Assignment No:- 01

Problem statement:-

Implement single pass algorithm for clustoring of file.

objectives:-

To study:

1. what is clustering?
2. single pass algorithm for clustering.
3. measure of association
4. The graphical representation of clustering.

Theory:

Clustering.

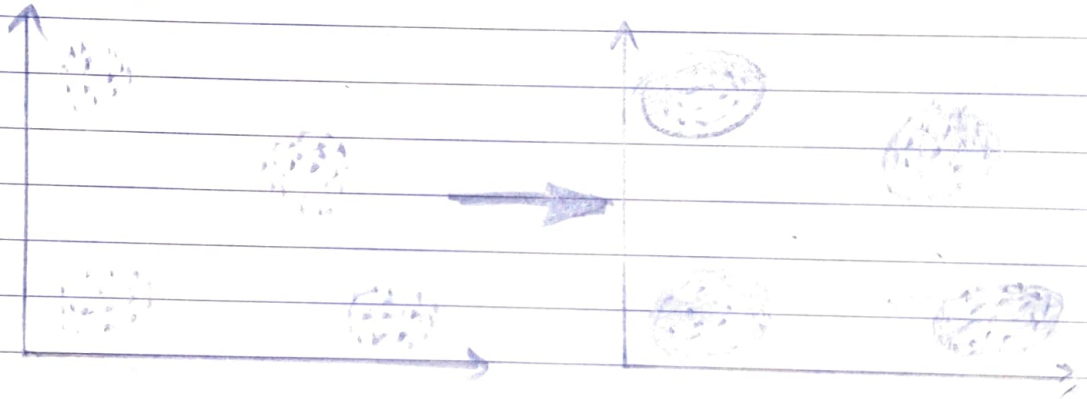
Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data

A definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

Clustering is the process of grouping the documents with which are relevant. It can be shown by a graph with nodes connected if they are relevant to the same request. A basic assumption is

that is documents relevant to a request are separated from those which are not relevant i.e. the relevant documents are more like one another than they are like non relevant one.

A simple graphical examples:



To Identify the 4 Clusters into which the data can be divided: the similarity criterion is distance: two or more objects belong to the Same Cluster if they are "close" according to a given distance (in this case geometrical distance). This is called distance-based clustering.

Another kind of clustering is Conceptual clustering: two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, Objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

The Goals of Clustering

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. For instance, user could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects. (Outlier detection)

Clustering Requirements:

The main requirements that a clustering algorithm should satisfy are:-

1. Scalability;
2. Dealing with different types of attributes;
3. Discovering clusters with arbitary shape;
4. Minimal requirements for domain knowledge to determine input parameters;
5. Ability to deal with noise and outliners;

6. Insensitivity to order of input records;
7. High dimensionality.
8. Interpretability and usability.

Single pass Clustering.

Single pass clustering quickly by which we make incremental clustering to stream data. This clustering technique provides us with a simple yet flexible technique for stream data. Gives a collection of clusters and a threshold value h, if a new document n has the highest similarity more than h to some cluster, the document n is appended to the cluster, and if there exists no cluster, a new cluster is generated which contains only the document n. Clearly single-pass clustering is suitable for incremental clustering to temporal data since, once a document is assigned to a cluster, it is not changed in the future.

The algorithm is as follows:

1) Let h be a threshold value
2) Let s be an empty set and d_1 be the first document. We generate a new cluster C_1 consisting of d_1
3) When a new document $d(i+1)$ come in, calculate the similarity values to all

the Cluster C

4) Let simmax be the highest value and Cdi the most similar Cluster. If simmax > h, add di to Cdi and adjust the center of Cdi. Otherwise, we generate a new Cluster Cdi that contains only di.

5) Repeat the process above until no data comes. In (4) we define simmax = MAX(Sim (~di, ~C)). Also we define similarity of a document d and a Cluster C where the center is VC as below (called Cosine similarity):

$$Sim(\sim d, \sim C) = d_{\sim} \cdot V_{\sim C} / \|\sim d\| \|V_{\sim C}\|$$

Measures of association.

