

A DISSERTATION
ON
**GENETIC ALGORITHM BASED WEB PAGE
CATEGORIZATION**

SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by:

SHASHIKANT MADIA
University Roll No.:- 2K14/SWE/16

Under the supervision of

DR. O.P.VERMA
HOD, Computer Science and Engineering Department, DTU



2014-2016

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
DELHI TECHNOLOGICAL UNIVERSITY
DELHI – 110042, INDIA



Department of Computer Engineering Delhi Technological University

Certificate

I, **ShashikantMadia**, hereby declare that the work which is being presented in my M.Tech dissertation entitled "**GENECTIC ALGORITHM BASED WEB PAGE CATEGORIGATION**", in partial fulfillment of the requirement for the award of the degree of **Master of Technology (Computer Engineering)** submitted to the Department of Computer Engineering, Delhi Technological University, Delhi is an authentic record of my own work carried out under the supervision of **Prof. O. P. Verma, Head of Department, Delhi Technological University, Delhi**. The work presented in this thesis has not been submitted by me for the award of the degree elsewhere.

Date:

ShashikantMadia

Place: Delhi

Roll No. 2K14/SWE/16

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Prof. O. P. Verma

Place: Delhi

Head of Department

Department of Computer Engineering

Delhi Technological University

DECLARATION

Foremost, I would like to express my sincere gratitude to my Supervisor **Prof. O. P. Verma**, Head of Department of Computer Engineering for his continuous encouragement, patience, motivation, enthusiasm, and immense knowledge. His guidance and insightful comments helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for this thesis.

The thesis has been kept on track and been seen through to completion with the support of numerous people including my friends, well-wishers. At the end of my thesis, I would like to express my thanks to all those who contributed in many ways to the success of this study. Last but not the least, I would like to thank my parents for their unconditional support, both financially and emotionally throughout my research.

SHASHIKANT MADIA
Roll no: 2K14/SWE/16
M.Tech (Software Engineering)

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my Supervisor **Prof. O. P. Verma**, Head of Department, Computer Science Engineering for his continuous encouragement, patience, motivation, enthusiasm, and immense knowledge. His guidance and insightful comments helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for this thesis.

The thesis has been kept on track and been seen through to completion with the support of numerous people including my friends, well-wishers. At the end of my thesis, I would like to express my thanks to all those who contributed in many ways to the success of this study. Last but not the least, I would like to thank my parents for their unconditional support, both financially and emotionally throughout my research.

SHASHIKANT MADIA
Roll no: 2K14/SWE/16
M.Tech (Software Engineering)
Department of Computer Science and Engineering
Delhi Technological University
Delhi – 110042

ABSTRACT

The incredible increase in the amount of information on the World Wide Web has caused the birth of topic specific crawling of the Web. During a focused crawling process, an automatic Web page classification mechanism is needed to determine whether the page being considered is on the topic or not. In this study, a genetic algorithm (GA) based automatic Web page classification system is developed which uses both HTML tags and terms belong to each tag as classification features. With such a huge amount of data on web, search engine need some mechanism that gather pages from the Web in order to index them so that results are returned to the users according to their need. To achieve this, Web Page Categorization comes into existence. This can be down using a focused crawler which categorizes different web pages to some predefined categories. Web crawling is the process which downloads Web pages to support a search engine. Downloading all Web pages results in wastage of hardware and software resources. Focused Web crawler seeks, gathers and maintains pages relevant to pre-defined set of topics rather than downloading all the documents. Genetic algorithm is used in focused crawler to get optimised categories which are further updated for WebPage classification to extract documents from index ableWeb. Literature review is performed based on focused Web crawler classification. We used Genetic Algorithmbased focused crawler which gives best features for categorization. This work results in high relevancy and more coverage considering indexable Web.

List of Figures

Figure 1 Traditional WebPage Categorization.....	2
Figure 2 Subject Classification.....	2
Figure 3 Functional Classification.....	3
Figure 4 Binary Classification.....	4
Figure 5 Multiclass classification.....	4
Figure 6 Single-label classification.....	5
Figure 7 Multilabel Classification.....	5
Figure 8 Soft Classification.....	6
Figure 9 Flat classification.....	7
Figure 10 Hierarchal Classification.....	7
Figure 11 Architecture of Web crawler.....	8
Figure 12 Focused Web crawler architecture.....	13
Figure 13 Proposed methodology.....	23
Figure 14 Genetic algorithm based classifier.....	26
Figure 15 Proposed algorithm for mutation.....	29
Figure 16 Proposed classification process.....	31
Figure 17 Seed URLs for indexible Web.....	34
Figure 18 Feature set.....	38
Figure 19 Crawler sees different from user sees.....	43
Figure 20 Information generated at run time.....	44
Figure 21 Document formation in 2D matrix	46
Figure 22 Average fitness in each iteration.....	47
Figure 23 Best fitted chromosome with weights.....	47
Figure 24 Stanford University crawled data.....	48

List of Tables

Table I Various open source Web crawlers	11
Table II Literature survey of focused Web crawler.....	18
Table III Example of crossover operation.....	28
Table IV Accuracy Comparison between Naïve Bayes classifier Proposed.....	46
Table V Example of crossover operation.....	46

Contents

CERTIFICATE	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT.....	iv
List of Figures	v
List of Tables	vii
CHAPTER 1: INTRODUCTION	1
1.1. Classification of Web pages	1
1.1.1 Traditional WebPage Categorization.....	2
1.1.2 Subject Classification.....	2
1.1.3 Functional Classification.....	2
1.1.4 Binary Classification.....	3
1.1.5 Multiclass classification.....	4
1.1.6 Single-label classification.....	5
1.1.7 Multilabel Classification	5
1.1.8 Soft Classification.....	6
1.1.9 Flat classification.....	6
1.1.10 Hierarchal Classification.....	7
1.1.11 Architecture of Web crawler	7
1.2. Applications of crawlers	8
1.2.1 Web Search Engines:.....	8
1.2.2 Web Archiving:	9
1.2.3 Web Mining:.....	9
1.2.4 Social Network Analysis:	9
1.3. Types of web crawlers	9
1.3.1 General purpose crawler:.....	9
1.3.2 Focused Crawler:.....	9

1.3.3 Incremental Crawler:	10
1.3.4 Parallel Crawler:	10
1.3.5 Distributed Crawler:	10
1.4. Different open source Web crawlers	10
1.5. Focused Web Crawler	12
1.6. Architecture of focused Web Crawler	12
1.7. Issues of focused Web crawler	13
1.8. Hidden Web crawler	14
CHAPTER 2: LITERATURE SURVEY.....	15
2.1. Literature survey on focused Web crawler	15
2.1.1. Soft computing methods:.....	15
2.1.2. Other techniques:.....	17
2.1.3 : RESEARCH GAPS	20
2.2PROBLEM FORMULATION AND OBJECTIVES	21
2.2.1. Problem statement:.....	19
2.2.2. Objectives:	19
2.3 PROPOSED TECHNIQUE DESIGN	20
2.3.1. Indexible Web Methodology	21
2.3.2URL filtering.....	24
2.3.4. Feature extraction.....	24
2.3.5. Document formation	24
2.3.6. Genetic algorithm based classifier:	25
2.3.6.1. Coding	26
2.3.6.2. Initial population.....	27
2.3.6.3. Evaluation of population	27
2.3.6.4. Selection	27
2.3.6.5. Crossover	28
2.3.6.6. Mutation.....	29
2.3.6.7. Generation of new population	30
2.3.6.8. Termination condition	30
2.3.7. Classification.....	30
2.3.8. Category selection.....	32

CHAPTER 3: EXPERIMENTAL SETUP	33
3.1. Hardware and software configuration:	33
3.2. DataSet:	33
3.3. Input data:	34
3.3.1.List of surnames:	35
3.3.2.List of cities:	35
3.3.3. List of institutes:	35
3.3.4.List of departments:	35
3.3.5.List of designation:	35
3.3.6.URL filtering list:	35
3.3.7.Feature set:.....	35
3.4. Genetic algorithm parameters:	36
3.5. Platforms and technologies:	36
3.5.1.JAVA:.....	36
3.5.2.NetBeans IDE:.....	36
3.5.3.WordNet library:.....	37
3.5.4.Jsoup library:	37
3.5.5.Oracle:	37
3.5.6.Selenium Web driver:.....	37
3.6. Implementation details:	37
3.7. Hurdles faced in implementation	39
3.7.1. URL meta information is empty	39
3.7.2. Crawling webpage	39
3.7.3 Crawler sees different from user sees.....	40
3.7.4. Information in image form	41
3.7.5. Finding form interface	41
3.7.6. Irrelevant links containing relevant useful links.....	41
 CHAPTER 4: RESULTS	 42
4.1. Genetic algorithm results	42
4.2. Crawled data for different universities	44

4.3. Recall computation for crawled data	45
4.4 Comparison with other techniques	45
CHAPTER 5: CONCLUSION AND FUTURE SCOPE	47
REFERENCES	48

CHAPTER 1

INTRODUCTION

Webpages provides a lot of information to the computer users. Websites is a collection of Web Pages which contains specific information. In this scenario to find or retrieve particular page or information is hard task. So to solve this problem easily there are different web page classification methods. Using this method we can identify web pages. Based on web page information i.e .content of particular page we have to classify the web page. Web page classification comes under the domain of web mining. Web mining is the addition of information gathered by traditional data mining methodologies and techniques with information gathered over the internet. Web page classification retrieves web pages based on different features consists of tags and termson web pages.The general view of web page Classification can be divided into multiple types: Subject Classification deals with the subject or topic of web page. Functional Classification deals with the role web page plays. Sentimate Classification concerns about opinion presented in the web page. Binary Classification divides each instance into one of the two categories. Multi-class Classification cares for more than two classes. Multiclass Classification can be further catrgorized into single-label and multi-label classification. Flat Classification in that categories are measured in parallel, i.e., one category does not supersede another. Hierarchical Classification the categories are divided into a hierarchical tree-like structure, in which each category may have a number of subcategories. There are many ways for Web Page Classification such as using the content of the web page, using the structure of web page, using clustering methods etc. This paper is focuses on the Web Page Classification using a genetic algorithm based focused crawler.

1.1 Web Page Classification

Web page classification, also known as Web page categorization, is the process of assigning a Web page to one or more predefined category labels “News”, “Sports”, “Business”, ETC. Classification is traditionally posed as a supervised learning problem in which a set of labeled data is used to train a classifier which can be applied to label future examples.

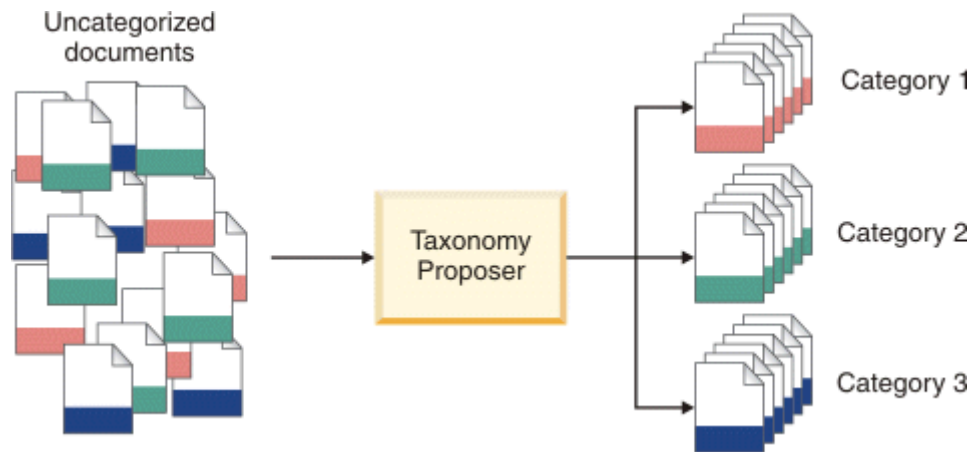


Fig1. Traditional WebPage Categorization

1.1.1. **Traditional WebPage Categorization:** The general problem of Web page classification can be divided into more specific problems: subject classification, functional classification, sentiment classification, and other types of classification

1.1.2 **Subject classification:** Subject classification is concerned about the subject or topic of a Web page. For example, judging whether a page is about “arts,” “business,” or “sports” is an instance of subject classification.

