded by advertising budget

- Operators
  - Contractors (Search Engine Optimizers) for lobbies, companies
  - Web masters
  - Hosting services

- Forums
  - E.g., Web master world (www.webmasterworld.com)
    - Search engine specific tricks

  Discussions about academic papers

  More spam techniques
- **Doorway pages**
  - Pages optimized for a single keyword that re-direct to the real target page
- **Link spamming**
  - Mutual admiration societies, hidden links, awards – more on these later
  - *Domain flooding:* numerous domains that point or re-direct to a target page
- **Robots**
  - Fake query stream – rank checking programs
    - "Curve-fit" ranking programs of search engines

  Millions of submissions via Add-Url

UNIT IV

WEB SEARCH – LINK ANALYSIS AND SPECIALIZED SEARCH

PART – A

QUESTIONS AND ANSWERS

1. What is the use of Link analysis?

Link analysis which is used to efficiently identify web communities, based on the structure of the web graph.

2.Define link spam:

Link spam are links between pages that are specifically set up to take advantage of link-based ranking algorithms such as Google's Page Rank (PR).Links added to a web page for the purpose of spam indexing

3.Write any one of the link analysis technique:

Our first technique for link analysis assigns to every node in the web graph a numerical score between 0 and 1, known as its Page Rank . The Page Rank of a node will depend on the link structure of the web graph. Given a query, a web search engine computes a composite score for each web page that combines hundreds of features such as cosine similarity and term proximity ,together with the PageRank score. This composite score is used to provide a ranked list of results for the query.

4.How can we assign a page Rank score to each node of the graph?

In assigning a Page Rank score to each node of the web graph, we use the teleport operation in two ways: (1) When at a node with no out-links, the surfer invokes the teleport operation. (2) At any node that has outgoing links, the surfer invokes the teleport operation with probability $0 < \alpha < 1$ and the standard random walk with probability $1-\alpha$, where α is a fixed parameter chosen in advance. Typically, α might be 0.1.

5.How the web pages will be scored based on queries?

For a given a query, every web page is assigned two scores.

One is called its **hub** score and the other its **authority**score . For any query, we compute two ranked lists of results rather than one. The ranking of one list is induced by the hub scores and that of the other by the authority scores.

6.Explain authority with an example:

Authority: The pages that will emerge with high authority scores.

Example: In this approach stems from a particular insight into the creation of web pages, that there are two primary kinds of web pages useful as results for broad-topic searches. By a broad topic search we mean an informational query such as "I wish to learn about leukemia". There are authoritative sources of information on the topic; in this case, the National Cancer Institute's page on leukemia would be such a page. We will call such pages authorities; in the computation we are about to describe, they are the pages that will emerge with high authority scores.

7. Explain hub with an example:-

Hub: These hub pages are the pages that will emerge with high hub scores

On the other hand, there are many pages on the Web that are hand-compiled lists of links to authoritative web pages on a specific topic. These hub pages are not in themselves authoritative sources of topic-specific information, but rather compilations that someone with an interest in the topic has spent time putting together. The approach we will take, then, is to use these hub pages to discover the authority pages. In the computation we now develop, these hub pages are the pages that will emerge with high hub scores

8.What is relevance at information retrieval?

how well a retrieved document or set of documents meets the information need of the user. Relevance may include concerns such as timeliness, authority or novelty of the result.

9. Define ranking for web?

When the user gives a query, the index is consulted to get the documents most relevant to the query. The relevant documents are then ranked according to their degree of relevance, importance etc.