Association is the similarity between objects characterized by discrete state attributes. The measure of similarity or association is designed to quantify likeness between the objects. In such a way that an object in a group is more like the other members of the group that is like object outside the group then a Cluster method Cable such a group structure to the be discovered.

There are five commonly used measures of association In IR.

1. $|X \cap Y|$ Simple matching coefficient
2. $|X \cap Y| / |X| + |Y|$ Dices Coefficient.

3. $|X \cap Y| / |X \cup Y|$ Jaccard's Coefficient.
4. $|X \cap Y| / |X|^{1/2} * |Y|^{1/2}$ Cosine Coefficient
5. $|X \cap Y| / \min(|X|, |Y|)$ Overlap Coefficient.

In shorrt, measure of association is ~~Called~~ Calculated by this program by taking into account frequency of occurrence of words in both the documents. i.e. least value of ~~fr~~equency of occurrence of a common word in both documents is considered for finding out the measure of association.

Classification methods:-
1. multi ~~state~~ attribute (E.g.: Colour)
2. Binary state (E.g.: Keyword)
3. Numerical (E.g.: Hardness Scale or weighted Keyword)
4. probability distribution.

Cluster hypothesis:

The hypothesis can be simply stated as closely associated document tend to be relevant to the same request. This hypothesis is referred as Cluster hypothesis.

The basic assumption in retrival system is that documents relevent to a request are separated from those which have not relevant. Compute the association

between all pairs of documents.

a. Both of which are relevant to a request and
b. one of which is relevant and the other is not.

Two approaches of clustering

1. The clustering is based on a measure of similarities between the objects to be clustered. The example of first approach is graph theoretic method which can be used to define clusters in terms of graphs derived from measure of similarity.

2. The cluster method proceeds directly from the object description.

Graphical representation of clustering.

Here similarity matrix is used in order to draw the graph, the documents having measure of association greater than threshold value can be represented by the edge in the graph. This is an identical cluster which can shown as connected graph.

From the graph below, it can be easily understood that all documents are associated. But documents like 2 & 5 are not directly associated & same is the case for documents 4 & 5. In this way clusters

Can be depicted.

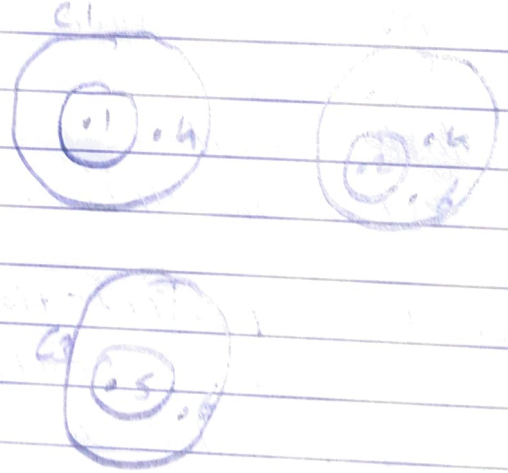
Example:

Objects - {1, 2, 3, 4, 5, 6}

Threshold: 0.5

1						
2	3					
3	5	6				
4	8	9	7			
5	4	3	85	6		
6	2	8	4	5	6	
	1	2	3	4	5	6

Clusters are:



Input:- Document representative (minimum 5 files)

Scope:- Use of matching function for comparing document representative. define appropriate threshold value.

Program Implementation:- Code written in C/C++ to implement Single pass algorithm for clustering with proper output.

Output:- Clusters of documents

Conclusion:-
Thus, we have implemented the single-pass algorithm for clustering.

Assignment No. 02

Problem Statement:

To implement a program for Retrival of documents using Inverted files

Objectives:-

To study Indexing. Inverted files and Searching information with the help of Inverted file.

Outcomes:

At the end of the assignment the students should have:

1. Understood use of indexing in fast retrieval
2. Understood working of inverted index.

Infrastructure:- Desktop/laptop System with Linux or its derivatives.

Software used:-

LINUX/ Windows OS/ Virtual Machine/IDE/ C/ C++/ Java/ Python

Theory:

Indexing

In Searching for a basics query is to Scan the text sequentially. Sequential or online text searching involves finding the occurance of a pattern in a text. online Searching is appropriate then the text is.

