UUU

Unit I

CHAPTER 1

Introduction

Syllabus

Introduction to Information Retrieval : Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.

Syllabus Topic : Introduction

1.1 Introduction

Q. 1.1.1 Write a short note on Information Retrieval. (Ref. Sec. 1.1) **(5 Marks)**

- Information Retrieval (IR) is the activity of obtaining material which can usually be documents of an unstructured nature i.e. usually text which satisfies an information need from within large collections which is stored on computers. For example, Information retrieval can be getting a credit card out of your wallet so that you can type in the card number.

- The activities of information retrieval are not engaged for only few people like reference librarians, professional searchers etc. but nowadays hundreds of millions of people engage in information retrieval every day when they use a web search engine or search their email. Information retrieval is dominant form of information access.

- Information retrieval also extends support to users in browsing or filtering document collections or processing a set of retrieved documents. Information retrieval useful to

distinguish three prominent scales which are, **in web search**, the system searches over billions of documents stored on millions of computers.

- Email programs provide a spam filter, manual or automatic means for classifying mail so that it can be placed directly into particular folders.
- In enterprise, institutional and domain-specific search retrieval is provided for collections such as a corporation's internal documents, a database of patents, or research articles on biochemistry.
- Information Retrieval has the ability to represent, store, organize and access information items. A set of keywords are required to search which summarizes the description of the user information that is needed.

1.1.1 Information versus Data Retrieval

- Data Retrieval contains which documents of the collection that consist the keywords in the user query which does not satisfy the user's information need.
- The user of Information Retrieval system worries more about retrieving information about subject rather than with retrieving data that satisfies a given query.
- For a data retrieval system a single error object among thousands retrieved objects can be total failure, but for information retrieval small errors are unnoticed because the text is not structured as compared to data retrievals structured text.
- The area of Information Retrieval has developed much in modelling, document classification and categorization, system architecture, user interfaces, data visualization, filtering, languages etc.

1.1.2 Basic Concepts

Q. 1.1.2 Explain the Interaction of user with retrieval system using diagram. (Ref. Sec. 1.1.2) **(5 Marks)**

1.1.2.1 The User Task

- The information needs to be translated into a query by the user. In information retrieval system there are the set of words that convey semantics of the information need whereas in data retrieval system a query expression is used to convey the constraints which are satisfied by objects.
- In some situations, for example, a user wants to search something but ends up searching with another thing means the user is browsing the documents not searching it.

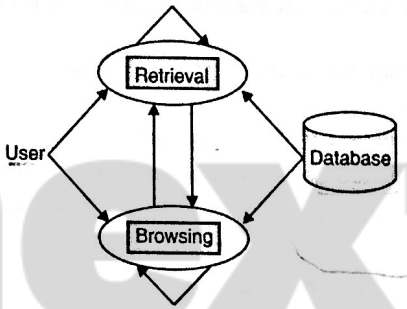

Fig. 1.1.1 : Interaction of user with retrieval system

- The Fig. 1.1.1 shows the interaction of the user through the different task.

1.1.2.2 Logical View of the Documents

- Documents earlier were represented through a set of index terms or keywords whereas modern computers represent documents by full set of words which reduces the set of representative keywords.
- This can be done by eliminating stopwords i.e. articles and connectives, the use of stemming which reduces different words, and identification of noun groups that eliminates verbs etc. Then compression can be done, this operations are text operations.
- These text operations reduce complexity of the document representation from full text to set of index terms.

1.1.3 Past, Present and Future

Q. 1.1.3 Write a short note on Past, Present and Future of Information Retrieval. (Ref. Sec. 1.1.3) **(5 Marks)**

1.1.3.1 Early Developments

- There has been a greater development in organising information for retrieval and usage. For example, the table of contents of a book. As there was an increase in the number of information, it became necessary to build data structures to get faster access.
- Index is the data structure for faster information retrieval which is a collection of selected words and associated pointers.
- Over centuries manual categorization of hierarchies was done for indexes. Nowadays, modern computers have made possible the construction of large indexes automatically.

1.1.3.2 Information Retrieval in the Library

- Libraries were the first to adopt Information Retrieval systems for retrieving information.
- In first generation, it consisted automation of previous technologies and search was on author name and title.
- In second generation, it included searching by subject heading, keywords etc.
- In third generation, it includes graphical interfaces, electronic forms, hypertext features etc.