10. Difficulties in Evaluating IR Systems?

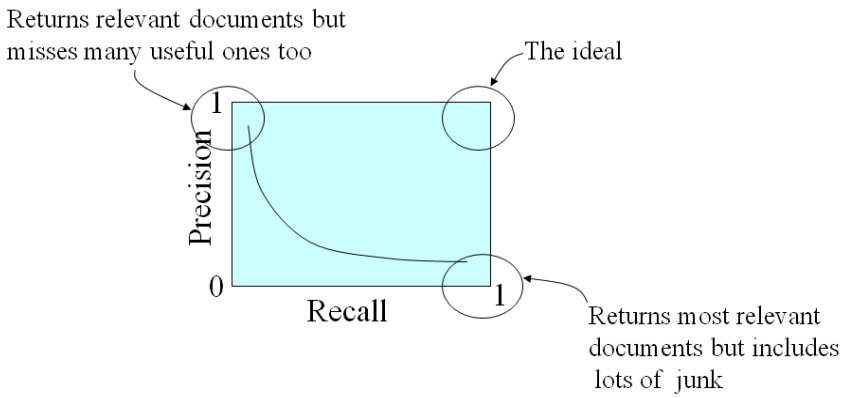
- Effectiveness is related to the **relevancy** of retrieved items.
- Relevancy is not typically binary but continuous.
- Even if relevancy is binary, it can be a difficult judgment to make.
- Relevancy, from a human standpoint, is:
  - Subjective: Depends upon a specific user's judgment.
  - Situational: Relates to user's current needs.
  - Cognitive: Depends on human perception and behavior.
  - Dynamic: Changes over time.

11.Define Precision and Recall?

$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

12.Draw the Trade-off between Recall and Precision?



13.What is MapReduce?

The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples.

14.DefineHadoop?

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

15. What are the factors affecting the performance of CLIR systems.

Limited size of Dictionary, Query translation/transliteration performance

16. What are the Challenges in CLIR.(Cross lingual Information Retrieval)

Translation ambiguity, Phrase identification and translation, Translate/transliterate a term, Transliteration errors, Dictionary coverage, Font, Morphological analysis, Out-of-Vocabulary (OOV) problem

17.Define Snippets.

Snippets are short fragments of text extracted from the document content or its metadata. They may be static or query based. In static snippet it always the first 50 words of the document pe the content

of its description. In query based snippet is one selectively extracted on the basis of its relation to the searcher's query.

18.List the advantages of invisible web content.

- Specialized content focus large amounts of information focused on an exact subject.
- Contains information that might not be available on the visible web.
- Allows a user to find a precise answer to a specific question.
- Allows a user to find WebPages from a specific date or time.

20. What is collaborative filtering?(nov/dec 2016)

It is a method of making automatic predictions about the interests of a single user by collecting preferences or taste information from many users.

21.What do you mean by item based collaborative filtering?

Item based CF is a model based approach which produces recommendations based on the relationship between items inferred from the rating matrix. The assumption behind this approach is that users will prefer items that are similar to other items they like.

22.What are problem of user based CF?

The two main problems of user based CF are that the whole user databases has to be kept in memory and that expensive similarity computation between the active user and all other users in the database has to be performed.

23.Define user based collaborative filtering.

User based collaborative filtering algorithms work off the premise that if a user A has a similar profile to another user B, then A is more likely to prefer things that it prefers when compared with a user chosen at random.

PART – B
QUESTIONS AND ANSWERS

1. Explain about link analysis:
   - Meta-search Engines.
   - HTML structures & Feature Weighting.

   Two methods of Link analysis:
   - Page Rank
   - HITS-hyperlink Induced Topic Search

   Three levels of link analysis:
   - Microscopic level
   - Mesoscopic level

- Macroscopic level

Limitations of Link analysis:

- Meta tags/invisible text
- Pay-for-place
- Stability
- Topic Drift
- Convent evolution

2. Explain about HITS algorithms(nov/dec 2016)

- Hypertext induced Topic selection is a link analysis method developed by John Kleinberg in 1999 using Hub and Authority scores.

Two sets of inter-related pages:

- Hub Pages-good lists of links on a subject
- Authority pages-occur recurrently on good hubs for the subjects.

