Multimodal, Web Page Modeling for Content Scoring based on Segmentation and Incremental Profile Amalgamation

A THESIS

Submitted By

K.S.Kuppusamy

In partial fulfillment of the requirements for the award of the degree

of

Doctor of Philosophy

In

Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF ENGINEERING AND TECHNOLOGY
PONDICHERRY UNIVERSITY
PUDUCHERRY – 605014

DECEMBER 2012

CERTIFICATE

This is to certify that this thesis titled "**Multimodal, Web Page Modeling for Content Scoring based on Segmentation and Incremental Profile Amalgamation"** submitted by **Mr. K.S.Kuppusamy**, to the Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India for the award of the degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of bonafide research work carried out by him under my guidance and supervision.

This work is original and has not been submitted, in part or full to this or any other University / Institution for the award of any other degree.

Place : Puducherry
Date :

Prof. G. Aghila., M.E., Ph.D
(Supervisor)
Dept. of Computer Science,
School of Engineering and
Technology,
Pondicherry University,
Puducherry – 605014
India.

DECLARATION

I hereby declare that this thesis titled "**Multimodal, Web Page Modeling for Content Scoring based on Segmentation and Incremental Profile Amalgamation"** submitted to the Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India for the award of the degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of bonafide research work carried out by me under the guidance and supervision of **Prof. G. Aghila.** This work is original and has not been submitted, in part or full to this or any other University / Institution for the award of any other degree.

Place : Puducherry **K.S.Kuppusamy**

Date :

ACKNOWLEDGEMENT

I would like to express my heartiest gratitude to my guide and mentor **Dr. G. Aghila**, Professor, Department of Computer Science, School of Engineering and Technology, Pondicherry University for her precious guidance, exquisite suggestions and making me to take correct decisions at important crossroads during the course of this research work. The perfect blend of caring approach with the strong academic foresight of my guide has provided me, the constant support throughout this research.

I would like to express my sincere gratitude to my doctoral committee members **Dr. N. Sreenath**, Professor, Department of Computer Science and Engineering, Pondicherry Engineering College and **Dr. T. Chitralekha**, Reader, Department of Computer Science, School of Engineering and Technology, Pondicherry University for their thought-provoking and innovative suggestions, which fueled to the enhancement of this research work.

I express my sincere thanks to **Dr. P. Dhavachelvan**, Head, Department of Computer Science, School of Engineering and Technology, Pondicherry University for permitting me to do this research work and his relentless support. I am immensely grateful to **Dr. V. Prithiviraj**, Dean, School of Engineering and Technology, Pondicherry University for his motivational suggestions.

Heartfelt thanks to my co-researchers at the Department of Computer Science, Pondicherry University, **Ms. R. Sunitha, Ms. V. Uma, Ms. SanthiBala** and **Mr. P.Thiyagarajan,** for rendering their support and triggering me to explore the novel approaches which improved this research work greatly. I express my warm regards to the teaching, technical and non-teaching staffs of the Department of Computer Science for their timely help.

I am deeply indebted to my parents **Mr. K.V. Srinivasan** and **Ms. K.S.Nagalammal,** without whom I wouldn’t have become whatever I am today. I express special thanks to my younger brother **Mr. K.S. Jenarthanan,** for his affection and timely support throughout my life. I express my thanks to my wife **Ms. Leena,** for her love and affection which made me to focus on my research work.

K.S.Kuppusamy

ABSTRACT

This thesis is aimed at web content scoring by modeling the web pages as a collection of segments which are evaluated using a multimodal approach. As the web pages are constantly evolving into diversified content sources which hold various topics at different sections, treating the whole web page as an atomic unit would lead to the ignoring of the structural semantics exposed by them. In order to address this problem, this thesis has proposed a model termed SCOPAS (Semantic Computation of Page Score) which fine-grains the evaluation of a web page through segmentation, classification and personalization. A variable magnitude approach is incorporated into the evaluation process by assigning different weightage to the web page segments using various structural semantics.

A hybrid web page segmentation technique has been proposed in the SCOPAS model which utilizes the page-tree and densitometry to mark the segment boundaries. The resultant segments of the web page are evaluated through content scoring process in a multimodal manner, with the help of six different weight coefficients proposed in this research work. In order to further enrich the content scoring process, the segments were given class specific weights based on the classification of segments which is carried out with the help of intra-segment features. This thesis has proposed a web page segment classification technique, ClaPS (Classification of Page Segments) which classifies the segments using a multi-class, single label, hard classification approach with the help of decision trees.