Small and it is the only choice if the text collection is very volatile or the index space overhead cannot be afforted. A second option is to build data structures over the text to speed up the search. It is worthwhile building and maintaining an index when the text collection is large and semi-static. Semi-static collections can be updated at reasonably regular intervals but they are not deemed to support thousands of insertions of single words per second. This is the case for most real text databases not only dictionaries or other slow growing literary works. There are many indexing techniques.

Three of them are inverted files, suffix arrays and signature files.

Inverted files:-

An inverted file is a word-oriented mechanism for indexing a text collection in order to speed up the matching task. The inverted file structure is composed of two elements. Vocabulary and occurrence. The vocabulary is the set of all different words in the text. For each word a list of all the text positions where the word appears is stored. The set of all those list

Searching with the help of inverted file:-

The search algorithm on an inverted index has three steps.

1. Vocabulary search.

Searching with the help of inverted file:
The search algorithm on an inverted index has three steps.

1. Vocabulary Search
2. Retrieval of occurrence
3. manipulation of occurrences.

Single-word queries can be searched using any suitable data structure to speed up the search, such as hashing tries, or B-trees. The first two give O(m) search cost. However, simply storing the words in lexicographically order is cheaper in space and very competitive in performance. Since the word can be binary searched at O(log n) cost, prefix and range queries can also be solve with binary search, tries, or B-trees but not with hashing. If the queries is formed by single words then the process end by delivering the list of occurance. Context queries are more deficult to solve with inverted indices. Each element must be searched separately and a list generated for each one.

Then, the lists of all elements are traversed in synchronization to find places where all the words appear in sequence or appear close enough. If one list is much shorter than the other it may be better to binary search its elements into the longer lists instead of performing a linear merge. If block addressing is used it is necessary to traverse the blocks for these queries, since the position information is needed. It is then better to intersect the list of obtain the blocks which contain all the searched words and then sequentially search the context query in those blocks. Some care has to be exercised at block boundries, since they can split a match. example,

Text:
1 6 9 11 17 19 ...
This is a text. A text has many words. Words are made from letters.

Inverted Index:

Vocabulary	occurrences.
Letters	60 ...
made	50 ...
many	28 ...
Text	11, 19 ...
words	33, 40 ...

Algorithm

1. Input the conflated file
2. Build the index file for input file.
3. Input the query
4. Print the index file and result of query.

Conclusion:-

Implementation is concluded by stating analysis of retrieval of documents using Inverted files.

Assignment No: 03

Problem Statement:-
Write a program to Calculate harmonic mean (F-measure) and E-measure for above example

Objectives:-
1. To evaluate the retrieval performance of IR systems.
2. To understand importance of harmonic mean (F-measure) and E-measure in information retrieval.
3. To study indexing structures for information retrieval

Outcomes:-
At the end of the assignment the students should have:
1. Understood to calculate harmonic mean (F-measure) and E-measure in information retrieval.
2. understood method to evaluate the retrival performance of IR systems.

Theory:
F-score / F-Measure:)

F1 score considers both precision and recall. It is the harmonic mean (average) of the precision and recall. F1 score is best if there is some sort of balance between precision (p) & recall (r) in the system. Oppositely f1 score isn't so high if one measure is improved at the expense of the other. For example,

if P is 1 & R is 0, F1 score is 0
F1 score = 2*(Recall * precision) / (Recall + precision)

Information Systems can be measured with two metrics: precision and recall. Thus, precision and recall have been extensively used to evaluate the retrieval performance of IR systems or algorithms. However, a more careful reflection reveals problems with these two measures: First, the proper estimation of maximum recall for a query requires detailed knowledge of all the documents in the collection. Second, in many situations the use of a single measure could be more appropriate. Third, recall and precision measure the effectiveness over a set of queries processed in batch mode. Fourth, for systems which require a weak ordering through, recall and precision might be inadequate.

Alternative measures: The harmonic mean/F measure: The F-measure is also a single measure that combines recall and precision.

$$f(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

where,
r(j) is the recall at the j-th position in the ranking.

P(j) is the precision at the j-th position in the ranking
F(j) is the harmonic mean at the j-th position in the ranking.