1.1.3.3 The Web and Digital Libraries

Three changes have taken place,

(i) Cheaper to various sources of information.

(ii) Due to digital communication it provides greater access to networks.

(iii) Free access to publish on large medium.

Syllabus Topic : History of IR

1.2 History of Information Retrieval

Q. 1.2.1 Explain the History of Information Retrieval in detail. (Ref. Sec. 1.2) **(5 Marks)**

- The idea to use computers for searching information was published in the article As We May Think by Vannevar Bush in 1945.
- The first description regarding computer searching for information was described by Holmstrom in 1948. In 1950, automated information retrieval systems were introduced.
- The first large information retrieval research group was formed by Gerard Salton at Cornell in 1960. Lockheed Dialog system which is the Large-scale retrieval systems came in early 1970's.
- The US Department of Defence along with the National Institute of Standards and Technology (NIST), cosponsored the Text Retrieval Conference (TREC) in 1992. Its goal was to look into the information retrieval community by supplying the infrastructure that was needed for evaluation of text retrieval methodologies.
- Earlier to the large use of public day-to-day search engines, Information Retrieval was found in commercial and intelligence applications in 1960.
- As with many computer technologies the retrieval systems grew with increases in processor speed and storage capacity. Information retrieval system finds information that is similar to a user's query.

Syllabus Topic : Components of IR

1.3 Components of Information Retrieval

Q. 1.3.1 Explain the components of Information Retrieval in detail using diagram. (Ref. Sec. 1.3) **(5 Marks)**

- Three major components of Information Retrieval are Document subsystem, User subsystem and Searching /Retrieval subsystem.

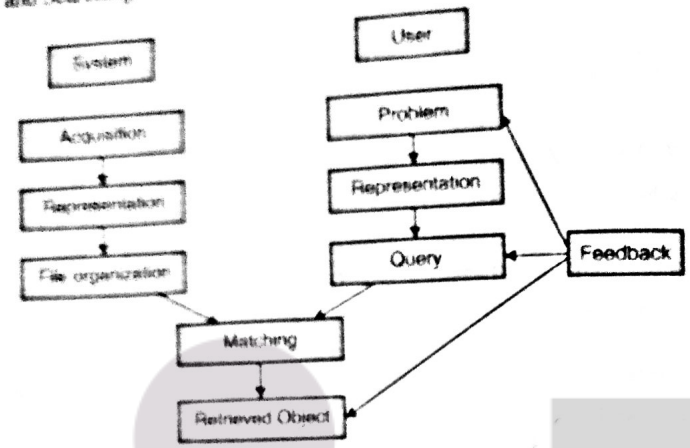

Fig. 1.3.1 : Traditional information retrieval system

1.3.1 Document Subsystem

➜ **1. Acquisition**

It selects documents and other objects from various web resources which consist of mostly text based documents. The data is collected by web crawlers and stored in database.

➜ **2. Representation**

It consist of many ways like,

- Indexing contains free text terms, controlled vocabulary, manual & automatic techniques.
- **Abstracting** : It contains summarizing.
- **Bibliographic description** : It contains author, title, sources, date, and metadata.
- Classifying, clustering.
- Organizing in fields and limits.

➜ **3. File organization**

- **Sequential :** It contains document by document data.
- **Inverted :** It contains term by term, list of records under each term.
- **Combination.**

Syllabus Topic : Issues Related to IR

1.4 Issues in Information Retrieval

Q. 1.4.1 Explain the issues of Information Retrieval in detail. (Ref. Sec. 1.4) **(5 Marks)**

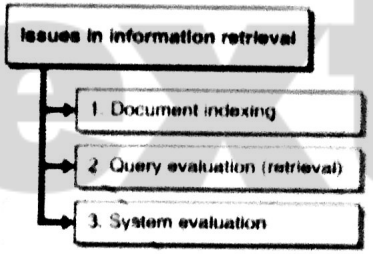

Fig. 1.4.1 : Issues in information retrieval

- Main problems of information retrieval are document and query indexing, query evaluation and system evaluation.

➜ **1. Document indexing**

The main goal is to find the important meanings and create an internal representation. The Factors to be considered are Accuracy to represent meanings (semantics), Exhaustiveness and Facility for computer to manipulate.

➜ **2. Query evaluation (retrieval)**

Here in retrieval model how can a document be represented with the selected keywords and how are document and query representations compared to calculate a score.

➜ **3. System evaluation**

Here, we see if the system is efficient in time and space.