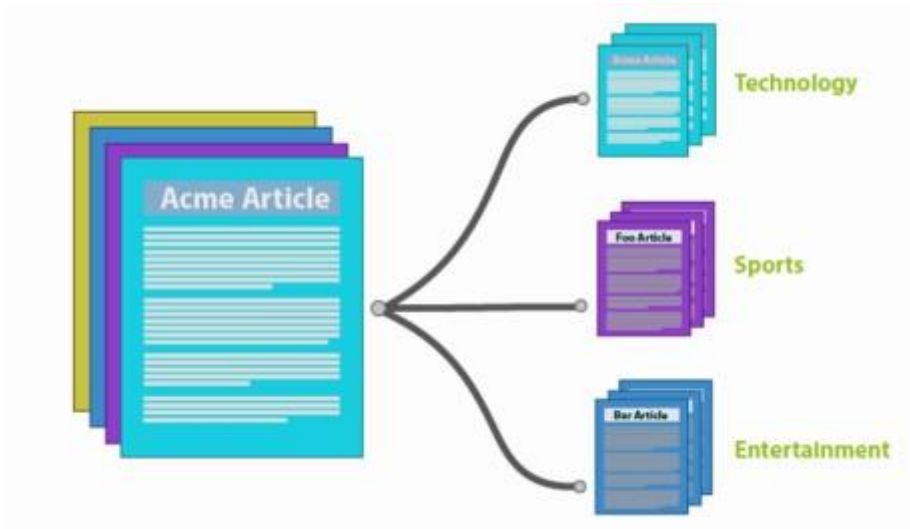


Fig 2: Subject Classification

1.1.3 **Functional Classification:** Functional classification cares about the role that the Web page plays. For example, deciding a page to be a “personal homepage”, “course page” or “admission page” is an instance of functional classification.

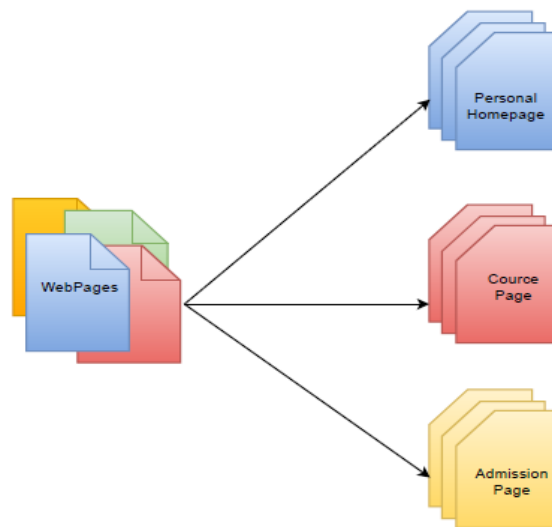


Fig3: Functional Classification

Other Type of classification: Other types of classification include genre classification, search engine spam classification and so on.

Types of Classification:

Based on number of classes: Based on the number of classes in the problem classification can be divided into

- i. Binary classification and
- ii. Multiclass classification

1.1.4 Binary Classification: Binary classification categorizes instances into exactly one of two classes either the first one or the second. This can be well illustrated with the help of the diagram below.

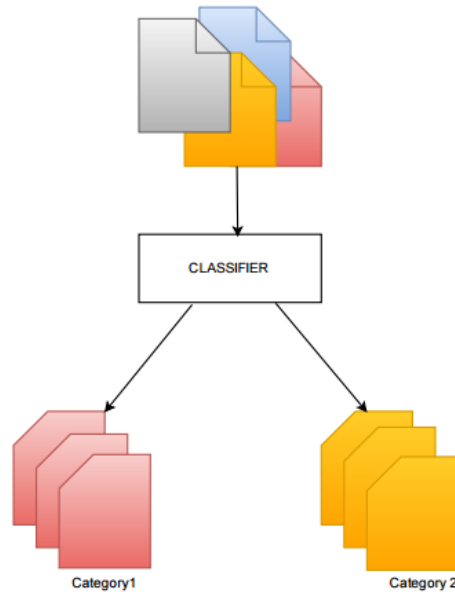


Fig4: Binary Classification

1.1.5 Multiclass Classification: Multiclass classification deals with more than two classes. If a problem is multiclass, for example, four-class classification, it means four classes are involved, for example, Arts, Business, Computers, and Sports. It can be either single-label, where exactly one class label can be assigned to an instance or multilabel, where an instance can belong to any one, two, or all of the classes.

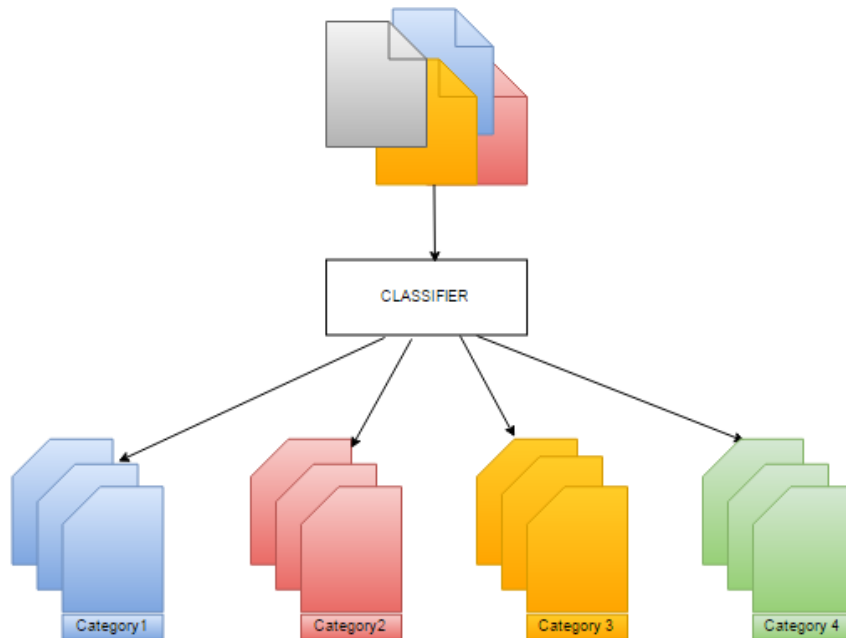


Fig5: Multiclass classification

Based on the number of classes that can be assigned to an instance, classification can be divided into

- i. Single-label classification and
- ii. Multilabel classification.

1.1.6 **Single-label Classification:** In single-label classification, one and only one class label is to be assigned to each instance,

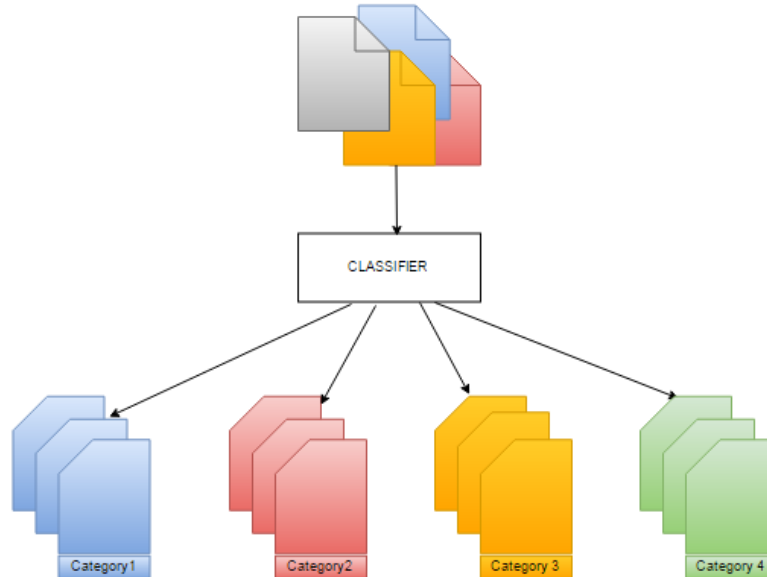


Fig5: Single-label classification

1.1.7 **Multilabel classification:** In multilabel classification, more than one class can be assigned to an instance.

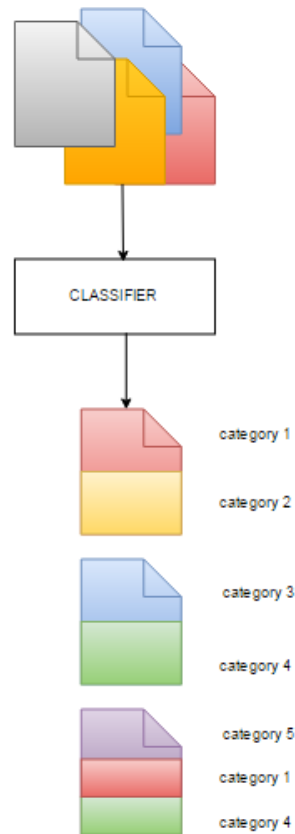


Fig6: Multilabel Classification

Based on the type of class assignment, classification can be divided into

- i. Hard classification and
- ii. Soft classification.

Hard Classification: In hard classification, an instance can either be or not be in a particular class, without an intermediate state.

Soft classification: In soft classification, an instance can be predicted to be in some class with some likelihood (often a probability distribution across all classes).

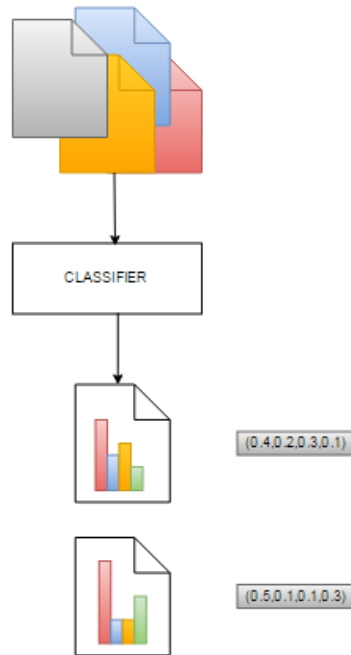


Fig7: Soft Classification

1) **Based on the organization of categories:** Web page classification can also be divided into

- i. Flat classification and
- ii. Hierarchical classification.

1.1.9 **Flat Classification:** In flat classification, categories are considered parallel, that is, one category does not supersede another.

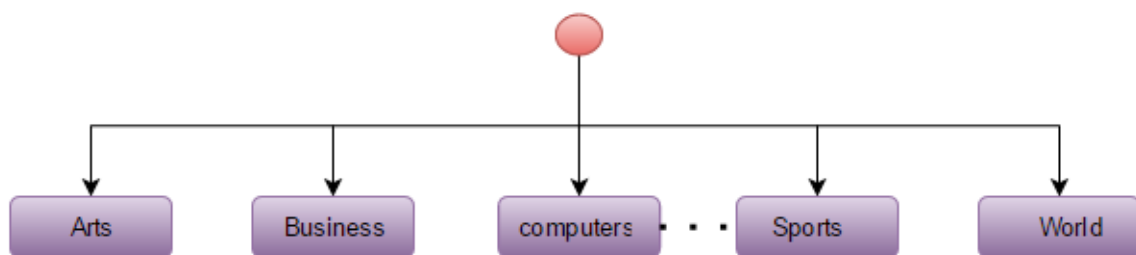


Fig8: Flat classification

1.1.10 **Hierarchical Classification:** In hierarchical classification, the categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories.

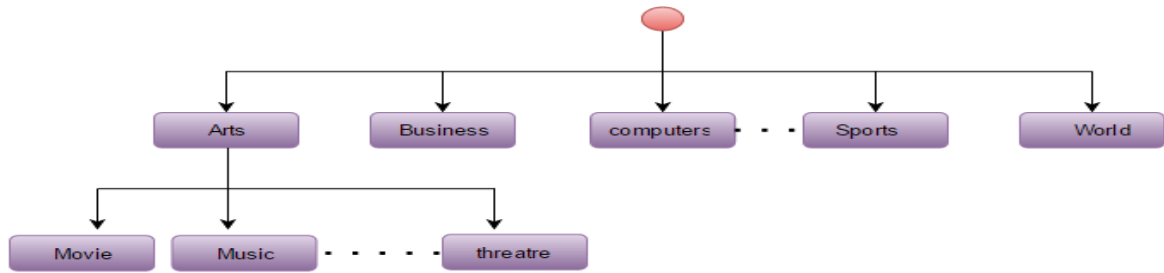


Fig9: Hierarchal Classification

1.1.11 Architecture of Web crawler

The following figure shows the architecture of Web crawler. The main components of Web crawler are scheduler, downloader, Queue of the URLs. Scheduler arranges the URLs to be processed by the downloader based on the various parameters such as priority and using certain techniques.

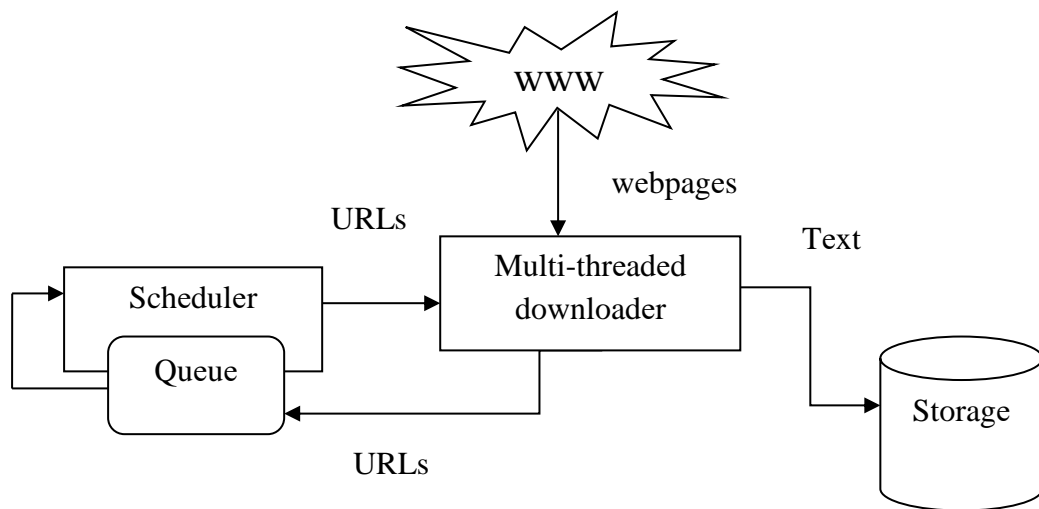


Figure 10 Architecture of Webcrawler(Search engine-essential information 2011)

1.2 Applications of crawlers

Web Crawlers are used for various purposes as given in (Olston&Najork 2010). Some of them are listed below

Web Crawlers are used for various purposes as given in (Olston&Najork 2010). Some of them are listed below:

1.2.1 Web Search Engines: A Web Search Engine is a system that assembles a repository of webpages and indexes them according to some policy. Search engine respond to user queries by returning webpages from its repository that match the query. A Web Crawler is an important component of Search Engine that is responsible for bulk downloading of webpages in an automatic manner. Also it has another important function of refreshing the repository of webpages.