Determining max value of F can be interpreted as an attempt to find the best possible compromise between recall and precision. The function F assumes values in the interval [0, 1] It is 0 when no relevant documents have been retrieved and is 1 when all ranked documents are relevant further, the harmonic mean F assumes a high value only when both recall and precision are high to maximize F requires finding the best possible compromise between recall and precision.

Alternative measures: E measure

E measere was proposed by Van Rijesbergen which combines recall and precision. User is E measure is defined as

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

where,
r(j) is the recall at the j-th position in the ranking.
P(j) is the precision at the j-th position in the ranking

b≥0 is a user specified parameter
E(j) is the E metric at the j-th position in the ranking.
If b=1 E(j) measure works as Complement of the Harmonic mean F(j)
If b>1 indicates that the user is more interested in precision than in recall
If b<1 Indicates that user is more interested in recall than in precision
Notice that setting b=1 in the formula of the E-measure yields f(j) = 1-E(j)
If b=0 E(j) = 1-P(j) low values of b make E(j) a function of precision.
If b→∞ lim b→∞ E(j) = 1-r(j) high values of b make E(j) a function of recall
For b=1, the E-measure becomes the F measure.
Thus. Single values measures can also be stored in a table to provide a statistical summary for instance, these summary table statistics could include the number of queries used in the task the total number of Documents retrieved by an queries the total number of relevant docs retrieved by an queries the total number of relevant docs for an queries, as judged by the specialists.
Sample code in C++

- Code
- Output

Conclusion :-

Implementation is concluded by executing a program to calculate (F-measure) and E-measure for sample input used in above example.

Assignment No: 04

Problem Statement:

To Implement a program for feature extraction in 2D Color images (any features like colors, texture etc.) and to extract features from input image and plot histogram for the features.

Objective:-

1. To study program for feature Extraction in 2D Colour images for features like, Colour and Textures and plot Histogram.
2. Given feature Extraction source code is implemented using Java or python language.
3. The input to the program is image file that is to be modified using program by changing Colour.

Outcomes:-

At the end of the assignment the students should have

1. understood the feature extraction process and its applications.
2. Apply appropriate tools in analyzing the web information.

Infrastructure: Desktop / laptop system with Linux or its derivatives.

Software used:- LINUX / Windows OS / virtual machine / IDE, python 3.9.12

Input:- Image file
Output:- Features of Image file.

Theory:
Introduction:-

Technology determines the types and amounts of information we can access. Currently, a large fraction of information originates in silicon. Cheap, fast chips and smart algorithms are helping digital data processing take over all sorts of information processing. Consequently, the volume of digital data surrounding us increases continuously. However, an information centric society has additional requirements besides the availability and capability to process digital data. We should also be able to find the pieces of information relevant to a particular problem. Having the answer to a question but not being able to find it is equivalent to not having it at all. The increased volume of information and the wide variety of data types of data make finding information a challenging task. Current searching methods and algorithms are based on assumptions about technology and goals that seemed reasonable before the widespread use of computers.

However, these assumption no longer hold in the Contex of information retrival systems. The pattern originated in the information retrieval domain. However, information retrieval has expanded into other fields like office automatic genome databases, fingerprint identification medical Imaging data mining. multimedia etc. since the pattern works with any kind of data, It is applicable in many other documaints. You will see examples from text searching, telecommunications, stock prices, medical imaging and treademark. Symbols. The key idea of the pattern is to map from a large, complex problem space into a small, simple feature space The mapping represents the creative part of the solution. Every type of application uses a different kind mapping. mapping into the feature space is also the hard part of this pattern. Traditional searching algorithms are not viable for problems typical to the information retrival domain since they the were designed for exact matching, their use for similarity search is cumbersome, In Contrast, feature Extraction provides an elegant and efficient alternative. with information retrival expanding into other fields, this pattern is applicable in a wide range of applications.

Feature Extraction:-

Texture is an important feature that identifies the object present in any image. The texture is defined by the special distribution of pixels in the neighborhood of an Image. The gray level special dependency is represented by a two-dimensional matrix known as GLCM and it is used for texture analysis. The GLCM matrix specifies that how often the pairs of pixels with certain values occure in an image. the statistical measure are then derived using the GLCM matrix. The textural features represent the spatial distribution of gray tonal variations within a specified area. In Images, the neighboring pixel is correlated and spatial values are obtained by the redundancy between the neighboring pixel values. The color feature are represented by color histogram in six color spaces namely RGB, HSV, LAB, CIE, HUE, and OPP.