The HITS algorithm

$H(x) <- \sum a(y)$

$A(x) <- \sum h(y)$

3. Explain CLIR.(nov/dec 2016)

- CLIR-Cross Lingual Retrieval
- Dictionary-based Query Translation
- Document Translation approach
- Interlingua based Approach
- Pseudo-Relevance Feedback (PRF)for CLIR
- Challenges in CLIR

UNIT V

DOCUMENT TEXT MINING

PART – A

QUESTIONS AND ANSWERS

1. Define Information filtering.(nov/dec 2016)

Information filtering delivers to users only the information that is relevant to them, filtering out all irrelevant new data items

2. Differentiate information filtering and information retrieval

Information retrieval is about fulfilling immediate queries from a library of information available. Example : you have a deal store containing 100 deals and a query comes from a user. You show the deals that are relevant to that query.

Information Filtering is about processing a stream of information to match your static set of likes, tastes and preferences.Example: a clipper service which reads all the news articles published today and serves you content that is relevant to you based on your likes and interests.

3. State some applications of Information retrieval

- Automatic delivery of news/alerts
- Online display advertising
- Publish/subscribe systems

4. What is Relevance Feedback?

Feedback given by the user about the relevance of the documents in the initial set of results.

5. Define text mining

The discovery by computer of new, previously unknown information, by automatically extracting information from a usually large amount of different unstructured textual resources.

6. Differentiate Text Mining vs. Data Mining ,web mining, information retrieval

In Text Mining, patterns are extracted from natural language text rather than databases

Text Mining vs • Web Mining – In Text Mining, the input is free unstructured text, whilst web sources are structured.

Text Mining vs • Information Retrieval (Information Access) – No genuinely new information is found. – The desired information merely coexists with other valid pieces of information.

7. Name any two Document Clustering methods.

- K-Means clustering.
- Agglomerative hierarchical clustering.

8. What is Text Preprocessing?

Text pre-processing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis and tagging components, such as morphological analyzers and part-of-speech taggers, through applications, such as information retrieval and machine translation systems.

9. Define classification

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

10. Define clustering

Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. Help users understand the natural grouping or structure in a data set. Used either as a stand-alone tool to get insight into data distribution or as a preprocessing step for other algorithms.

11. Define naivesbaysesclassifiers(nov/dec 2016)

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

12. What is decision tree?

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

13. Define Agglomerative hierarchical clustering

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single

cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

14. Define expectation–maximization (EM).(nov/dec 2016)

Expectation–maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

15. What is supervised learning?

In supervised learning both input and output are provided. The network then processes the inputs and compares its resulting output against the desired outputs. Errors are then propagated back through the systems causing the system to adjust the weights which control the network.

16. What is unsupervised learning?

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis.

17. What is dendrogram?

A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering. Dendrograms are often used in computational biology to illustrate the clustering of genes or samples, sometimes on top of heatmaps.

PART – B

QUESTIONS AND ANSWERS

1. Explain naive Bayes classifiers with an example.(nov/dec 2016)
   Ans.

In general all of Machine Learning Algorithms need to be trained for supervised learning tasks like classification, prediction etc. or for unsupervised learning tasks like clustering.

By training it means to train them on particular inputs so that later on we may test them for unknown inputs (which they have never seen before) for which they may classify or predict etc (in case of supervised learning) based on their learning. This is what most of the Machine Learning techniques like Neural Networks, SVM, Bayesian etc. are based upon.

So in a general Machine Learning project basically you have to divide your input set to a

Development Set (Training Set + Dev-Test Set) & a Test Set (or Evaluation set). Remember your basic objective would be that your system learns and classifies new inputs which they have never seen before in either Dev set or test set.