1.2.2 Web Archiving: Web Archiving is the process of collecting portions of World Wide Web and storing in an archive so that the information is preserved for future use by researchers, academicians, historians etc. Some organisations also need to archive their Web content for corporate heritage, legal or regulatory purposes. Web Crawlers are used in Web Archiving for automating the process of collection of webpages. Examples of Web Crawlers used for Web archiving are Heritrix, HTTrack, Wget etc.

1.2.3 Web Mining: Web Mining is the process of applying data mining techniques for discovering patterns from the Web. Web mining can be divided into three categories that are Web Usage Mining, Web Content Mining and Web Structure Mining. Web Usage mining deals with the user's behaviour while using World Wide Web. Web Content mining is mining or extracting information from the webpage contents. Web Structure mining analyses the structure of website, using graph theory. Crawlers are used to collect data from the Web in Web Structure mining and Web Content Mining. Many open source crawlers like crawler4j, Nutch, Heritrix, GRUB, WebSPHINX, etc. have been used for Web mining applications.

1.2.4 Social Network Analysis: Social Network theory is concerned with properties related to connectivity and distances in graphs. Examples of social networks include network between papers through citations, network between people on social networking sites and network between webpages by hyperlinking to other webpages. Social network analysis has diverse applications like espionage, citation indexing etc. Crawlers are used for collecting datasets that are analysed using graph theory.

1.3 Types of web crawlers

Web Crawlers can be categorised into many types depending upon the strategy followed by them for crawling and the goal they want to achieve (Udapure 2014).

1.3.1 General purpose crawler: The general purpose crawlers download all the webpages without regard to any specific topic. The aim is to cover as much web as possible within given time. General purpose crawlers are also known as Universal crawlers. These are large scale

crawlers and incur high cost in terms of network bandwidth usage, but this cost is amortised over many number of queries by users. Also, the repository needs to be updated more often in general purpose crawlers. These are generally used by Universal Search Engines.

1.3.2 Focused Crawler: Focused crawlers aim at downloading pages on specific topic or subject. It is also known as topical crawlers. Since Focused crawling is subject specific, it minimizes the usage of resources like time, space and network bandwidth. The goal of Focused Crawling is to download relevant pages keeping the number of irrelevant pages downloaded to minimum. Focused crawlers are based on the observation that relevant pages point to other relevant pages either directly or through path of links of short length.

1.3.3 Incremental Crawler: The main goal of the Incremental Crawler is keep the repository of webpages updated all the time. An Incremental Crawler refreshes the existing repository of webpages incrementally. Depending upon the change frequency of webpages, it visits the webpages with high change rate more frequently and the other webpages less frequently. The advantage of Incremental Crawler is that it saves network bandwidth, since only webpages with high change rate are downloaded instead of all webpages being downloaded.

1.3.4 Parallel Crawler: When multiple crawlers run in parallel, this configuration is called Parallel Crawler. A Parallel crawler basically consists of several crawling processes which run simultaneously on the World Wide Web. Links to be crawled are divided among the multiple crawling processes depending upon some criteria. The parallel crawler can be geographically distributed or can be on local network. Its benefit is that use of multiple crawling processes reduces the total crawling time significantly. Thus, bulk of webpages can be downloaded in reasonable amount of time.

1.3.5 Distributed Crawler: Distributed Crawlers are based upon the technique of distributed computing. In order to achieve wide coverage of the Web, many crawlers are geographically distributed on the Internet and a central server is used for management of communication and synchronization of the distributed nodes. Each crawler does the crawling of the part of the Web assigned to it. Its advantage is that it is robust against the system crashes. It also uses the principle of load balancing.

1.4 Different open source Web crawlers

There are various open source Web crawlers which can be used according to the need. Each crawler has their own properties. Following table shows the open Web crawlers comparison based on theoretical features. We did comparison to check which open source Web crawler

contains more features and easy to operable so that, that open source Web crawler can be used for the implementation according to the need of the application. Star mark (*) in the table shows the presence of some feature in the open source Web crawler, whereas blank shows that presence or absence of the feature isn't known. Features mentioned here are as described

1. **Flexibility:** Crawler is operable with the changing environment.
2. **Scalable:**The crawler architecture permit scaling up the crawl rate by adding extra machines and bandwidth.
3. **Extensible:** Crawler should be designed to be extensible in many ways to cope with new data formats, new fetch protocols and so on.
4. **Distributed:**The crawler should have the ability to execute in a distributed fashion across multiple machines.
5. **Cross platform:** Crawlers can operate on multiple platforms.
6. **Multithreaded:** To achieve better parallelism by dividing the crawling process among separate independent threads.
7. **Configurable:** Crawler be highly configurable allowing definition of :
 1. stop / resume crawl
 2. item type inclusion / exclusion rules
 3. multiple start urls per source (Web site)
 4. cache crawled items
8. **Focused:** Crawler focused to particular topic or universal.
9. **Interface:** Environment provided to the developer/user.
10. **Index:**Some type of indexing done by Web Crawling tools.

Table I: Comparison of different open source Web crawlers

Features	Nutch	Scrapy	Heritrix	Norconex http collector	Crawler 4J	YaCy	Web sphinx	Jspider	Xapian	Ebot
Language	Java	Python	Java	Java	Java	Java	Java	Java	C++	Erlang
Flexible	*	*	*	*				*		
Dynamic ally Scalable	*							*	*	*
Extensible	*	*	*	*	*			*		

Cross platform	*	*	*	*		*	*		*	Linux
Multi-threaded	*		*	*	*		*		Does not provide explicitly	
Distributed	*			*		*		*		*
Configurable			*		*		*			*
Focused	*	*	*	Universal						
Interface	CL	CL	Both			GUI	GUI		CL	
Index	Lucene		Arc files			NoSQL			Omega	NoSQL

1.5 Focused Web Crawler

Web crawler downloads all the documents from the Web. But the process of downloading all the documents results in wastage of hardware and software resources. Focused Web crawler (Chakrabarti et. al. 1999) is a Web crawler that seeks, acquires, indexes and maintains the pages that are relevant to a pre-defined set of topics rather than collecting and indexing all accessible documents over the Web. A focused Web crawler analyzes, to locate the links that are likely to be most relevant for the crawl i.e. relevancy based on the pre-defined set of topics, thus avoiding irrelevant regions of the Web.

For evaluation of focused crawlers many metrics are used like recall, precision, harvest ratio, etc. Recall is the ratio of number of relevant pages retrieved by the total relevant webpages in the repository. Precision is the ratio of relevant webpages retrieved by the total number of webpages in the repository. Harvest ratio is the number of relevant webpages retrieved by the total number of webpages retrieved by crawler from the repository.

1.6 Architecture of focused Web Crawler

The following figure shows the architecture of focused Web crawler (Chakrabarti et. al. 1999). The main components are classifier, watchdog priority controls, worker thread and distiller. Watchdog priority controls picks the URLs from the crawl table, assign URLs to the memory buffers based on the priority of the URLs and the memory buffers. Watchdog priority controls is responsible for load balancing among the memory buffers. Memory buffers are basically the priority queues which are processed concurrently on worker threads. Filtering of the documents

is performed based on the topic models generated by trainer.

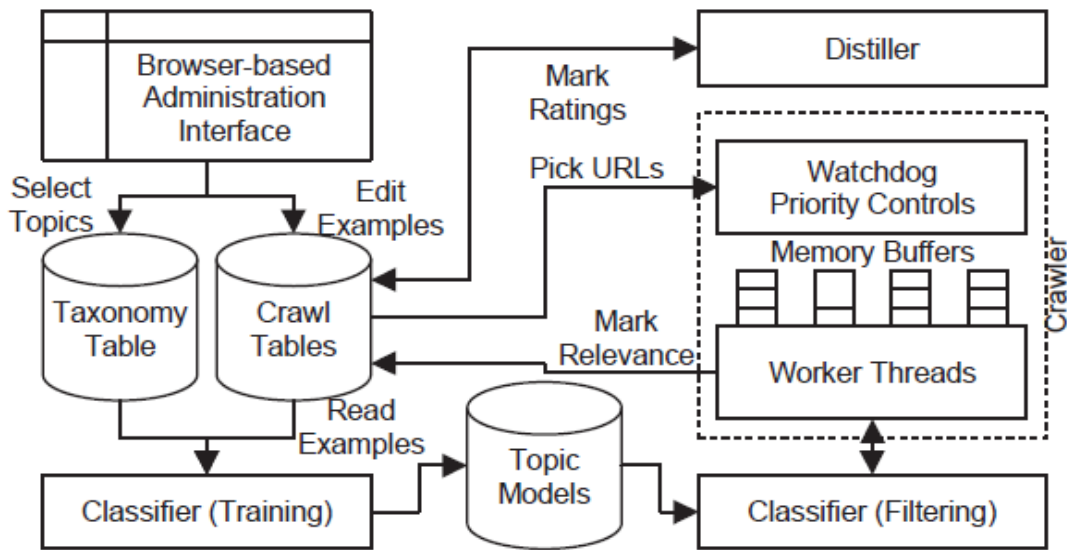


Figure 12 Focused Web Crawler Architecture (Chakrabarti et. al. 1999)

Worker thread picks one of the URL from these memory buffers and filtering of the document is performed. Worker thread, after receiving information from the filter, computes the relevance of the document. Based on the relevancy of the document, outlinks of the documents are fetched and process continues, until memory buffers are empty.

For evaluation of focused crawlers many metrics are used like recall, precision, harvest ratio, etc. Recall is the ratio of number of relevant pages retrieved by the total relevant pages in the repository. Precision is the ratio of relevant pages retrieved by the total number of pages in the repository. Harvest ratio is the number of relevant pages retrieved by the total number of paged retrieved by crawler from the repository.

1.7 Issues of focused Web crawler

There are various issues of focused Web crawler. Some of them are listed below.

1. Focused crawler suffer from tunneling problem i.e. inability of focused crawler to tunnel on-topic pages by following the links of off-topic pages.
2. Heavy network load.
3. **Focused crawler not able to target hidden Web content.**

1.8 Hidden Web crawler

Focused Web crawler gathers pages based on pre-defined set of topics but the major drawback of focused Web crawler is that focused Web crawler is unable to target hidden Web content. Large amount of information present in hidden Web which cannot be accessed by simply following the links of indexable Web but can be accessed through search forms and query interface that lead to Web accessible databases, which leads to more relevant information. Hidden Web crawler is a Web crawler which sends queries to the form interfaces to gather information returned from the hidden Web databases. In this, focused Web crawler is using genetic algorithm to access hidden Web content along with accessing content from indexable Web.

CHAPTER 2

LITERATURE SURVEY

2.1. Literature survey on focused Web crawler

(S. Chakrabarti et al 1999) first introduced the term focused crawling and implemented a focused crawler using Yahoo taxonomies. Bayesian classifier was used to determine if the current page is relevant to topic or not. Each topic had initial seed pages associated with it and the neighbouring pages (pages with links on that page) of the page currently visited by the crawler based on the output of the classifier in the form of relevancy of the page. The use of taxonomy also helps at better modelling of the negative class as irrelevant pages are usually not drawn from a homogenous class but could be classified in a large number of categories with each having different properties and features. Harvest ratio is computed based on the relevant pages. Harvest ratio is the number of relevant pages retrieved by the total number of paged retrieved by crawler from the repository and robustness and acquisition rate of resources so that there would not have over utilization of resources.

(Korde, Vandana, and C. NamrataMahender 2012) discussed about text classification process in which firstly pre-processing of the documents is performed to present document into clear word format, then indexing is performed to create vector space for the terms present in the document, then feature selection phase to select subset of features from the documents, then classification is performed and based on the classification performance of the classification process is measured based on precision and accuracy etc. In this, they have compared various classification techniques such as NN, Bayesian classifier, GA, SVM, K-NN, decision tree considering their methodologies along with their advantages and disadvantages.

2.1.1. Soft computing methods:

Soft computing as an emerging approach to computing which parallels remarkable ability of human mind to reason and learn in an environment of uncertainty and imprecision. (Deshmukh, Ankit R., and Sunil R. Gupta 2014) had mentioned soft computing which consists of several computing paradigms like Neural Networks, Fuzzy Logic, and Genetic algorithms and support vector machine.