The textual features are considered for classifying the image. These texual features are calculated in the spatial domain and a set of gray tone spatial dependency matrix was computed. The texturan features are based on the fact that describes how the gray

tone appears in a spatial relationship to another

GRAY LEVEL CO-OCCURENCE MATRIX (GLCM)

In statistical texture analysis, from the distribution of intensities the texture features are obtained at specified position relative to one another in an image. the statistics of texture are classified into first order, second order and higher order statistics. The method of extracting second order statistical texture feature is done using Gray Level Co-Occurrence matrix (GLCM). First order texture measure is not related to pixel neighbor relationships and it is calculated from the original image. GLCM consider the relation between two pixels at a time, called reference pixel and a neighbor pixel. GLCM is defined by a matrix in which the number of rows and columns are equal to the number of gray levels G. In an image. the matrix elements $P(I,j|\Delta x, \Delta y)$ is the relative frequency where I and j represent the intensity and both are separated by a pixel distance $\Delta x, \Delta y$. The different textural features such as Energy, entropy, Contrast, homogeneity, correlation, dissimilarity, inverse difference moment

and maximum probability can be computed using GLCM matrix.

Significance of Extracted Feature:

1. Color: It signifies the object identification and extraction of from scene.
2. Brightness: Brightness is one of the most significant pixel characteristics. It should be used only for no quantitative references to physiological sensations and perceptions of light.
3. Entropy: It characterized the texture in image
4. Contrast:- Contrast is the dissimilarity or diffeence between things.
5. Shape of image
6. Size of image
7. Owner, file name, file type etc.

Steps:-

A. Importing an Image:-
Importing an image in python is easy. following code will help you import an image on python.

```
image = imread('c:\Users\Travish\Desktop\1.jpg')
show_img(image)
```

B. Understanding the underlying data
The image have several colors and many pixels.

1. To visualize how this image is stored, think of every pixel as a cell in matrix.
2. Now this cell contains three different intensity information, catering to the color Red, Green and Blue, so a RGB image becomes a 3-D matrix
3. Each number is the intensity of Red, Blue and green colours.

```
red, yellow = image.copy(), image.copy()
red[:, :, (1,2)] = 0
yellow[:,:,2] = 0
show_images (images = [red, yellow], titles = ["Red Intensity', 'Yellow Intensity"])
```

C. Converting Images to a 2-D matrix:-

1. Handling the third dimension of images somethings can be complex and redundant.
2. In feature extraction, it becomes much simpler if we compress the image to a 2-D matrix
3. This is done by Gray scaling, Here is how you convert a RGB image to Gray Scale

```
from skimage.color import rgb2gray
gray_image = rgb2gray (image)
show_images (images = [image, gray_image], titles = ["color", "Grayscale"])
print("Colored image shape:", image.shape
print "Grayscale image shape:", gray_image.shape
```

Now let's try to binarize this gray scale image:-

Blurring an Image:-

Last part of this assignment is more relevant for feature for feature extraction: Blurring of images.

```
from skimage.filter import gaussian_filter
blurred_image = gaussian_filter(gray_image, sigma=20)
show_images(images=[gray_image, blurred_image],
    titles=["Gray Image", "20 Sigma Blur"])
```

Example:-

```
image = imread('C:\Users\Tavish\Desktop\7.jpeg')
show_img(image)
red, yellow = image.copy(), image.copy()
red[:,:,2] = 0
yellow[:,:,2] = 0
show_images(images=[red, yellow], titles=['Red
  Intensity', 'Yellow Intensity'])
from skimage.color import rgb2gray
gray_image = rgb2gray(image)
show_image(images=[image, gray_image], title=
    ["Color", "Grayscale"])
print "Colored image shape:", image.shape
print "Grayscale image shape:", gray_image.shape
from skimage.filter
import threshold_otsu
thresh = threshold_otsu(gray_image)
binary = gray_image > thresh
```

show_images([images=[gray_image, binary_image, binary], titles=["Grayscale", "Otsu Binary"]])

from skimage.filter import gaussian_filter

blurred_image = gaussian_filter(gray_image, sigma=20)

show_images([images=[gray_image, blurred_image], titles=["Gray Image", "20 Sigma Blur"]])

Second part of the Assignment - Plotting the Histogram.