Syllabus Topic : Boolean Retrieval

1.5 Boolean Retrieval

Q. 1.5.1 Explain the basic terms in Boolean Retrieval. (Ref. Sec. 1.5) **(5 Marks)**

– In the Boolean retrieval model, we give any query in the form of a Boolean expression of terms which are combined with the operators AND, OR and NOT.

Some of the basic terms are as follows :

1. Documents	2. Collection
3. Term	4. Information need
5. Query	6. Inverted Index
7. Posting	

➜ **1. Documents**

It is the unit of information that we want to return as a result of a query. For example, Newspaper article, Wikipedia page.

➜ **2. Collection**

It is the group of documents that retrieval is performed on. For e.g. Newspaper archive, Wikipedia.

➜ **3. Term**

It is the smallest unit of information in a query. For e.g. Token.

➜ **4. Information need**

It is the topic about which the user desires to know more, and is differentiated from a query.

➜ **5. Query**

Here user conveys to the computer in an attempt to communicate the information need.

➜ **6. Inverted Index**

Also called as inverted file. It is an index which always maps index back from terms to the parts of a document where they occur.

➜ **7. Posting**

Each item in the list, which records that a term appeared in a document.

1.5.1 An Example Information Retrieval Problem

Q. 1.5.2 Explain with an example the concept of information Retrieval problem in detail. (Ref. Sec. 1.5.1) **(5 Marks)**

– Suppose for example, in Shakespeare's Collected Works book we wanted to find which plays of Shakespeare contain the words Brutus and Caesar and not Calpurnia. In one way we can start reading from beginning through the text and noting for each play whether it contains Brutus and Caesar and excluding it from consideration if it contains Calpurnia.

– The simple way is using computer which does a linear scan of documents which is referred as grepping through text. Grepping allows useful possibilities for wildcard pattern matching through the use of regular expressions.

– To avoid linear scan we can index the documents. To answer the query Brutus and Caesar and not Calpurnia, we take the vectors for Brutus, Caesar and Calpurnia, complement the last, and then do a bitwise and :

110100 and 110111 and 101111 = 100100

The basic idea of an inverted index is shown in Fig. 1.5.1.

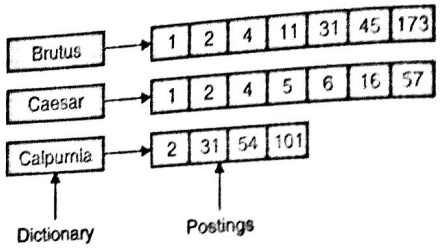

Fig. 1.5.1 : Two parts of inverted index

1.5.2 A First take at Building an Inverted Index

Q. 1.5.3 Write a short note on Inverted Index and explain with example. (Ref. Sec. 1.5.2) **(5 Marks)**

The major steps for building a inverted index are :

1. Collect the documents to be indexed :

   Friends, Romans, countrymen. | So let it be with Caesar

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

   Friends | Romans | Countrymen. | So .....

3. Linguistic preprocessing is done, producing a list of normalized tokens, which are the indexing terms :

   friend | roman | countrymen

4. Index the documents that each term occurs in by creating an inverted index, consisting of a dictionary and postings.

- Each document has a unique docID serial number, called as the document identifier (docID) in document collection.

- When we find every new document we assign successive integers. The input to indexing is a list of normalized tokens for each document.

- The core indexing step is sorting by which the terms are alphabetical. Multiple occurrences of the same term are merged. Instances of the same term are grouped and the result is split into a dictionary and postings.

1.5.3 Processing Boolean Queries

To process a query using an inverted index and the basic Boolean retrieval model, consider simple conjunctive query :

☛ **Brutus and Calpurnia**

Use Fig. 1.5.2

1. Locate Brutus in the dictionary.
2. Retrieve its postings.
3. Locate Calpurnia in the dictionary.
4. Retrieve its postings.
5. Intersect the two postings lists, as shown in Fig. 1.5.2

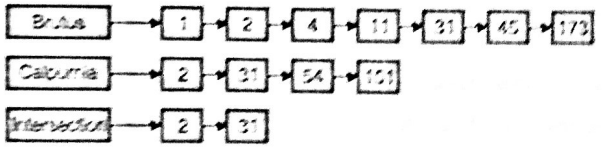

Fig. 1.5.2 : Intersecting the postings lists for Brutus and Calpurnia from Figure

The intersection operation is an important where we efficiently intersect postings list so we can find documents quickly that contains both terms.