(Li, Jun, Kazutaka Furuse, and Kazunori Yamaguchi 2005) proposed a method which uses a decision tree on anchor text of hyperlinks. They have taken two assumptions as crawl in limited domain and entry page presence to the URL domain. Decision tree is used to predict relevance of target pages and a graph is created. Training data has represented as relevant and non-relevant pages and positive and negative examples. Training is performed using SVM by considering Boolean parameters for relevancy. For each relevant page, shortest path is calculated using dijkstra algorithm. They have ignored hyperlinks whose anchor text is blank. Based on their observation, they improved recall and allowed to search deep relevant pages.

(Luong, Hiep Phuc, Susan Gauch, and Qiang Wang 2009) attempted to automate the entire ontology learning process from the collection of domain-specific literature, to text mining to build new ontologies or enrich existing ones. The process contains initial set of words, based on those words queries are generated in which short queries result in less number of relevant results whereas long queries containing keywords and .pdf formats results in more relevant results. Training based on LibSVM classifier which performs classification that separate the hyperplane into two classes. SVM based filtering technique that automatically filters out the non-relevant documents collected by the crawler so that only those most likely to be relevant are passed along for information extraction.

(Özel, Selma Ayşe 2012) describes features extraction and selection through tags and terms associated with the tags. Based on the tags and terms, categories are created. For each document, category-document similarity is computed based on cosine similarity and check whether document is positive document or negative for a category. Genetic algorithm is used in which chromosomes are the set of categories and fitness function is computed based on more number of positive documents for a category, then find the probability and cumulative probability and choosing randomly to select two chromosomes for reproduction. Then, all newly generated and old chromosomes are sorted and pop-size top chromosomes are picked for next iteration.

(Singh, Chain, Ashish Kr Luhach, and Amitesh Kumar 2013) describes a method in which based on the query, n documents are returned and keywords of all those documents are arranged and documents are marked as [0,1] based on the presence and absence of those keywords in the document. All those vectors are initial set of populations. GA is applied on the documents to get more relevant documents so that outlinks of the most relevant document can

be fetched and process can be continued. GA is basically used to arrange documents in the queue based on their relevance computed using their fitness function. All the new and old terms are entered into the google and based on returned documents average relevance is calculated.

(Belmouhcine, Abdelbadie, and Mohammed Benkhalifa. 2015) describes a method in which there are three phases named as pre-processing, classification and evaluation measures. While pre-processing, Web graph of the Web page is created in which there are two representations of the neighbors as boolean or weighted neighbor vector representation. Neighbours were taken because correlation is computed between the label of a page and attribute of page or the Webpages in the neighborhood of the page. They have taken weighting scheme as LF-IPF (Link Frequency- Inverted Page Frequency) similar to TF-IDF. Classification is performed using NB, K-NN, SVM and random forest. Evaluation measures are based on micro- averaging and macro- averaging techniques.

2.1.2. Other techniques:

There are various other techniques which can also be useful for the classification. [Korde, Vandana, and C. NamrataMahender 2012]These are as described as Naïve bayes, link scoring, document scoring, random forest, associate classifier, centroid based classifier, linear least square fit etc.

(Wang, Wenxian, et al. 2010) used naïve bayes classifier to estimate the page rank. Three sub processes named as page analysis, characteristic extraction and relevancy analysis were described in the classifier phase whereas other phases of crawler perform their operations as done previously. Page analysis analyse the content of the page to extract information in order to decide which links to follow. Characteristics extraction is done in the form of vectors using TF-IDF algorithm and Bayesian algorithm is used to compute the relevancy of the pages and to which class that page belongs.

(Taylan, Duygu, MitatPoyraz, SelimAkyokuş, and Murat Can Ganiz 2011) focused on the classification of links instead of downloading the pages to compute the relevancy of the page. They used link scoring to decide which links to crawl and in what order they are arranged without downloading all the pages. Html parser and link extractor extract the links of the Web page and analyse only the links of the Web page instead of the contents. Relevance is calculated based on the weights of the terms present in the document. Link scoring for relevant pages is based on Naive Bayes classifier and cosine similarity. Link scoring for irrelevant pages is like

tunneling in which based on max tree length to be taken links are analysed of the pages which are retrieved from following links of irrelevant page.

(Rajesh. L, Shanthi. V 2012) performs the classification of documents based on TF-IDF and cosine similarity. Their proposed work is firstly, there is a removal of stop or stem words, calculate the frequency of remaining words in the documents, match those words with the set of topics and mark 0 or 1 depending on the presence or absence of the words in the document, then they compute the TF-IDF of the documents and classify the documents into dictionary. They not only considered how frequently a word occurs in documents but also how frequently a word appears in document collections.

Table 2: Literature survey of focused Web crawler

Authors	Journal/Conference, Year	Title	Classifier used	Relevance prediction technique	Performance parameters	Subject system
Chakrabarti, Soumen, Martin Van den Berg, and Byron Dom.	Computer Networks, Elsevier (J)1999	Focused crawling: a new approach to topic-specific Web resource discovery	Bayesian classifier	Hypertext classifier and distillation	Robustness, acquisition rate of resources, harvest ratio	Yahoo, Alta vista search engine
Korde, Vandana, and C. NamrataMahender	International Journal of Artificial Intelligence & Applications (IJAIA) (J) 2012	Text classification and classifiers: A survey	Bayesian classifier, Decision Tree, K-NN, SVMs, NNs	Not applicable	Effectiveness of the classifier	Not applicable
Li, Jun, KazutakaFuruse, and Kazunori Yamaguchi	International conference on World Wide Web. ACM (C) 2005	Focused Crawling by Exploiting Anchor Text Using	SVM	Decision tree	Recall	University of Tokyo, Kyoto University, Keio University

		Decision Tree				
Luong, HiepPhuc, Susan Gauch, and Qiang Wang	International Conference on Information, Process, and Knowledge Management. IEEE (C) 2009	Ontology-based Focused Crawling	SVM (LiBSVM Classification tool)	SVM	Accuracy	Amphibian Morphology Ontology, interactive ontologybased query system
Authors	Journal/Conference	Title	Classifier used	Relevance prediction technique	Performance parameters	Subject system
Özel, Selma Ayşe	Expert Systems with Applications. Elsevier(J) 2011	A Web page classification system based on a genetic algorithm using tagged-terms as features	Genetic algorithm	Genetic algorithm	Accuracy	Dmoz (open directory project) and google.com
Singh, Chain, Ashish Kr Luhach, and Amitesh Kumar	International Journal of Computer Applications(J) 2013	Improving Focused Crawling with Genetic Algorithms	Genetic algorithm	Genetic algorithm	Average relevance	Google.com
Belmouhcine, Abdelbadie, and Mohammed Benkhalifa.	Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics. ACM (C) 2015	Implicit Links based Web Page Representation for Web Page Classification	SVM, NB, K-NN, random forest	Not applicable	Precision, recall	Open Directory Project.

Wang, Wenxian, et al.	Intelligent Information Technology and Security Informatics (IITSI). IEEE (C) 2010	A focused crawler based on naive bayes classifier.	Naïve Bayes classifier	Bayesian theorem	Harvest ratio	SINA, TOM Web sites
Taylan, Duygu, MitatPoyraz, SelimAkyokuş, and Murat Can Ganiz	Innovations in Intelligent Systems and Applications (INISTA), IEEE (C) 2011	Intelligent Focused Crawler: Learning which Links to Crawl	Link Scoring	SVM	Harvest ratio	Google and Turkish Web sites
Rajesh. L, Shanthi. V	International Journal of Web Technology (J) 2012	A novel approach for Web crawler to classify the Web Documents	Document scoring pertaining to dictionary	Token by token comparison	Precision	Not applicable

J-journal, C-conference

2.1.3: RESEARCH GAPS

In previous literature survey, various focused Web crawling techniques have been discussed which aims to achieve high performance like precision, recall, accuracy, harvest ratio etc. A good focused crawling algorithm should contain various features that lead to high performance. One such feature is classification process to find relevance of the URLs. In order to find the relevancy of the URLs based on the classification, firstly classifier needs to be trained on the pre-defined set of topics and training set of documents, then that trained classifier is used to filter the documents to get relevant documents. Various methods has introduced for classification process like SVM, Decision Tree, Decision Rule, Naïve Bayes classifier etc. SVM classifier has different variations. One of variation is computing the relevancy based on the similarity of the document with the categories created from the set of topics. Naïve bayes classifier uses multinomial or multi vibrator Bayesian theorems in order to match documents with the categories. These methods provide good results but these are not intelligent algorithms.

Various swarm intelligent algorithms like swarm particle algorithm, ant colony algorithm and genetic algorithm have been used to achieve high performance. Combination of various swarm algorithms have also been used in many focused crawling approaches. QProber technique (Gravano, Luis, Panagiotis G. Ipeirotis, and Mehran Sahami. 2003) involves finding form interface, generation of queries and shooting over the simple form interface, retrieve results to get high coverage and high specificity. In summary research gaps are

1. Retrieving and querying Web data that has attracted a lot of attention. In the literature, keyword based interfaces are used for querying the databases that does not require detailed knowledge of database.
2. Querying the database is also proposed for classifying the database. These techniques are more interested in determining the number of matches that each query probe generates rather than inspecting documents retrieved by the queries.
3. As large amount of data is present in Web that can be accessed by focused crawler to crawl webpages related to a particular topic.
4. Techniques in the literature use queries result to generate categories of topic. The query is chosen randomly and the results obtained are not always optimized.

2.2:PROBLEM FORMULATION AND OBJECTIVES

Many focused crawling techniques have been discussed previously in order to achieve relevancy of the documents and coverage of the Web. Main aim of focused Web crawler is to retrieve relevant documents to achieve efficiency and to achieve more coverage.

Usually crawlers retrieve content only from the set of webpages reachable purely by following links, ignoring search forms and interfaces which lead to Web accessible databases containing relevant information. In order to extract content from Web behind the search interfaces, probing technique is used in focused crawler.

2.2.1 Problem statement: For categorization to crawl Web, using probing technique to generate rules for optimized categories generated from genetic algorithm.

2.2.2 Objectives:

1. In order to design a focused Web crawler for web page categorization. Firstly, various focused crawling techniques are studied and compared.
2. Then, design of genetic algorithm based focused crawling to target Web content is proposed. Generation of optimised categories for a cluster using genetic algorithm to get contents searchable Web is proposed.
3. Then the designed focused crawler approach is to be developed on a suitable platform. The platform to be used again depends upon the features of the designed focused crawler that needs to be implemented. After the implementation of the designed focused crawling approach, it will run to obtain results.
4. Then the designed focused crawling approach is verified and validated. The validation is done by analyzing the results obtained after running the designed crawler.

2.3 PROPOSED TECHNIQUE DESIGN

Genetic algorithm is an optimization technique which is based on the concept of hereditary and revolution. Genetic algorithm provides optimal solution to a particular problem based on the fitness value. Fitness of each chromosome is computed and best fitted chromosomes can be used for further processing. In order to classify webpages of indexible Web genetic algorithm based focused Web crawler is proposed. The proposed system consists of indexible Web terminology, URL filtering, feature extraction, genetic algorithm based classifier and classification as shown in figure 3. In this study, our aim is to determine whether webpage have some information of Indian origin faculty webpage working in foreign universities or not i.e. Binary classification is used for class labels. This process starts with crawling indexible Web based on their respective terminology. For indexible Web, process starts with the seed URLs which are scraped and whose out links are extracted for further processing.

Next step starts with URL filtering in which a subset of URLs is selected based on the keywords related to faculty of any university. In feature extraction, certain tags and terms as features are used and extract features using genetic algorithm. In document formation, 2-D array is created to represent the presence or absence of feature in the document. The genetic algorithm based classifier learning part consists of: (i) coding, (ii) generation of initial population. (iii)

Evaluation of initial population, (iv) selection, (v) crossover, (vi) mutation, (vii) generation of new population such that steps (iii) to (vii) are repeated until convergence to learn a (sub) optimal classifier. After the learning process, learned classifier is used for the classification to classify the unseen data. Using best fitted chromosome, a category is selected. Query is updated based on the returned results. The returned documents are analysed, verified and validated using analysis and considerations.