Histogram is a graphical representation showing how frequently various colour values occur in the image.

Steps:

Importing image data:-

import matplotlib.pyplot as plt.

The image should be used in a PNG file as matplotlib supports only PNG images,

img = plt.imread("flower.png")

Histogram Creation using numpy array:-

> To create a histogram of our image data, we use the hist() function.

> plt.hist(n_img.ravel(), bins=256, range=(0.0, 1.0), fc="k", ec='k')

Histogram Calculation:-

> Here we use cv2.calcHist() in-built function in opencv to find the histogram

> cv2.calcHist([images, channels, mask, histsize, ranges [, hist [, accumulate]]])

images: it is the source image of type uint8 or float32 represented as "[img]".

Channels: It is the index of channel for which we calculate histogram.

for grayscale image, its value is [0] and color image, you can pass [0], [1] or [2] to calculate histogram of blue, green or red channel respectively.

mask: mask image. To find histogram of full image, it is given as "none".

hist size: this represents our BIN count. For full scale, we pass [256].

ranges: this is our RANGE, Normally, it is [0, 256].

Algorithm

1. Open Colored 2D bitmap file in binary mode.
2. Read the header structure.
3. Extract the various feature
4. Print the values of features.

Conclusion:- Thus, we have successfully implemented a program for feature extraction in 2D Colour Images and plotted histogram for the features.

Assignment No: 05

problem Statements-
Build the web crawler to pull product information and links from an e-commerce website. (Python)

objective:-
To understand the working of web crawler and implement it.

Outcomes:
At the end of the assignment the students should have
1. understood how web crawler works.

Infrastructure:- Desktop/laptop System with Linux or its derivatives.

Software used:- Linux/windows os/virtual machine/ Jos/C/C++/Java.

theory:
Search Engines.
A program that search document for specified keywords and returns a list of documents where the keyword were found is a search engine. Although search engine is really a general class of programs, the term is often used to specifically describe systems like google, Alta Vista and Excite that enable users to search for documents on the words wide web and USENET newsgroup.

Typically a search engine works by sending out a spider to fetch as many documents as possible. Another program, called an indexer, then reads these documents and creates an index based on the words contained in each documents. Each search engine use a proprietary algorithm to create it indices. Such as, ideally, only meaningful results are returned for each query. Search engines are special sites on the web that are designed to help people find information stored on other sites. There are differences in the ways. Various search engines work but they all perform three basic tasks:

1) they search the Internet - based on important words.
2) They keep an index of the words they find and where they find them.
3) They allow users to look for words or combinations of words found in that index.

fig1 shows general search engine architecture. Every engine relies on a crawler module to provide the grist for its operation. Crawlers are small programs that browse the web on the search engine's behalf, similar to how a human user would follow links to reach different pages. The programs are given a starting set of URL's whose pages they retrieve from the web. The crawlers extract URLs.

appearing in the retrieved pages, and give this information to the Crawler control module. This module determine what links to visit next, and feeds the links to visit back to the Crawlers. The Crawlers also pass the retrieved pages into a page repository. Crawlers continue visiting the web until local resources, such as storage and exhausted.