As the informational needs of users are spanning across a spectrum, the model encompasses personalization component which builds the profile-bag of the user using a hybrid profile data collection approach and an incremental profile amalgamation technique. The initial profile bag of the user is constructed using an explicit profile data collection process. The profile-bag of the user is incrementally updated through four actions monitoring parameters viz. bookmarking, time-threshold, persistence and hard copying.

The evaluation of the proposed SCOPAS model was carried by analyzing the results for individual components. The individual components of the model are evaluated using specific metrics like NMI (Normalized Mutual Information), Kappa Statistics, True Positive Rate (TPR) and NDCG (Normalized Discounted Cumulative Gain) etc. The results of the experiments conducted provide the empirical evidence to the concepts proposed in the model.

The SCOPAS model is utilized in domain specific realizations which are adaptation of the SCOPAS model to achieve applications specific tasks like re-ranking the search engine’s result listing (SCOPAS-Rank), change detection in various versions of the same web page (CaSePer - Change detection using Segmentation and Personalization), rendering of web pages in small screen devices (MORPES – Mobile Rendering of Pages using Evaluation of Segments) and web page summarization based on segmentation. The above mentioned domain specific realizations are also tested with various metrics like MSFS (Mean of Segments in First Shot), MPSC (Mean of Page shot Count), LMP (Links in Mean Page) which revealed encouraging results.

The conclusions derived out of this thesis are that the content scoring process can be enriched by fine-graining the evaluation process and incorporating a multimodal approach with the help of weight coefficients. The introduction of segment classification process into the scoring facilitates a variable magnitude approach. The incremental profile amalgamation component addresses the challenges of incorporating user-specific information requirement context.

The future directions of this research work includes development of specialized parsers for handling various content types, introduction of image analysis techniques to extract semantics from the images and incorporation of Natural Language Processing techniques to further enrich the scoring mechanism.

TABLE OF CONTENTS

3 The SCOPAS – Segmentor

4 The ClaPS – Segment Classifier

5 The SCOPAS - Evaluator

6 The SCOPAS Profiler

7 SCOPAS Experiments and Model Realization

8 Conclusions and Future Directions

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Expansion**
ARFF	Attribute Relation File Format
ARI	Adjusted Rand Index
CaSePer	Change Detection using Segmentation and Personalization
CIC	Completely Irrelevant Count
ClaPS	Classification of Page Segments
CLS	Concept Learning System
CRC	Completely Relevant Count
CSF	Coefficient Strength Factor
DCG	Discounted Cumulative Gain
DoC	Degree of Coherence
DOM	Document Object Model
FOAF	Friend Of A Friend
FPR	False Positive Rate
FV	Feature Value
GUI	Graphical User Interface
ID3	Iterative Dichotomiser
IDCG	Ideal Discounted Cumulative Gain
IDF	Inverse Document Frequency
IMP	Images in Micro-Page
IR	Information Retrieval
ISF	Inverse Segment Frequency
ISP	Images in Source Page
k-NN	k- Nearest Neighbor
LMP	Links in Micro-Page
LSP	Links in Source Page
MAP	Mean Average Precision
MORPES	Mobile Rendering of Pages using Segment Evaluation
MPETL	Mean Page Evolution Track Length
MPSC	Mean of Page Shot Count

MSC	Mean Segment Count
MSFS	Mean of Segments in First Shot
MSM	Mean Segments Modified
MSMP	Mean Segments Modified with Personalized keywords
MSUD	Mean Segments Undetected
MTIT	Mean of Terms from Item Interest
MTWB	Mean of Terms from Weblog
MTWH	Mean of Terms from Workplace Homepage
MUSEUM	Multidimensional Segment Evaluation Method
NDCG	Normalized Discounted Cumulative Gain
NMI	Normalized Mutual Information
ODP	Open Directory Project
PRC	Partially Relevant Count
RIF	Revisit Interval Flag
ROC	Receiver Operating Characteristics
SCOPAS	Semantic Computation of Page Score
SMP	Segments in Micro-Page
SSP	Segments in Source Page
SVM	Support Vector Machine
TF	Term Frequency
TPR	True Positive Rate
URI	Uniform Resource Identifier
VIPS	Vision based Page Segmentation
WWW	World Wide Web
XML	Extensible Markup Language

Chapter I: Introduction

This work titled “Multimodal, Web Page Modeling for Content Scoring based on Segmentation and Incremental Profile Amalgamation” addresses the challenges in evaluating the relevance of web pages with respect to the user’s information requirement context. This work approaches the content scoring problem by modeling the web pages with the help of segmentation, which functions as a fine-grained control mechanism. The scoring process includes a segment classification technique which facilitates variable magnitude approach. In addition to this, personalization is also incorporated, as the requirement of information varies drastically across users, influenced by various local and global parameters.