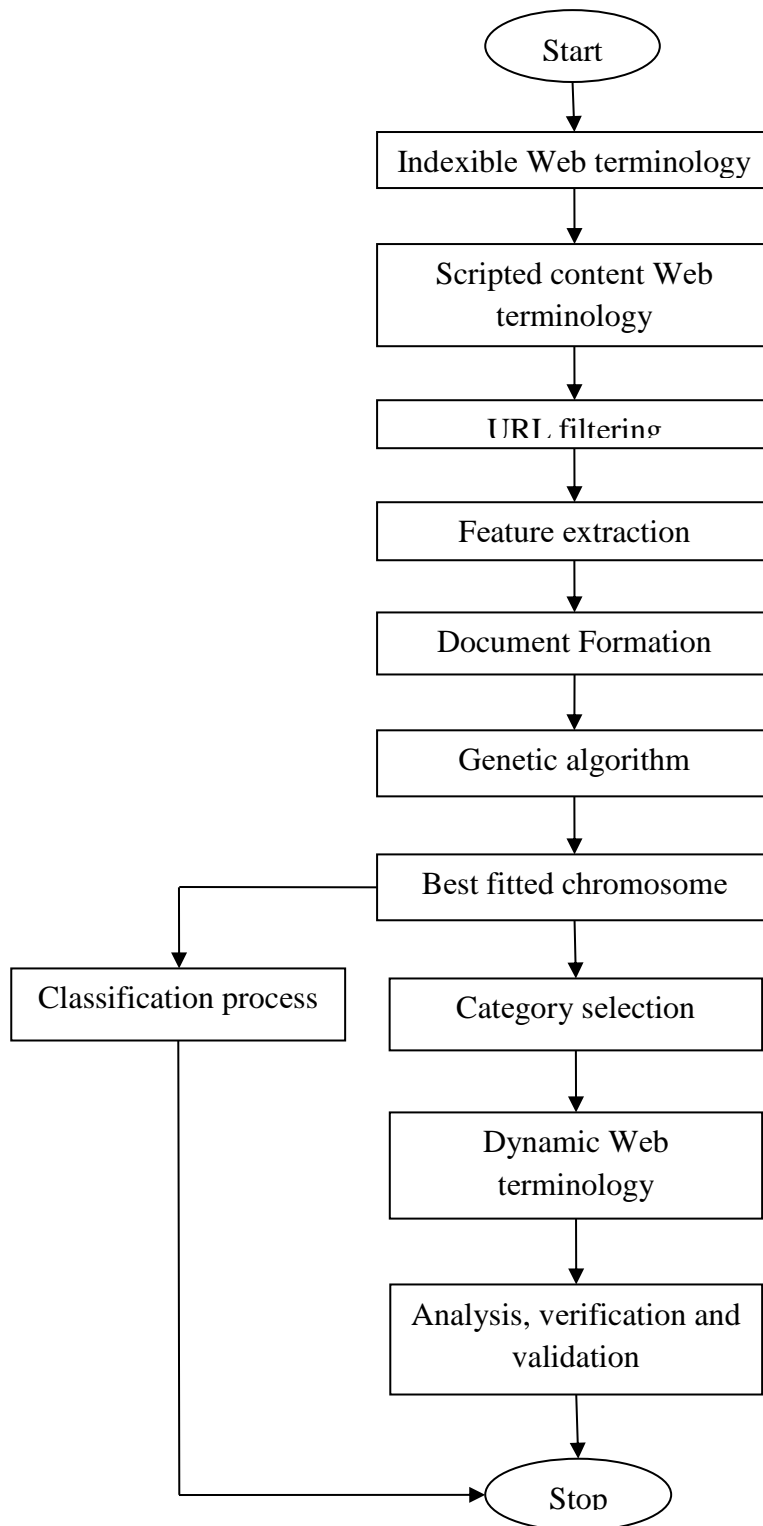


Figure 13 Proposed methodology

2.3.1 Indexible Web Methodology

Indexible Web is the searchable Web which can be reached by following the links present on the webpage. Indexible Web is the Web which donot require login authentication, form

submission. In the proposed terminology for indexible Web, firstly seed URLs which are the URLs of the foreign university websites are taken which are crawled upto certain depth. All the crawled URLs are filtered so that they remain within the domain. There are certain constraints which undergoes while crawling and resolved. Constraints are

1. Some links are bookmarks(containing # in the link).
2. URL might start mailto, file etc.
3. File formats present in the URLs such as pdf, ppt, docx etc.
4. Presence of duplicate URLs on the sites.
5. Links outside the domain like google, facebook, linkedin etc.
6. Presence of certain words in the URLs is not of concern such as alumini, students, calender, events etc.
7. Presence of the words related to the faculty are of concern such as people, staff etc.

2.3.2 URL filtering

The URLs present in the training set are crawled up to certain depth and those crawled URLs are filtered based on the filtering list. Filtering list consist of keywords related to faculty such as people, staff, directory- staff, all-people etc. Since data is unknown, presence of these words might lead us to the URLs containing faculty information.

2.3.3 Feature extraction

Tags such as <title>, <h1>, <h2>, <h3>, <h4>, , , <table>, , <a>, <p>, <meta> which denotes title, header at level 1, header at level 2, header at level 3, header at level 4, image, bold, table, list items, anchor, paragraph respectively are used to extract features that are needed in both classifier learning and classification process.

After analysis and observations, list of Indian surnames, cities, institutes, designations, departments and universities etc are the terms chosen for the above mentioned tags. Feature set is created consisting of these tags and terms. For example <tag-terms> forms one feature in the feature set. For example <title-list of surname>, <table-list of institutes>, <bold-list of departments> etc. <h1>, <h2>, <h3>, <h4> are grouped together to represent one header to reduce number of features extracted.

2.3.5 Document formation

Filtered URLs and extracted features together form documents. Document formation creates a 2-D array consisting of URLs in rows and features in the columns whereas the entries in the array are 0 or 1 depending on the presence or absence of the feature in the document as shown in equation 1.

$$D(i, j) = \begin{cases} 1, & \text{if } T_i \text{ contains } F_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where $D(i, j)$ represent document 2-D array where i represents i^{th} URL from the filtered list T and j represents j^{th} feature from the feature set F . Before creating this 2-D array, stop word removal and stemming using Wordnet is performed on the terms fetched from the URLs present in the filtered list.

2.3.6 Genetic algorithm based classifier:

Genetic algorithm is the optimization process based on the concept of hereditary and evolution. Genetic algorithm is used to select best chromosome among various based on the fitness value. In the proposed methodology, genetic algorithm is trained using training set of URLs, feature set and fitness computation process to get best chromosome. The best chromosome achieved from the training process is used for classification of testing set of URLs.

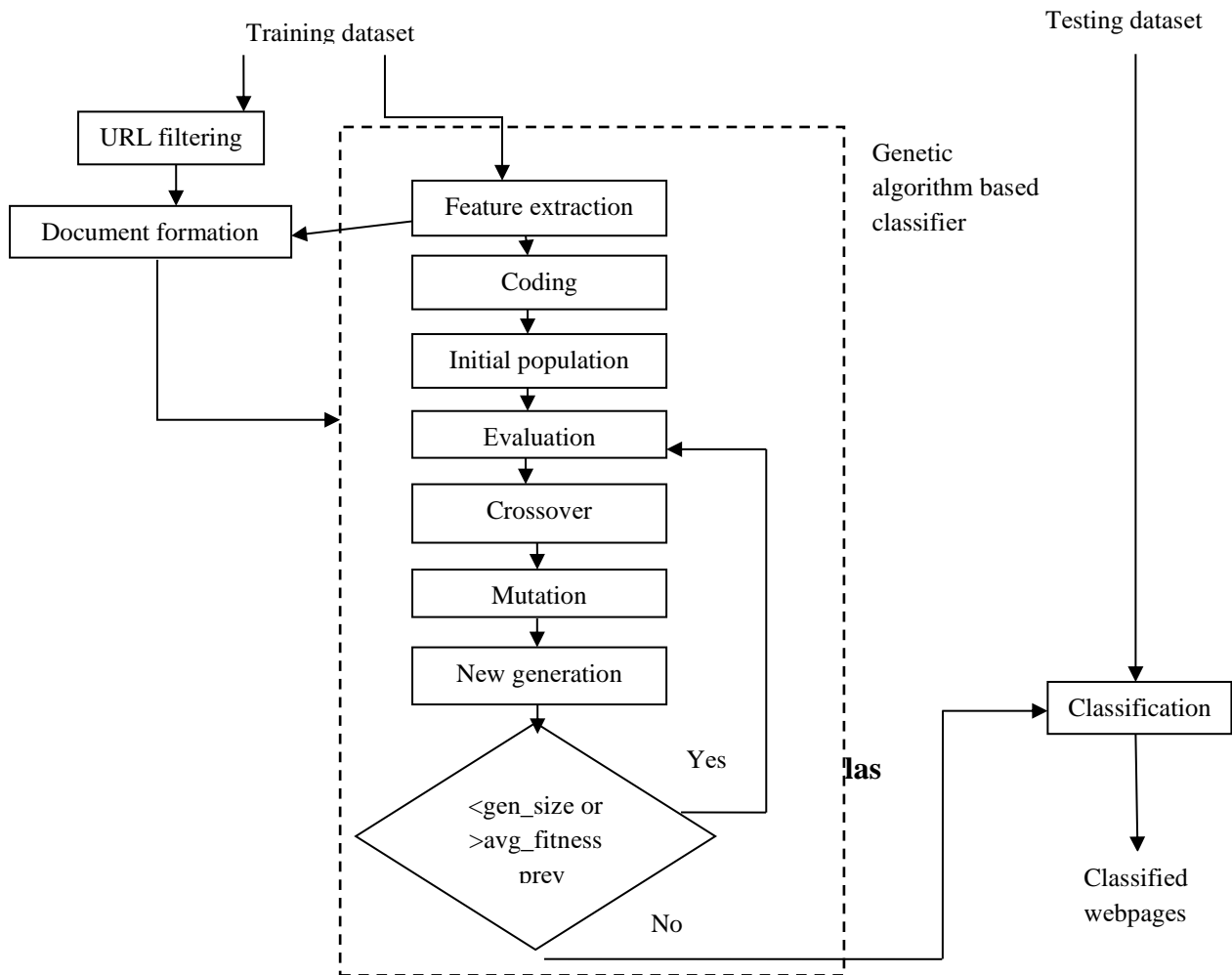


Figure 14 shows the genetic algorithm based classifier process in which based on the feature set, chromosomes are generated randomly and 2-D array of the URLs are formed. Based on the chromosomes and documents formed, fitness of each chromosomes corresponding to all documents is formed. Best fitted two chromosomes are selected for crossover and mutation and generate new chromosomes. Newly generated chromosomes and old chromosomes are sorted and checked for the termination condition. If termination condition met, best chromosome is selected for classification process else iteration process continues.

2.3.6.1 Coding

A chromosome consists of feature weights list which are real numbers in range [0, 1] and is represented in equation 2.

$$W = (w_{11}, w_{12}, \dots, w_{1N_1}, \dots, w_{m1}, \dots, w_{mN_m}) \quad (2)$$

Where W_{ij} denotes the term j in tag i . We used title, header, image, paragraph, table, list, bold, meta and anchor tags in this order. In the proposed work, initial weights are assigned randomly and will be updated in genetic algorithm process.

2.3.6.2 Initial population

Initial population consist of population size chromosomes generated randomly using coding scheme. Size of each chromosome equals to the feature set. Population size taken in the proposed work is 30.

2.3.6.3 Evaluation of population

Fitness of every chromosome present in the population is computed by evaluating the cosine similarity of the chromosome with every document as shown in equation 3. $Cos_simi(C, D_i)$ represents the cosine similarity. After evaluating the cosine similarity, threshold value is taken which is the mean of the cosine similarities for a chromosome corresponding to all documents as shown in equation 4. This threshold value might provide average result but donot decrement the overall performance. And then the average of the cosine similarities of the chromosome corresponding to the documents is computed. That average is the fitness for the chromosome. Fitness computation is as shown in equation 5.

$$Cos_simi(C, D_i) = \frac{\sum_{j=1}^n C[j]*D_i[j]}{\sqrt{\sum_{j=1}^n C[j]*C[j]} + \sqrt{\sum_{j=1}^n D_i[j]*D_i[j]}} \quad (3)$$

$$threshold = \frac{\sum_{i=1}^m Cos_simi(C, D_i)}{m} \quad (4)$$

$$fitness = avg(Cos_simi(C, D_i) \forall i : Cos_simi(C, D_i) > threshold) \quad (5)$$

Where n is the number of elements in the feature set, m is the number of documents present in the training dataset. C represents the chromosome and D_i represents the i^{th} document from the training set.

2.3.6.4 Selection

For the selection of the chromosomes, a novel technique is used in which a dummy chromosome is created as a parameter for selection. Dummy chromosome is created containing elements equals to the average of the corresponding elements of all the chromosomes present in the population as shown in equation 6.

$$C_m[i] = avg(\sum_{j=1}^n C[j][i]) \quad (6)$$

Where $C_m[i]$ represents i^{th} element of the dummy chromosome and $C[j][i]$ represents i^{th} element of the j^{th} chromosome. Then fitness of that dummy chromosome is computed using equation 5. Based on the minimum difference between the fitness of the dummy chromosome and the chromosome of the population, chromosomes are selected for further processing.

2.3.6.5 Crossover

In the proposed approach, uniform crossover technique is used in which a chromosome sized random dummy chromosome is generated which contains random weights. And then that dummy chromosome is compared with the crossover probability as shown in equations 7 and 8.

$$\text{if } (r[i] < P_c) \text{ then } c1[i] = P1[i] \text{ and } c2[i] = P2[i] \quad (7)$$

$$\text{if } (r[i] > P_c) \text{ then } c1[i] = P2[i] \text{ and } c2[i] = P1[i] \quad (8)$$

Where $P1[i]$, $P2[i]$, $r[i]$, $c1[i]$, $c2[i]$ are the i^{th} weight of the feature of the first parent chromosome, second parent chromosome, dummy chromosome, first child and second child respectively. P_c denotes crossover probability. And then fitness of the newly generated children is computed using equation 5.