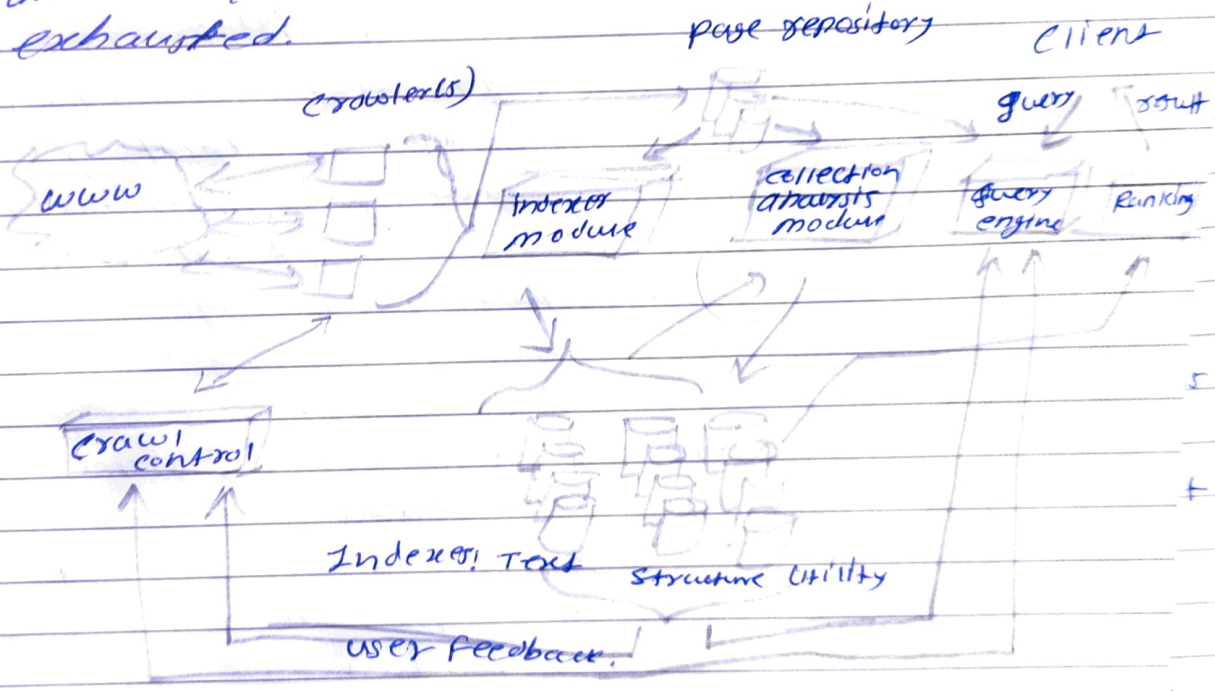


fig. general search engine architecture.

Web crawlers:

Web crawlers are programs that exploit the graph structure of the ~~speed of these~~ Progra web to move from the page to page. It may be observed thus the noun 'crawlers' is not indicative of the speed of there

programs, as they can be considerably fast.
A key motivation for designing web crawlers has been to retrive web pages and add them or their representations to a local repository. Such a repository may then serve particular application needs such as those of a web search engine. In its simplest form, a crawler starts from a seed page and then uses the external links within it to attend to other pages. The crawlers is the means by which web crawler collect pages from the web. It operates by iteratively downloading a web page, processing it, and following the links in that page to other web pages perhaps on other servers.

The end result of crawling is a collection of web pages, HTML or plain text at a central location practice

In a more traditional IR system the documents to be indexed are available locally in a database or file system. Web Crawlers first information retrival system was based on salton's vector-space retrival models the first system used a simple vector-space ritrival model. In the vector-space model, the queries and documents represent vectors in a highly dimensional word space. The components of the vector in a particular dimension is the significans of a documents, the component of the vector

along that ~~vector~~ words axis would be strong In this vector space, then, the task of querying becomes that of determining what documents vector are most similar to the query vector. Practically speaking this task amounts to comparing the query vector, components by component, to all the documents vectors that have a word in common with the query vector. Web Crawler determined a similarity number for each of these comparisons that formed the basis of the relevance score returned to the user.

Web Crawler's first IR system had three pieces: a query processing module, an inverted full-text index, and a metadata store. The query processing module parses the searcher's query, looks up the words in the inverted index, forms the result list, looks up the metadata for each result, and builds the HTML for the result page. The query processing module used a series of data structures and algorithms to generate results for a given query. First, this module put the query in a canonical form, and parsed each space-separated words in the query. Finally the query processor looked up each word in the dictionary and ordered the list of words for optimal query execution. Web Crawler's key contribution to distributed systems is to show that a reliable, scalable and responsive system can be built using

simple techniques for handling distribution, load balancing and fault tolerance.