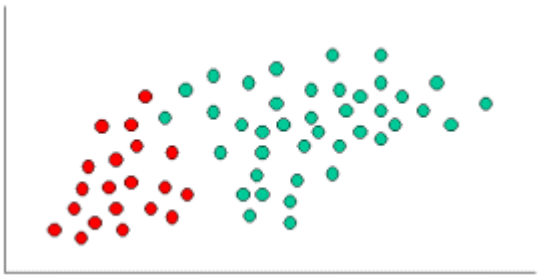
The test set typically has the same format as the training set. However, it is very important that the test set be distinct from the training corpus: if we simply reused the training set as the test set, then a model that simply memorized its input, without learning how to generalize to new examples, would receive misleadingly high scores.

In general, for an example, 70% can be training set cases. Also remember to partition the original set into the training and test sets *randomly*.

To demonstrate the concept of Naïve Bayes Classification, consider the example given below:

Naive Bayes Classifier Introductory Overview

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.



To demonstrate the concept of Naïve Bayes Classification, consider the example displayed in the illustration above. As indicated, the objects can be classified as either GREEN or RED. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently exiting objects.

Since there are twice as many GREEN objects as RED, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership GREEN rather than RED. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of GREEN and RED objects, and often used to predict outcomes before they actually happen.

Thus, we can write:

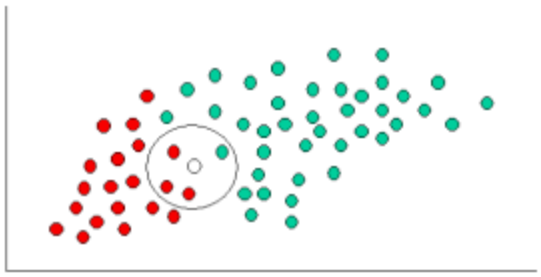
$$Prior\ probability\ for\ GREEN \propto \frac{Number\ of\ GREEN\ objects}{Total\ number\ of\ objects}$$

$$Prior\ probability\ for\ RED \propto \frac{Number\ of\ RED\ objects}{Total\ number\ of\ objects}$$

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are:

$$Prior\ probability\ for\ GREEN \propto \frac{40}{60}$$

$$Prior\ probability\ for\ RED \propto \frac{20}{60}$$



Having formulated our prior probability, we are now ready to classify a new object (WHITE circle). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label. From this we calculate the likelihood:

$$Likelihood\ of\ X\ given\ GREEN \propto \frac{Number\ of\ GREEN\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ GREEN\ cases}$$

$$Likelihood\ of\ X\ given\ RED \propto \frac{Number\ of\ RED\ in\ the\ vicinity\ of\ X}{Total\ number\ of\ RED\ cases}$$

From the illustration above, it is clear that Likelihood of X given GREEN is smaller than Likelihood of X given RED, since the circle encompasses 1 GREEN object and 3 RED ones. Thus:

$$Probability\ of\ X\ given\ GREEN \propto \frac{1}{40}$$

$$Probability\ of\ X\ given\ RED \propto \frac{3}{20}$$

Although the prior probabilities indicate that X may belong to GREEN (given that there are twice

as many GREEN compared to RED) the likelihood indicates otherwise; that the class membership of X is RED (given that there are more RED objects in the vicinity of X than GREEN). In the Bayesian analysis, the final classification is produced by combining both sources of information, i.e., the prior and the likelihood, to form a posterior probability using the so-called Bayes' rule (named after Rev. Thomas Bayes 1702-1761).

$$\textit{Posterior probability of } X \textit{ being GREEN} \propto$$
$$\textit{Prior probability of GREEN} \times \textit{Likelihood of } X \textit{ given GREEN}$$
$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$
$$\textit{Posterior probability of } X \textit{ being RED} \propto$$
$$\textit{Prior probability of RED} \times \textit{Likelihood of } X \textit{ given RED}$$
$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Finally, we classify X as RED since its class membership achieves the largest posterior probability.

Note. The above probabilities are not normalized. However, this does not affect the classification outcome since their normalizing constants are the same.

As indicated, the objects can be classified as either `GREEN` or `RED`. Our task is to classify new cases as they arrive, i.e., decide to which class label they belong, based on the currently existing objects.