Using merge algorithm we can show intersection of two postings by,

```
INTERSECT (p1, p2)
1.  answer ← < >
2.  while p1 ≠ nil and p2 _= nil
3.  do if doc ID(p1) = doc ID(p2)
4.       then ADD(answer, doc ID(p1))
5.            p1 ← next(p1)
6.            p2 ← next(p2)
7.       else if doc ID(p1) < doc ID(p2)
8.            then p1 ← next(p1)
9.            else p2 ← next(p2)
10. return answer
```

1.5.4 The Extended Boolean Model versus Ranked Retrieval

- The Boolean retrieval model is opposite of ranked retrieval models like vector space model where users use free text queries where we type one or more words rather than using a precise language operators to building up query expressions.
- A proximity operator means to specify two terms in a operator query that occur close to each other in a document, where closeness may be measured by limiting the allowed number of intervening words or by reference to a structural unit such as a sentence or paragraph.

Syllabus Topic : Dictionaries and Tolerant Retrieval

1.6 Dictionaries and Tolerant Retrieval

- Earlier, we have seen the ideas for inverted indexes for handling Boolean and proximity queries.
- In this section, we will see techniques that are robust to typographical errors in the query, as well as alternative spellings.

1.6.1 Search Structures for Dictionaries

- First task is to find whether every query term exists in the vocabulary and, if so, identify the pointer to the corresponding postings, when we have given an inverted index and a query.
- Dictionary which is a classical data structure has two broad classes of solutions which are hashing and search trees. The entries in vocabulary of data structure are called as keys.
- In some search engines, hashing is used for dictionary lookup. Each vocabulary term which is key is hashed into an integer over a large enough space.
- At query time, we hash each query term separately, and further pointer to the corresponding postings, taking into account any logic for resolving hash collisions. Certainly no easy way to find minor variants of a query term is there.
- Search trees masters many of the above issues, for example, permit us to enumerate all vocabulary terms beginning with automat. The well known search tree is the binary tree where each internal node has two children where the search starts from root.
- Every internal node constitutes a binary test. The main problem here is that of rebalancing; as terms are inserted into or deleted from the binary search tree.

1.6.2 Wildcard Queries

Q. 1.6.1 Write a short note on : Wildcard Queries. (Ref. Sec. 1.6.2) **(5 Marks)**

- In the following situations wildcard queries are used :

1. The user is unsure of the spelling of a query term, for example, Syria vs Siria which leads to wildcard entry S*ria.
2. The user is aware of multiple varieties of spelling a term and sees for the documents containing any of the variants, for example, Color vs colour.
3. The user sees documents containing varieties of a term that would be caught by stemming.
4. The user is unsure of the correct rendition of a foreign word or phrase.

- A query like mon* is called as trailing wildcard query, as the * symbol occur at the end of the search string. We can calculate the set W of terms in the dictionary with the prefix mon, as we walk down the tree.
- A query like *mon is called as leading wildcard query, as the * symbol occur at the start of the search string. Here, we consider a reverse B-tree on the dictionary where each root-to-leaf path of the B-tree corresponds to a term in the dictionary written backwards.

1.6.2.1 General Wildcard Queries

Q. 1.6.2 Write a note on Permuterm indexes. (Ref. Sec. 1.6.2.1) **(5 Marks)**

To handle general wildcard queries there are two techniques, which have a common strategy i.e. express the given wildcard query q_w as a Boolean query Q on a specially constructed index, such that the answer to Q is a superset of the set of vocabulary terms matching q_w. Further we check each term in the answer to Q against q_w, discarding those vocabulary terms that do not match q_w.

Permuterm Indexes

Permuterm index is a index form of inverted index. To mark the end of a term a special symbol $ into our character set is introduced. For example, the term hello is shown as the augmented term hello$. Then we construct a permuterm index, in which the various rotations of each term all link to the original vocabulary term. The set of rotated terms in the permuterm index are said as the permuterm vocabulary.