Table3: Example of crossover operation

	F1	F2	F3	F4	F5	F6	-----	FN
P1	0.75	0.78	0.45	0.77	0.55	0.23	-----	0.56
P2	0.80	0.56	0.67	0.45	0.33	0.66	-----	0.99
r	0.70	0.67	0.59	0.66	0.23	0.79	-----	0.32
C1	0.75	0.78	0.67	0.77	0.55	0.66	-----	0.56
C2	0.80	0.56	0.45	0.45	0.33	0.23	-----	0.99

Consider for example as shown in Table 4, creation of the child chromosomes from the crossover operation. F1, F2 and so on upto FN are the features present in the feature set. P1, P2, r are first parent chromosome, second parent chromosome, dummy chromosome. C1 and C2 are generated after comparing dummy chromosome with the crossover probability using equations 7 and 8. C1 and C2 are newly generated chromosomes.

2.3.6.7 Mutation

A modified mutation technique is proposed in which mut_no is calculated to determine the number of features in the chromosome that has been changed. As shown in equation 9, pop_size represents the size of the population, $P(m)$ represents the mutation probability and $chromosome_size$ represents the number of elements in the chromosome.

$$mut_no = pop_size * P(m) * chromosome_size(9)$$

In this, a dummy chromosome is calculated in which each element is the average of each feature from all chromosomes in the population of the present iteration and computing its fitness. Selection of the chromosome for the mutation is done using minimum difference between the fitness of the dummy chromosome and the i^{th} chromosome from the population as shown in equation 10.

$$C_s = C_i \text{ if } \sum_{i=0}^n (\min(Diff(fitness_d, fitness_i)))(10)$$

Where C_s represents selected chromosome for mutation and C_i represents i^{th} chromosome. $Fitness_d$ and $fitness_i$ represents fitness of dummy and i^{th} chromosome for population respectively, n is the number of chromosomes in a population. After selecting the chromosome, the mut_no features is selected and their weights is updated based on the random number. Figure 6 shows the algorithm to generate new child by using mutation. In this algorithm, C represents the selected chromosome, C_a represents the dummy chromosome and C_m represents newly generated chromosome. Arr is an array to represent the intermediate state to store randomly generated number j , k and i are simple variables. After generating new chromosome, fitness of that chromosome is computed. Fig 15 algorithm for mutation

Input: Selected Chromosome C , Arr , C_a .

Output: Mutated Chromosome C_m .

1. $k = 0$;
2. for $j = 1$ to mut_no
 - a. Generate random number ran between $[0, 1]$.
 - b. Generate randomly a number j between $[1, chromosome_size]$.
 - c. if ($ran < C[j]$)
 $C_m[i] = C[j]$.
 - d. else $C_m[i] = rand(C[j], C_a[i])$.
 - e. $Arr[k++] = j$.
3. end of for loop.
4. for $i = 1$ to $chromosome_size$
 - a. if i present in Arr array continue. b. else $C_m[i] = C[j]$.
5. end of for loop.
6. Return C_m .

2.3.6.7 Generation of new population

All the chromosomes present in the population of the current iteration and newly generated chromosomes from crossover and mutation are sorted based on their fitness and highly fitted pop_size chromosomes are selected for next iteration. Average fitness of the newly generated population is computed.

2.3.6.8 Termination condition

In order to achieve convergence, improved termination condition is used. Convergence conditions are as shown in equation 11.

$$T = \begin{cases} \text{yes, if } (tp > genSz \vee avgFtCrP < avgFtPvP) \\ \text{no,} & \text{otherwise} \end{cases} \quad (11)$$

Where tp represents the total chromosomes present till iteration, $genSz$ represents the maximum number of the chromosomes that can be generated in the system. $avgFtCrP$ and $avgFtPvP$ represent the average fitness of the current and previous population respectively. When the genetic algorithm is terminated, chromosome with the highest fitness is selected from all the chromosomes present and used for classification of the webpages.

2.3.7 Classification

In the classification phase, figure 7 shows the classification process of the proposed algorithm.

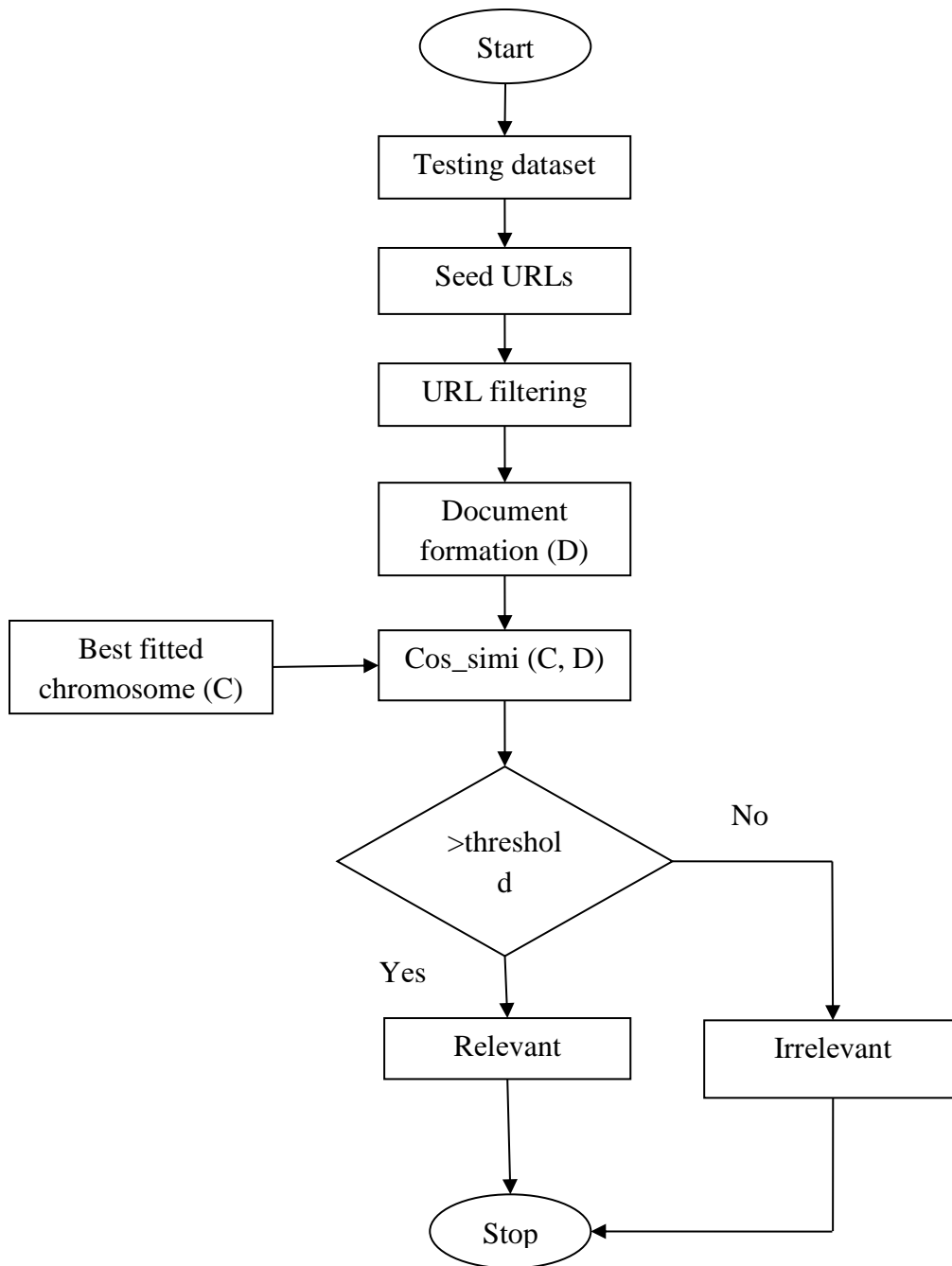


Figure 16 Proposed classification process

In the proposed classification process, seed URLs from the testing dataset are taken to crawl them upto certain depth. Then the URL filtering of the crawled data is performed to filter out the webpages whose URLs do not contain these words present in the filtering list.

Filtered URLs are then passed to document formation phase where they are represented as binary vector of size equal to the number of features taken into consideration. Then the

Cos_simi of the webpage D and best fitted chromosome C is computed. If Cos_simi is greater than threshold, webpage is marked as relevant else irrelevant.

2.3.8 Category selection

Best fitted chromosome achieved after training process of genetic algorithm process is used for the category selection process. Term associated with tag having highest weight is used as a category for the probing technique.

CHAPTER 3

EXPERIMENTAL SETUP

In this chapter, we will describe the following details which are used for setting up the experiment as given below:

1. Hardware and software configuration
2. Dataset
3. Input data
4. Genetic algorithm parameters
5. Platforms and technologies
6. Implementation details

3.1 Hardware and software configuration:

The crawler was implemented under windows 7 operating system. The hardware used in the experiment had 3GB of RAM and Intel Core i3 CPU M 2.53 GHz processor.

3.2 DataSet:

To get Indian origin academician information working abroad, the dataset taken consists of the websites of the foreign universities. The URLs considered in the dataset are as shown in figure 8. Then this dataset of URLs is filtered based on URL filtering list. URL filtering list consist of words such as faculty, directory, people, staff, people-all, directory-people etc. Filtered URLs consists of all those URLs which contains one of these words in the URL itself. These set of webpages consists of irrelevant as well as relevant webpages.

For the dataset, features are extracted based on tags and terms. Tags used are title < t >, header (< h1 >, < h2 >, < h3 >, < h4 >), image < img >, bold < b >, paragraph < p >, table < td >, list < li >, anchor < a >. Terms consist of lists of surnames, institutes, cities, departments and designations. Surnames list containing surnames of the Indians, institutes list consists of educational institutes present in India, and cities list consists of cities of India. Departments list consist of departments present in foreign universities related to science and technology and designations list consist of the designations of the faculties such as professor, assistant professor and associate professor etc. After analysis, feature set is created based on the

combination of tags and terms such as title-designation, title-surnames, header-department, list-cities, table-institutes etc.

1	Url
2	http://www.harvard.edu/
3	http://www.academia.edu/
4	http://video.mit.edu/browse/
5	http://www.stanford.edu/
6	http://www.umd.edu/
7	http://www.princeton.edu/
8	http://dartmouth.edu/
9	http://www.arizona.edu/
10	http://www.berkeley.edu/
11	http://www.uchicago.edu/
12	http://www.columbia.edu/
13	http://www.tufts.edu/
14	http://www.vanderbilt.edu/
15	http://en.wikipedia.org/wiki/AC-262356
16	http://www.bu.edu/
17	http://www.msu.edu/
18	http://www.virginia.edu/
19	http://www.northwestern.edu/
20	https://www.vt.edu/

Figure 17: Seed URLs for indexible Web

For each Filtered URL, document formation takes place in which for the presence or absence of the feature from the feature set, 1 or 0 is marked respectively in 2D document matrix. For example, presence of surname in the title, marks corresponding <title-surname> feature as 1.

3.3 Input data:

In the experiment, list of surnames, cities, institutes, filtering list, feature set are taken as an input data. These are as described below:

3.3.1 List of surnames: List of Indian surnames contains around 5800 surnames. These surnames are used in the feature set as their presence in the URLs as well as in the tags helps to get relevant documents.

3.3.2 List of cities: List of Indian cities contains around 275 cities. This list of cities is used in the feature set as their presence in the tags helps to get relevant documents.

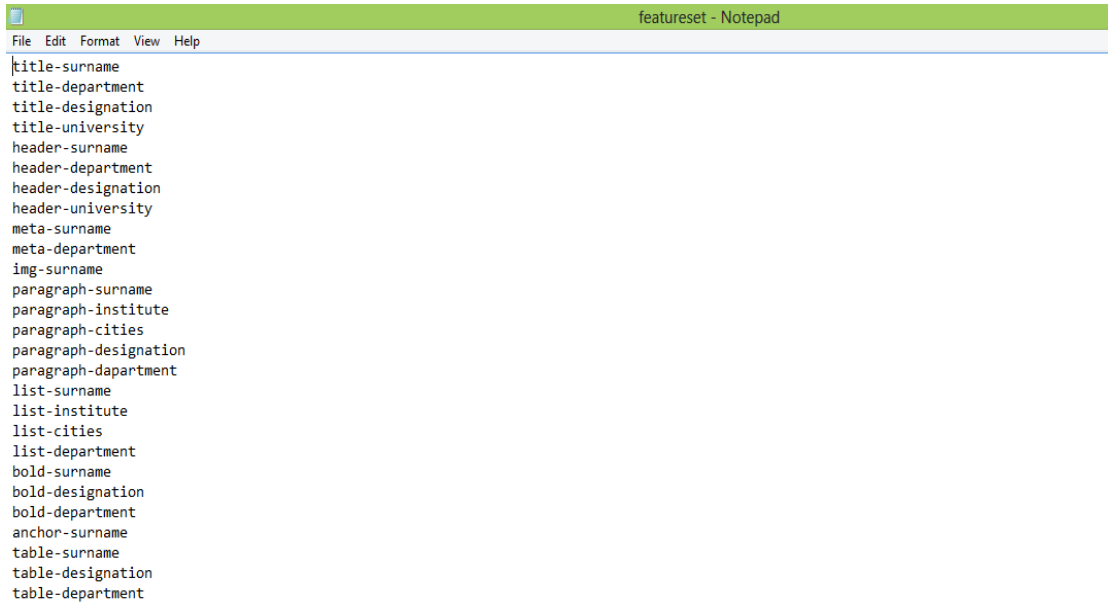
3.3.3 List of institutes: List of institutes contains around 2000 entries. This list of institutes is used in the feature set as their presence in the tags helps to get relevant documents

3.3.4 List of departments: List of departments contains the departments whose faculty is to be searched as Indian faculty.