Since there are twice as many `GREEN` objects as `RED`, it is reasonable to believe that a new case (which hasn't been observed yet) is twice as likely to have membership `GREEN` rather than `RED`. In the Bayesian analysis, this belief is known as the prior probability. Prior probabilities are based on previous experience, in this case the percentage of `GREEN` and `RED` objects, and often used to predict outcomes before they actually happen.

Thus, we can write:

Prior Probability of `GREEN`: `number of GREEN objects / total number of objects`

Prior Probability of `RED`: `number of RED objects / total number of objects`

Since there is a total of `60` objects, `40` of which are `GREEN` and 20 `RED`, our prior probabilities for class membership are:

Prior Probability for `GREEN`: `40 / 60`

Prior Probability for `RED`: `20 / 60`

Having formulated our prior probability, we are now ready to classify a new object (`WHITE` circle in

the diagram below). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more likely that the new cases belong to that particular color. To measure this likelihood, we draw a circle around X which encompasses a number (to be chosen a priori) of points irrespective of their class labels. Then we calculate the number of points in the circle belonging to each class label

2. Explain decision tree algorithm with example.

Ans.Very simply, ID3 builds a **decision tree** from a fixed set of **examples**. ... The leaf nodes of the **decision tree** contain the class name whereas a non-leaf node is a **decision** node. The **decision** node is an attribute test with each branch (to another **decision tree**) being a possible value of the attribute.

DecisionTreeAlgorithmID3

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1) create a node N;
(2) if tuples in D are all of the same class, C then
(3)     return N as a leaf node labeled with the class C;
(4) if attribute_list is empty then
(5)     return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7) label node N with splitting_criterion;
(8) if splitting_attribute is discrete-valued and
        multiway splits allowed then // not restricted to binary trees
(9)     attribute_list ← attribute_list – splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)    let Dj be the set of data tuples in D satisfying the outcome j; // a partition
(12)    if Dj is empty then
(13)        attach a leaf labeled with the majority class in D to node N;
(14)    else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```

3. Explain Agglomerative clustering with example.

Ans.In the agglomerative hierarchical approach, we start by defining each data point to be a cluster and combine existing clusters at each step. Here are four different methods for doing this:

1. Single Linkage: In *single linkage*, we define the distance between two clusters to be the minimum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest single linkage distance.

2. Complete Linkage: In *complete linkage*, we define the distance between two clusters to be the maximum distance between any single data point in the first cluster and any single data point in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest complete linkage distance.

3. Average Linkage: In *average linkage*, we define the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster. On the basis of this definition of distance between clusters, at each stage of the process we combine the two clusters that have the smallest average linkage distance.

4. Centroid Method: In *centroid method*, the distance between two clusters is the distance between the two mean vectors of the clusters. At each stage of the process we combine the two clusters that have the smallest centroid distance.

5. Ward's Method: This method does not directly define a measure of distance between two points or clusters. It is an ANOVA based approach. At each stage, those two clusters marge, which provides the smallest increase in the combined error sum of squares from one-way univariate ANOVAs that can be done for each variable with groups defined by the clusters at that stage of the process

4. Explain K-means algorithm with example.

Ans.Clustering is the process of partitioning a group of data points into a small number of clusters. For instance, the items in a supermarket are clustered in categories (butter, cheese and milk are grouped in dairy products). Of course this is a qualitative kind of partitioning. A quantitative approach would be to measure certain features of the products, say percentage of milk and others, and products with high percentage of milk would be grouped together. In general, we have n data points $\mathbf{x}i,i=1...n$ that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to find the positions $\mu i,i=1...k$ of the clusters that minimize the *distance* from the data points to the cluster. K-means clustering solves

$$\text{argmin}\mathbf{c}\sum i=1k\sum\mathbf{x}\in cid(\mathbf{x},\mu i)=\text{argmin}\mathbf{c}\sum i=1k\sum\mathbf{x}\in ci\|\mathbf{x}-\mu i\|22$$

where$\mathbf{c}i$ is the set of points that belong to cluster i. The K-means clustering uses the square of the Euclidean distance $d(\mathbf{x},\mu i)=\|\mathbf{x}-\mu i\|22$. This problem is not trivial (in fact it is NP-hard), so the K-means algorithm only hopes to find the global minimum, possibly getting stuck in a different solution.