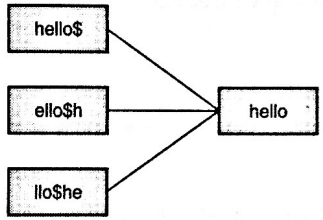

Fig. 1.6.1 : A portion of a permuterm index

1.6.2.2 k-Gram Indexes for Wildcard Queries

Q. 1.6.3 Explain k-Gram indexes for wildcard queries in detail. (Ref. Sec. 1.6.2.2) **(5 Marks)**

- For processing wildcard queries a second technique is called as the k-gram index which is a sequence of k characters. For example, cas, ast and stl are all 3-grams which occur in the term castle. Thus a special character $ is used to denote the beginning or end of a term, so the full set of 3-grams generated for castle is: $ca, cas, ast, stl, tle, le$.
- Here, the dictionary contains all k-grams that occur in any term in the vocabulary. For example, the 3-gram etr would point to vocabulary terms like, Metre, Metro etc.

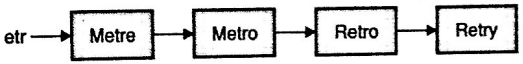

Fig. 1.6.2 : Postings list in a 3-gram index

This creates a difficulty with the use of k-gram indexes which further demands one more step of processing. A post filtering step is introduced to manage this.

1.6.3 Spelling Correction

Q. 1.6.4 Write a short note on Spelling Correction. (Ref. Sec. 1.6.3) **(5 Marks)**

For example, correcting spelling errors here we want to get documents which contain the term carrot when the user types the query carot.

1.6.3.1 Implementing Spelling Correction

Two algorithms for spelling correction are :

1. For a misspelled query, we choose the one which is nearest of different alternatives of correct spelling.
2. Select the one that is more common when two correctly spelled queries are tied. For eg. In grunt and grant, the more common is chosen as correction. The simple idea is that number of occurrences is considered of the term in the collection.

1.6.3.2 Forms of Spelling Correction

Q. 1.6.5 Write the two forms of Spelling Correction and explain any one in detail. (Ref. Sec. 1.6.3.2) **(2 Marks)**

There are two forms of spelling correction :

1. **Isolated-term correction :** Here we correct a single query term at a time, even when we have a multiple-term query. Here two techniques are there; edit distance and k-gram overlap.
2. **Context sensitive correction :** Here the typographical errors i.e. phrases are corrected.

1.6.3.3 Edit Distance

The edit distance between the two character strings s_1 and s_2 is the minimum number of edit operations required to transform s_1 into s_2.

The edit operations are as follows which are sometimes referred as Levenshtein distance,

- Insert a character into a string
- delete a character from a string
- replace a character of a string by another character

To find the edit distance between two strings in time $O(|s_1| \times |s_2|)$, where $|s_i|$ denotes the length of a string s. Here we are using the dynamic programming algorithm which is shown below,

```
Edit Distance(s1, s2)
1   intm[|s1|, |s2|] = 0
2   for i ← 1 to |s1|
3   do m[i, 0] = i
4   for j ← 1 to |s2|
5   do m[0, j] = j
6   for i ← 1 to |s1|
7   do for j ← 1 to |s2|
8   do m[i, j] = min{m[i − 1, j − 1] + if (s1[i] = s2[j]) then 0 else 1fi,
9    m[i − 1, j] + 1,
10   m[i, j − 1] + 1}
11  return m[|s1|, |s2|]
```

Where, m is matrix, (i, j) entry of the matrix.

1.6.3.4 k-Gram Indexes for Spelling Correction

The k-gram index is used to retrieve vocabulary terms that have many k-grams in common with the query. Suppose for example, three 2-gram indexes is Fig. 1.6.3 shows the postings for the three bigrams in the query **bord**.

If we wanted to retrieve vocabulary terms that contained at least two grams then a single scan of the postings can calculate all the terms.

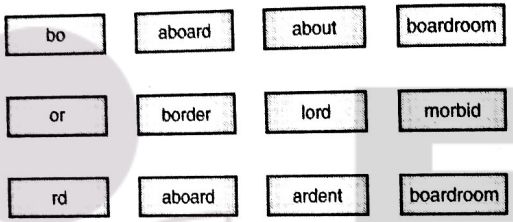

Fig. 1.6.3 : Matching minimum two of the three 2-grams in the query bord

1.6.3.5 Context-Sensitive Spelling Correction

- Typographical errors like flew form Heathrow is not corrected by Isolated-term correction where the terms are correctly spelled.
- The simple way to do is by calculating corrections of each of the three query terms even though each query term is correctly spelled, then try substitutions of each correction in the phrase.
- For each substitute phrase, the search engine runs the query which determines the number of matching results. These calculations are expensive as there are many corrections of the individual terms.