3.3.5 list of designation: List of designation contains all the designation of the faculty searched as a Indian faculty.

3.3.6 URL filtering list: URL filtering list contains keywords which are used to filter the URLs present in the dataset..

3.3.7 Feature set: Feature set is the list of tags- terms which are used for chromosomes and document formation. Tags such as title, h1 etc and terms as list of surnames, cities etc. Feature set is as shown in figure 16.



```
featureset - Notepad
File Edit Format View Help
title-surname
title-department
title-designation
title-university
header-surname
header-department
header-designation
header-university
meta-surname
meta-department
img-surname
paragraph-surname
paragraph-institute
paragraph-cities
paragraph-designation
paragraph-department
list-surname
list-institute
list-cities
list-department
bold-surname
bold-designation
bold-department
anchor-surname
table-surname
table-designation
table-department
```

Figure18: Feature set

3.4 Genetic algorithm parameters:

Genetic algorithm parameters were determined experimentally such that they were the good choice for our system. Parameters such as population size = 30, generation size = 400, crossover probability = 0.7, mutation probability = 0.5 are taken after analysis and observations.

3.5 Platforms and technologies:

In the experiment, different platforms and technologies are used to implement a crawler to crawl searchable Web. Different platforms and technologies are used as shown below:

3.5.1. JAVA: JAVA is used as a programming language to implement a crawler.

3.5.2. NetBeans IDE:NetBeans IDE is used as a platform to implement crawler. In this experiment, we have used NetBeans IDE 6.9.0.

3.5.3 WordNet library: WordNet library is an open source dictionary project. In this experiment, JWNL (Java WordNet Library) is used for stemming process. Figure 18 shows the wordnet interface in which after shooting any word, we can retrieve all the hyponyms of that words and performs the stemming process.

3.5.4 Jsoup library: Jsoup library is an open source Java Library for webpage parsing. This library is used to parse the webpages.

3.5.5 Oracle: Oracle is the database management system which is used to store tables of documents, chromosomes generated from genetic algorithm process, chromosomes involved in iterations of genetic algorithm process.

3.5.6 Selenium Web driver: Selenium Web driver is a testing tool. General workflow of selenium Test scripts are inserted into Web driver which executes on the Web browser to check the results of the test scripts. In this experiment, selenium Web driver is used to run scripts and finding the form interface and shooting of the query. Selenium Web driver is used to launch a browser as a dummy browser and run as the actual browser does and return the results. The returned results, runs JavaScripts and AJAX. Using this, we have find the Web form interface and shooting the queries and returned results are used for further processing.

3.6 Implementation details:

The crawler is implemented in JAVA programming language. Following are the main points regarding the implementation of the crawler.

1. Firstly, seed URLs are crawled upto depth 5, excluding all the constraints such as URLs outside domain, other types of URLs having extension .doc, .pdf etc.
2. Using URL filtering list and list of surnames, URLs crawled in the first step are filtered since presence of that word from filtering list and presence of surname itself in the URL are useful for further processing. Excluded URLs might not lead us to the faculty information.

3. After analysis and observation, feature set is created in which pair of tag and term constitutes one feature in feature set. Different tags such as title, anchor, h1, h2, h3, h4, bold, table, list etc and terms corresponding to the tags are list of surnames, list of cities etc are taken. Feature in the feature set is like <title, list of surnames>, <a, list of cities> etc.
4. Then, a 2D matrix is generated as document matrix in which rows corresponds to all filtered URLs and columns corresponds to the feature set and entries in the matrix are either 0 or 1 depending on the presence of that feature in particular URL. In document formation stage, firstly documents are downloaded using selenium and JSoup libraries to execute JavaScript and AJAX. Then, stop word removal is done using stop word list containing words such as a, an ,of etc. Then, stemming is performed using Java WordNet library in which after shooting word, it returned stemmed word. For example, if word shot is education, then it will return educate as a stemmed word.
5. Then genetic algorithm stage is performed in which each chromosome consists of weights corresponds to the feature set. Initially, random weights are assigned to each feature in a chromosome and population size number of chromosomes generated. Fitness of each chromosome is computed and two best fitted chromosomes are selected for crossover process. Fitness of each chromosome is computed using cosine similarity. Fitness of a chromosome is equals to average of cosine similarity of chromosome with all the document formed in document matrix. Best fitted chromosome is selected considering both vertical and horizontal computation. A dummy chromosome is created having weights corresponding to the feature is the average of all the weights of all chromosomes corresponds to that feature. Then fitness of that dummy chromosome is computed, chromosomes having minimum difference between fitness of the dummy chromosome and existing chromosome are selected.
6. Multiple crossover process is used based on the crossover probability and parent chromosomes and dummy chromosomes, two children are generated. Then, selecting average fitted chromosomes for the mutation process in which number of weights of the chromosome is changed based on the mutation number. Which weight correspond to which feature is changed, decided randomly but mutation number is created after multiplying mutation probability, population size and size of the chromosome. Then, a new chromosome is generated. Then, all the existing and generated chromosomes are sorted and top fitted population size chromosomes are selected for next iteration and average fitness of that iteration is computed for termination condition. Genetic

algorithm process is terminated when average fitness of current iteration is less than average fitness of previous iteration or total number of chromosomes is greater than generation size. If genetic algorithm terminated, best fitted chromosome is selected for classification

7. Then classification is performed in which testing set of seed URLs are crawled and document formation is done for all filtered URLs. Then, cosine similarity is performed between best fitted chromosome and document matrix's one row and based on the threshold value, each URL is classified as relevant or irrelevant.
8. Web terminology, a category is selected from best fitted chromosome. Category selection is to select highly weighted term from <tag, term> feature from best fitted chromosome. In our proposed approach, category selected is list of surnames which has highest weight in the feature set corresponds to the best fitted chromosome. Then, from the seed URLs of Web, firstly find the presence of searchable form interface. If present, fill the items present such as drop down as author and search bar with surname and shot the query. The returned number of results and results analysed and processed.

3.7 Hurdles faced in implementation

There are various hurdles which are faced while implementing the proposed system. Some of them are stated below.

3.7.1 URL meta information is empty

Extracting keywords based on the meta tag of the URLs undergoes a problem that URL do not contains any information in the meta tag. We cannot be able to obtain information based only on the meta tag. In order to create feature set, we then considered different tags and terms.

3.7.2 Crawling webpage

Following list contains the hurdles while crawling a webpage.

1. Some links are bookmarks(containing # in the link).
2. URL might start mailto, file etc.
3. File formats present in the URLs such as pdf, ppt, docx etc.
4. Presence of duplicate URLs on the sites.
5. Links outside the domain like google, facebook, linkedin etc.
6. Presence of certain words in the URLs is not of concern such as alumini, students, calender, events etc.

7. Presence of the words related to the faculty are of concern such as people, staff etc.

These contains are resolved after analysis and observation.

3.7.3 Crawler sees different from user sees

While parsing the webpage using crawler retrieves different content from what the browser shows to the user. A famous quote “What the user sees is different from what the crawler sees”. Since browser execute everything before showing results to the user while crawler gather only the view page source of the URL without executing anything. Figure 19 (b) shows the content retrieved by the crawler and Figure 19 (a) browser shows the content.



(a) What the user sees



(b) What the crawler sees

Figure 19: Crawler sees different from user sees

3.7.4 Information in image form

While crawling a URL, crawler was not be able to find information because the information present on the webpage was in the image format as well as the data appeared on the webpage

is at run time. Figure 20 shows the information on the webpage which is generated at run time.



Figure 20: Information generated at run time

3.7.5 Finding form interface

finding a searchable form interface and neglecting login and registration form interface is a difficult task. It requires analysis for the form interface and find which labels are present, which input tag is present for the form interface.

3.7.6 Irrelevant links containing relevant useful links

The focused crawling strategy based on the intuition that relevant pages often contain relevant links. It searches deeper when relevant pages are found, and stops searching at pages not as relevant to the topic. Unfortunately, this traditional method of focused crawling shows an important drawback when the pages about a topic are not directly connected.

CHAPTER 4

RESULTS AND DESCUSSION

Focused Web crawler using genetic algorithm to extract content from indexible Web is implemented and results are analysed based on the recall and coverage. Recall in information retrieval is defined as fraction of documents that are relevant to the query that are successfully retrieved.

$$Recall = \frac{\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}}{\{\text{retrieved documents}\}} \quad (12)$$

In this chapter, we will discuss about the training process of the genetic algorithm in which we show how many iterations computed to achieve convergence, what are their average fitness and which best fitted chromosome achieved. Then, describes how many URLs achieved at each stage for different universities such as Stanford, Lancaster, Harvard, Columbia etc. Then, describes the precision achieved for each university.

4.1 Genetic algorithm results

In the training process of the genetic algorithm, genetic algorithm is trained by generating new chromosomes until convergence is achieved. For the training of the chromosomes, Stanford university website URLs is used as documents. Figure 23 shows the document formed in terms of feature set. In this proposed system for genetic algorithm, 38 iterations achieved to reach convergence. Figure 24 shows the average fitness achieved.

	A	B	C	D	E	F	G
33	32	http://stanfordcareers.stanford.edu/discover-stanford/people/lourdes-andrade	0,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
34	33	http://stanfordcareers.stanford.edu/discover-stanford/people/scott-stocker	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
35	34	http://stanfordcareers.stanford.edu/discover-stanford/people/david-cuffy	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
36	35	http://stanfordcareers.stanford.edu/discover-stanford/people/scott-calvert	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
37	36	http://stanfordcareers.stanford.edu/discover-stanford/people/christopher-bennett	0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
38	37	http://stanfordcareers.stanford.edu/discover-stanford/people/israel-magallon	0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
39	38	http://stanfordcareers.stanford.edu/discover-stanford/people/swati-prabhu	1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
40	39	http://stanfordcareers.stanford.edu/discover-stanford/people/vanessa-alcantar	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
41	40	http://stanfordcareers.stanford.edu/discover-stanford/people/theo-mitchell	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
42	41	http://stanfordcareers.stanford.edu/discover-stanford/people/jo-ann-cuevas	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
43	42	http://stanfordcareers.stanford.edu/discover-stanford/people/dave-bunger	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
44	43	http://stanfordcareers.stanford.edu/discover-stanford/people/miguel-hernandez	0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
45	44	http://stanfordcareers.stanford.edu/discover-stanford/people/maria-eugenia-smith	0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,0,				
46	45	http://stanfordcareers.stanford.edu/discover-stanford/people/cindy-cho	0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
47	46	http://web.stanford.edu/group/SUDPS/crime-alert1516.shtml	0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,				
48	47	http://www.stanford.edu/group/parentsclub	0,1,0,0,0,				
49	48	http://www.stanford.edu/group/parentsclub/	0,1,0,0,0,				
50	49	http://web.stanford.edu/group/parentsclub/	0,1,0,0,0,				
51	50	http://www.stanford.edu/group/SUDPS/	0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,				
52	51	http://www.stanford.edu/group/SUDPS/employment.shtml	0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,0,1,0,0,1,				
53	52	http://web.stanford.edu/group/fms/fingate/contact/about_FMS.html	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,0,0,0,				
54	53	https://alumni-gsb.stanford.edu/get/page/directory/search?pgorg=bsa	0,1,0,0,0,				
55	54	https://alumni-esc.stanford.edu/get/page/directory/search	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,				
56	55	http://www.gsb.stanford.edu/faculty-research/faculty/peter-m-demarzo	0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,				
57	56	http://www.gsb.stanford.edu/faculty-research/faculty/dan-m-klein	0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,0,0,0,1,0,0,0,1,0,0,0,				