Robot exclusion.

The robot exclusion standard, also known as the Robots-Exclusion protocol or robots.txt protocol, is a convention to prevent co-operating web crawlers and other web robots from accessing all or a part of a website which is otherwise publicly viewable. Robots are often used by search engines to categorize and archive web sites, or by webmasters to proofread source code.

The standard is different but can be used in conjuction with sitemaps, a robot inclusion standard for websites. A robots.txt file on a website will function as a request that specified robots ignore specified files or directories in their search. This might be, for example, out of preference for privacy from search engine results, or the belief that the content of the selected directories might be misleading or irrelevant to the categorization of the site as a whole, or out of desire that an application only operates on certain data. A person may not want certain pages indexed. Crawlers should obey the Robot Exclusion protocol

the Robots Exclusion protocol (REP) is a very simple but powerful mechanism available to webmasters and SEOs alike.

perhaps. It is the simplicity of the file that means it is often overlooked and often the cause of one or more critical SEO issues. To this end, we have attempted to pull together tricks, tips and examples to assist with the implementation and management of your robots.txt file. The file defines directives that exclude web robots from directories or files per website host. The robots.txt file defines crawling directives, not indexing directives. Good web robots adhere to directives in your robots.txt file. Bad web robots may not. Do not rely on the robots.txt file to protect private or sensitive data from search engine. For example: website analytics folders (/web stats/, /stats/ etc.) Test or development areas (/test/, /dev/) Xml sitemap element if your URL structure contains vital taxonomy.

If a URL redirects to a URL that is blocked by a robots.txt file, the first URL will be reported as being blocked by robots.txt in google webmaster Tools. Search engines may cache your robots.txt file. (for example: google may cache your robots.txt file for 24hours) when deploying a new website from a development environment always check the robots.txt file to ensure no key directories are excluded. Excluding files using robots.txt may not save the

crawl budget from the same-crawl session. for example: if google cannot access a number of files it may not crawl other files in their place. URLs excluded by REP (Robots Exclusion protocol) may still appear in a search engine index.

Program Implementation: code written in Python to implement of the same with appropriate output.

Algorithm.

1. make user Interface
2. Input the URL of any website
3. establish HTTP Connection
4. read HTML page source code
5. Extract Hyperlinks of HTML page
6. Display the list of hyperlinks on the same page.

Conclusion:

Implementation is concluded by stating the basic working of web crawler.

Assignment No:- 06

Problem Statement:-
Write a program to find the live weather report (temperature, wind speed, description, and weather) of a given City. (Python).

Objective:-
1. To Get Weather Information using Python
2. To Evaluate the performance of the IR System and understand User interface for Searching.
3. To understand Information Sharing on the Web.

Outcomes:-
At the end of the assignment the student should have
1. Understood and implemented the program to find the live weather report using Python.

Infrastructure:- Desktop/Laptop System with Linux or its derivatives.

Software used:- LINUX/Windows OS/Virtual machine/Tools Python 3.9.

Theory:
Weather Information using Python.
Python is a growing language that has become increasingly popular in the programming community. one of Python's key features.

is that it's easy to work with APIs on websites and many weather entities have their API which you can access with just a couple of lines of code. one such great API is the open weather map's API, with it you can built a small program to access any given location's weather forecast anywhere across the globe. This article will help you to get weather information using Python.

What is Open WeatherMap?

The OpenWeatherMap(OWM) is a helpful and free way to gather and display weather information. Because it's an open-source project, it's also free to use and modify in any way. OWM offers a variety of features and is very flexible. Because of these qualities, it offers alot of benefits to developers. one of the major benefits of OWM is that it's ready to go unlike other weather applications. OWM has a web API that's ready to use You don't have to install any software or set up a database to get it up and running. This is a great option for developers who want to get weather reading on a website quickly and efficiently.

It has an API that supports HTML, XML and JSON endpoints. Current weather information extended forecasts, and graphical maps can be requisted by users. These maps show cloud cover, wind speed as well as pressure and precipitation.

Conclusion:-

Thus, we have successfully Implemented a program to find the live weather report (temperature, wind speed, description, and weather) or a given city using Python.