1.6.4 Phonetic Correction

Q. 1.6.6 Write a short note on Phonetic Correction. (Ref. Sec. 1.6.4) **(5 Marks)**

- In the last technique misspellings occur asthe user types a query that sounds like the target term. The idea behind this is to form a phonetic hash so that similar-sounding terms hash to the same value.
- Soundex algorithms are the algorithms for phonetic hashing. Soundex algorithm uses the following scheme :
  1. Turn every term to be indexed into a four-character reduced form. Build an inverted index from these reduced forms to the original terms and then call this the soundex index.
  2. Do the same with query terms.
  3. When the query calls for a soundex match, search this soundex index.
- For different soundex algorithm conversion of terms to four-character forms is there.
  1. Retain the first letter of the term.
  2. Change all occurrences of the following letters to '0' (zero) : A, E, I, O, U, H, W, and Y.
  3. Change letters to digits as follows :

     B, F, P, V to 1.

     C, G, J, K, Q, S, X, Z to 2.

     D, T to 3.

     L to 4.

     M, N to 5.

     R to 6.
  4. Repeatedly remove one out of each pair of consecutive identical digits.
  5. Remove all zeros from the resulting string. Add the resulting string with trailing zeros and return the first four positions, which will consist of a letter followed by three digits.

1.7 Exam Pack (Review Questions)

Syllabus Topic : Introduction

Q. 1 Write a short note on Information Retrieval. (Refer Section 1.1) **(5 Marks)**

Q. 2 Explain the Interaction of user with retrieval system using diagram. (Refer Section 1.1.2) **(5 Marks)**

Q. 3 Write a short note on Past, Present and Future of Information Retrieval. (Refer Section1.1.3) **(5 Marks)**

Syllabus Topic : History of IR

Q. 4 Explain the History of Information Retrieval in detail. (Refer Section 1.2) **(5 Marks)**

Syllabus Topic : Components of IR

Q. 5 Explain the components of Information Retrieval in detail using diagram. (Refer Section 1.3) **(5 Marks)**

Syllabus Topic : Issues Related to IR

Q. 6 Explain the issues of Information Retrieval in detail. (Refer Section 1.4) **(5 Marks)**

Syllabus Topic : Boolean Retrieval

Q. 7 Explain the basic terms in Boolean Retrieval. (Refer Section 1.5) **(5 Marks)**

Q. 8 Explain with an example the concept of information Retrieval problem in detail. (Refer Section 1.5.1) **(5 Marks)**

Q. 9 Write a short note on Inverted Index and explain with example. (Refer Section 1.5.2) **(5 Marks)**

Syllabus Topic : Dictionaries and Tolerant Retrieval

Q. 10 Write a short note on : Wildcard Queries. (Refer Section 1.6.2) **(5 Marks)**

Q. 11 Write a note on Permuterm indexes. (Refer Section 1.6.2.1) **(5 Marks)**

Q. 12 Explain k-Gram indexes for wildcard queries in detail. (Refer Section 1.6.2.2) **(5 Marks)**

Q. 13 Write a short note on Spelling Correction. (Refer Section 1.6.3) **(2 Marks)**

Q. 14 Write the two forms of Spelling Correction and explain any one in detail. (Refer Section 1.6.3.2) **(5 Marks)**

Q. 15 Write a short note on Phonetic Correction. (Refer Section 1.6.4) **(5 Marks)**

Chapter Ends

Unit II

CHAPTER 2

Link Analysis and Specialized Search

Syllabus

Link Analysis and Specialized Search : Link Analysis, hubs and authorities, Page Rank and HITS algorithms, Similarity, Hadoop and 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.

Syllabus Topic : Link Analysis

2.1 Link Analysis

Q. 2.1.1 Define Link Analysis. (Ref. Sec. 2.1) **(2 Marks)**

Q. 2.1.2 Write a note on Link Analysis. (Ref. Sec. 2.1) **(5 Marks)**

- For the development of web search the analysis of hyperlinks and the graph structure of the Web are the basics. Link analysis for web search is the predecessors in the field of citation analysis, aspects of which overlap with an area that is called as bibliometrics.
- It quantifies the influence of scholarly articles by analyzing the pattern of citations. Citations represent the conferral of authority from a scholarly article to others.
- For example, One may contrive to set up multiple web pages pointing to a target web page, with the intent of artificially boosting the latter's tally of in-links which is called as link spam.