Figure 21: Document formation in 2D matrix

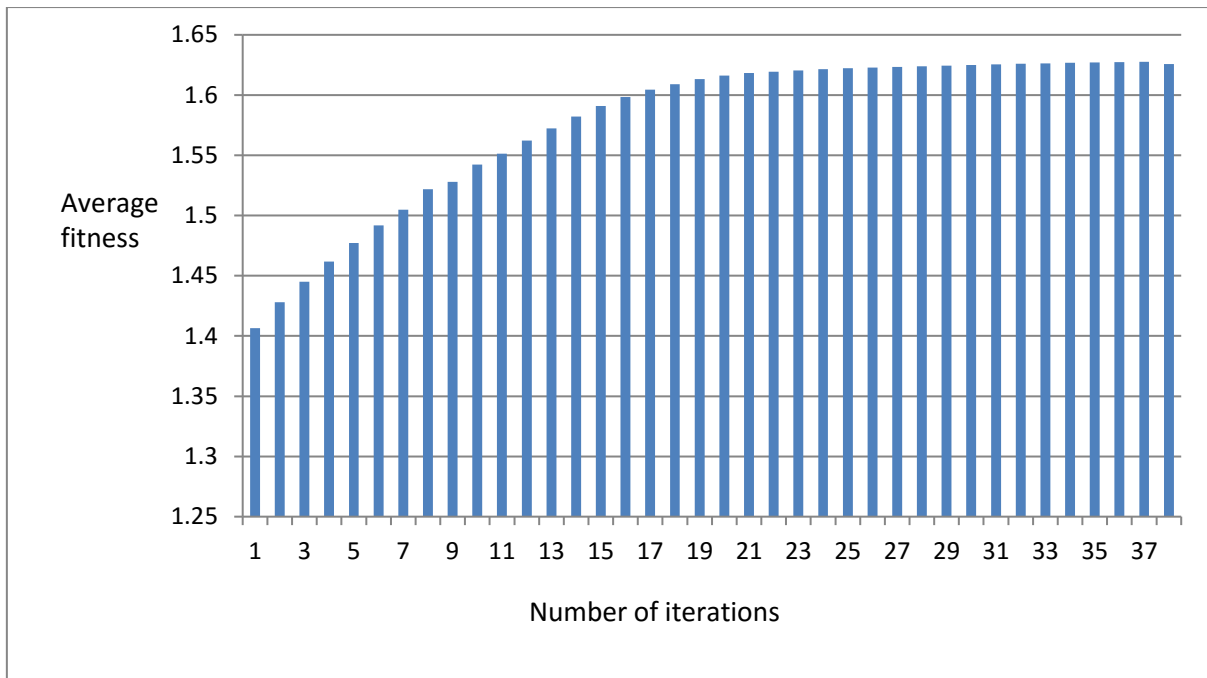


Figure 22: Average fitness in each iteration

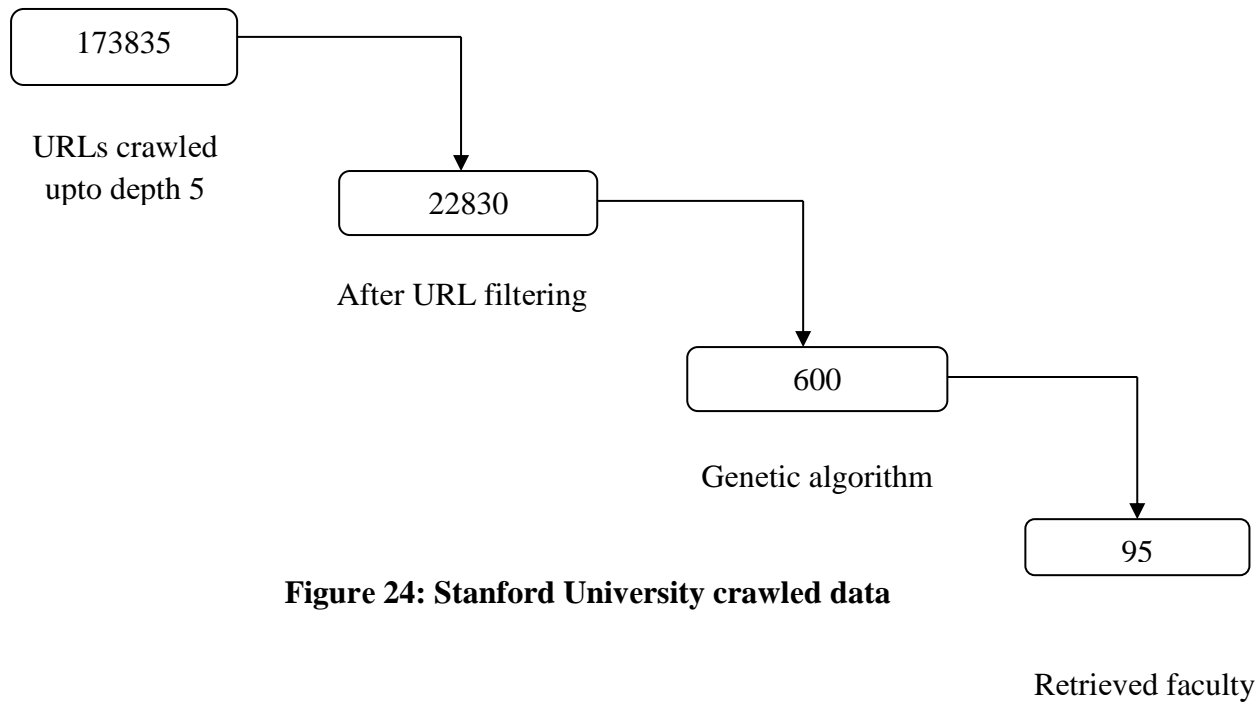
After reaching convergence, best fitted chromosome is extracted which has best average fitness.

	A	B
1	feature set	weights
2	title-surname	0.83
3	title-department	0.8
4	title-designation	0.77
5	title-university	0.9
6	header-surname	0.83
7	header-department	0.78
8	header-designation	0.84
9	header-university	0.85
10	meta-surname	0.2
11	meta-department	0.23
12	img-surname	0.57
13	paragraph-surname	0.34
14	paragraph-institute	0.36
15	paragraph-cities	0.27
16	paragraph-designation	0.31
17	paragraph-dapartment	0.39
18	list-surname	0.34
19	list-institute	0.32
20	list-cities	0.2
21	list-department	0.21
22	bold-surname	0.76
23	bold-designation	0.78
24	bold-department	0.79
25	anchor-surname	0.76
26	table-surname	0.49
27	table-designation	0.62
28	table-department	0.6

Figure 23: Best fitted chromosome with weights

4.2 Crawled data for different universities

The proposed crawler is crawled on various foreign universities website and crawled data is verified and analysed. Following figures 26 shows the number of URLs achieved at each stage of the proposed system.



4.3 Recall computation for crawled data

Precision is the performance parameter as fraction of documents that are relevant to the query that are successfully retrieved. Recall achieved for the Stanford University seed URLs is 91.98%.

4.4 Comparison with other techniques

The results of our purposed technique have been compared with the GA based tag and terms classification technique[20] and Naïve Bayes classifier[21] .We got good results using GA due to the fact that there is very much dependency in the problem which is analogous to many real world scenarios of same size. In the comparison between proposed GA and Naïve Bayes classifier, former performs much better when the problem size is large and we have very less specific knowledge . But naïve bayes can be good when attribute dependency is very less.

Comparison between Proposed GA and Naïve Bayes is in table IV.

Table IV: Accuracy Comparison between Naïve Bayes classifier and Proposed technique.

S.No.	Technique	No of records	% Accuracy
1.	Naïve Bayes classifier	12960	49.32
2.	GA based classifier (proposed)	173835	91.98

Comparing our proposed technique with existing other GA based classifiers[20] , which used terms and tags for its working our technique is better as it can deal with a very large size problem, as in terms of accuracy proposed technique is better results are shown in table V.

Table V: Percentage Accuracy for Proposed Technique and existing GA technique

S.NO	Technique	%Accuracy
1.	GA based Focused crawler approach(proposed)	91.98
2.	GA based Tag and Terms approach	89

But tag and term based GA approach [20] is not suitable for large problem set but it can be solved using proposed approach.

CHAPTER 5:

CONCLUSION AND FUTURE SCOPE

Focused Web crawler is the Web crawler which gathers, extracts documents based on pre-defined set of topics. Focused Web crawler saves hardware and software resources such as time consumption, memory wastage etc. But focused Web crawler is unable to target hidden Web content. Since hidden Web content is reachable after submitting form interface or the content which is generated at run time, focused Web crawler is not able to extract content by simple processing. Focused Web crawler requires special mechanism to achieve this. In this study, focused Web crawler is implemented using genetic algorithm to categorize content from Web. In this proposed system, genetic algorithm is trained using searchable Web URLs to get best fitted chromosome. Chromosome composed of feature set with associated weights based on tags and terms associated with the tags. Searchable Web URLs are filtered before using in genetic algorithm process based on the filtering list. Best fitted chromosome is used for the classification of the indexable Web to retrieve relevant or irrelevant documents and for category selection as well.

Recall is the performance parameter used for verification and validation. In the proposed study, recall achieved is 91.98% when applied on the crawled data of Stanford University.

For future work, we plan to increase the precision and coverage of the proposed crawler so that all the relevant documents can be retrieved within the domain successfully. Also, while implementing the proposed system, various hurdles were faced as mentioned in chapter 6. In order to overcome these hurdles, different mechanism can be implemented. Also what is the performance of the proposed crawler when higher number of relevant pages is retrieved.

REFERENCES

- [1] Barbosa, Luciano, and Juliana Freire. "Siphoning Hidden-Web Data through Keyword-Based Interfaces." *SBBD*. 2004.
- [2] Belmouhcine, Abdelbadie, and Mohammed Benkhalifa. "Implicit Links based Web Page Representation for Web Page Classification." *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*. ACM, 2015.
- [3] Chakrabarti, Soumen, Martin Van den Berg, and Byron Dom. "Focused crawling: a new approach to topic-specific Web resource discovery." *Computer Networks* 31.11 (1999): 1623-1640.
- [4] Chakrabarti, Soumen, Byron Edward Dom, and Martin Henk van den Berg. "System and method for focussed web crawling." U.S. Patent No. 6,418,433. 9 Jul. 2002.
- [5] Deshmukh, Mr Ankit R., and Sunil R. Gupta. "Data mining based soft computing methods for web intelligence." *Methods* 3.3 (2014). Gravano, Luis, Panagiotis G. Ipeirotis, and Mehran Sahami. "QProber: A system for automatic classification of hidden-Web databases." *ACM Transactions on Information Systems (TOIS)* 21.1 (2003): 1-41.
- [6] Gupta, Sonali, and Komal Kumar Bhatia. "A Comparative Study of Hidden Web Crawlers." *arXiv preprint arXiv:1407.5732* (2014).
- [7] Kausar, Md Abu, V. S. Dhaka, and Sanjeev Kumar Singh. "Web crawler: A review." *International Journal of Computer Applications* 63.2 (2013): 31-36.
- [8] Korde, Vandana, and C. Namrata Mahender. "Text classification and classifiers: A survey." *International Journal of Artificial Intelligence & Applications (IJAIA)* 3.2 (2012): 85-99.
- [9] Li, Jun, Kazutaka Furuse, and Kazunori Yamaguchi. "Focused crawling by exploiting anchor text using decision tree." *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 2005.
- [10] Liakos, Panagiotis, et al. "Focused crawling for the hidden Web." *World Wide Web* (2015): 1-27.
- [11] Luong, Hiep Phuc, Susan Gauch, and Qiang Wang. "Ontology-based focused crawling." *Information, Process, and Knowledge Management, 2009.eKNOW'09. International Conference on*. IEEE, 2009.

- [12] Marin-Castro, Heidy M., et al. "Automatic discovery of Web Query Interfaces using machine learning techniques." *Journal of Intelligent Information Systems* 40.1 (2013): 85-108.
- [13] Özel, Selma Ayşe. "A Web page classification system based on a genetic algorithm using tagged-terms as features." *Expert Systems with Applications* 38.4 (2011): 3407-3415.
- [14] Raghavan, Sriram, and Hector Garcia-Molina. "Crawling the hidden Web." (2000).
- [15] Singh, Chain, Ashish Kr Luhach, and Amitesh Kumar. "Improving Focused Crawling With Genetic Algorithms." *International Journal of Computer Applications* 66.4 (2013): 40-43.
- [16] Taylan, Duygu, et al. "Intelligent focused crawler: learning which links to crawl." *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*. IEEE, 2011.
- [17] Top 50 open source web crawlers for data mining. September 8, 2015. Available from: "http://bigdata-madesimple.com/top-50-open-source-web-crawlers-for-data-mining/"
- [18] Udapure, Trupti V., Kale R. D., and Dharmik R. C. (2014). "Study of Web Crawler and its Different Types." *IOSR Journal of Computer Engineering* 16.1.
- [19] Vieira, Karane, et al. "Siphon++: a hidden-Webcrawler for keyword-based interfaces." *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008.
- [20] Selma Ayşe Özel "A Web page classification system based on a genetic algorithm using tagged-terms as features" Selma Ayşe Özel , Science Direct Expert Systems with Applications (2011)
- [21] Keshavamurthy "Privacy Preservation Naïve Bayes Classification for a Vertically Distribution Scenario using Trusted Third Party" Keshavamurthy ,Department of Electronics and Computer Engineering, IIT Roorkee 2010 International Conference on Advances in Recent Technologies in Communication and Computing.