5. What is the expectation maximization algorithm? Give its applications.(nov/dec 2016)

   Ans.The **expectation maximization algorithm** is a natural generalization of maximum likelihood estimation to the incomplete data case. In particular, **expectation maximization** attempts to find the parameters that maximize the log probability logP(x;) of the observed data.

   In statistics, an **expectation–maximization** (**EM**) **algorithm** is an iterative method to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the *E* step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

Given the statistical model which generates a set of observed data, a set of unobserved latent data or missing values , and a vector of unknown parameters , along with a likelihood function , the maximum likelihood estimate (MLE) of the unknown parameters is determined by the marginal likelihood of the observed dataHowever, this quantity is often intractable (e.g. if is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult).

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying these two steps:

Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the conditional distribution of given under the current estimate of the parameters :

Maximization step (M step): Find the parameter that maximizes this quantity:

The typical models to which EM is applied uses as a latent variable indicating membership in one of a set of groups:

1. The observed data points may be discrete (taking values in a finite or countably infinite set) or continuous (taking values in an uncountably infinite set). Associated with each data point may be a vector of observations.

2. The missing values (aka latent variables) are discrete, drawn from a fixed number of values, and with one latent variable per observed unit.
3. The parameters are continuous, and are of two kinds: Parameters that are associated with all data points, and those associated with a specific value of a latent variable (i.e., associated with all data points which corresponding latent variable has that value).

However, it is possible to apply EM to other sorts of models.The motive is as follows. If the value of the parameteris known, usually the value of the latent variables can be found by maximizing the

log-likelihood over all possible values of , either simply by iterating over or through an algorithm such as the Viterbi algorithm for hidden Markov models. Conversely, if we know the

value of the latent variables , we can find an estimate of the parameters fairly easily, typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, or some function of the values, of the points in each group. This suggests an iterative algorithm, in the case where bothandare unknown:

1. First, initialize the parameters to some random values.
2. Compute the probability of each possible value of given
3. Then, use the just-computed values of to compute a better estimate for the parameters
4. Iterate steps 2 and 3 until convergence.

The algorithm as just described monotonically approaches a local minimum of the cost function.

22 DEC 2016

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 80277

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2016

Seventh Semester

Computer Science and Engineering

CS 6007 – INFORMATION RETRIEVAL

(Regulations 2013)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is Information Retrieval?
2. Specify the role of an IR system.
3. List the retrieval models.
4. Define Document Preprocessing
5. What is the purpose of web crawler?
6. What are the requirements of XML information retrieval systems?
7. Define authorities.
8. Define user based collaborative Filtering.
9. What are the characteristics of information filtering?
10. What are the desirable properties of a clustering algorithm?

PART B — (5 × 16 = 80 marks)

11. (a) Explain in detail about the components of IR.

Or

(b) Write a short note on

(i) Characterizing the web for search. (8)

(ii) Role of AI in IR. (8)

12. (a) Briefly explain weighting and cosine similarity.

Or

(b) Write about relevance feedback and query expansion.

13. (a) Elaborate on the web search architectures.

Or

(b) Describe meta and focused crawling.

14. (a) Compare HITS with Page rank in detail.

Or

(b) Explain in detail cross lingual information retrieval and its limitations in web search.

15. (a) Discuss in detail about the working of Naïve Bayesian classifier with an example.

Or

(b) Give an account of the Expectation Maximization problem.