1.1 The Web Ecosystem

The World Wide Web (WWW) has become the largest repository of information at the global scale. The World Wide Web has evolved into a colossal content source which holds information on almost all the fields known to humanity. The World Wide Web is exploding at a very fast rate and the size of the indexed web is estimated as atleast 8 billon pages as on October, 2012.[1] Human knowledge is evolving every day, so is the World Wide Web. The contents of the World Wide Web exhibit a greater level of dynamism. The World Wide Web has evolved from a simple and static content source into a richly dynamic information delivery channel.

Adding value to this massive size is the openness of access to information resources available in the World Wide Web. This combinatorial power of both size and openness has made the World Wide Web, a single-stop solution for the informational needs of billions of users across the spectrum. Another key contributor to this web-phenomenon is the ease with which the contents can be accessed. The facility of navigating the resources of World Wide Web without the need for learning any special syntax has fueled the web growth to an unprecedented level of success.

The content navigation through World Wide Web has been made further efficient with the help of web search engines. Without the evolution of web search engines the colossal size of the web which is now referred as the power would have become a bottleneck, in the information access procedure. Each web search engine employs its

[1] http://www.worldwidewebsize.com/

own unique procedure to decide the relevancy of a web page. For each web page a relevancy score is computed with respect to the query entered by the user. The scoring of web pages for efficient retrieval is explored in the following section.

1.2 Scoring of Web Contents

The scoring of contents with respect to the user's requirement continues to be a sparkling research area since past few decades, primarily fueled by the Knowledge Economy. Systematic access to relevant information was proposed as early as in 1945, in a pioneering article "As We May Think" by Vannevar Bush in which he has foreseen the modern retrieval eco-system that prevails today, during an era when the computers were in their infancy and internet was never heard-of (Bush, 1945).

The creative hypothesis proposed by Vannevar Bush was ahead of his time. In 1957, H.P Luhn proposed a statistical approach to mechanized encoding and searching of literary information which laid the foundation for many of the modern information retrieval approaches (Luhn, 1957). Although the studies on Information Retrieval began more than half a century ago, the field got into lime-light, once the World Wide Web became much popular among the masses and it captured the place of a de-facto information delivery channel.

As indicated by Manning and Raghavan in their book "Introduction to Information Retrieval", the term Information Retrieval can be defined as follows (Manning et al., 2008):*"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."*

Retrieval of relevant information is the cornerstone of this Information era. The World Wide Web is the colossal horizon for this retrieval which is mammoth sized, unstructured in organization, linked in relation and dynamic in temporal dimension. Convoluting this scenario is the varying context of this retrieval which demands systematic modeling of web pages with specific measures incorporating the individual users' requirements. As the web has evolved from the simple collection of static hyperlinked documents into a richly dynamic information delivery medium, the evaluation of its contents need to be carried out with a variable magnitude approach.

The Information Retrieval study encompasses retrieval of relevant data, which may be from a personal computer, an organization's network or the global area network – The World Wide Web. The challenges faced in each of these environments are specific to their size, architecture and access models. This thesis focuses on the web page scoring as its primary objective. The web page scoring is the process of assigning a relevance score to a web page, with respect to the user's information requirement. (Brin and Page, 1998a; Haveliwala, 2003; Li et al., 2002). As the web pages are different from the traditional text documents in terms of their linked nature and inherent structural attributes, the scoring of web pages shall be enriched with the help of segmentation, classification and personalization.

Web page segmentation is the process of splitting a web page into smaller components, based on various parameters like structural characteristics, content characteristics. Web page segmentation has been studied in detail by various researchers (Cai et al., 2003; Cao et al., 2010; Chakrabarti et al., 2008; Kohlschütter and Nejdl, 2008; Liu et al., 2011; Nguyen et al., 2012; Vineel, 2009; Yesilada, 2011). The web page segmentation has been studied for a spectrum of objectives which includes information retrieval (Xie et al., 2005), rendering of web pages in smaller screen displays (Hattori et al., 2007; Kang et al., 2010), annotation of contents (Mukherjee et al., 2003), duplicate detection (Kohlschütter and Nejdl, 2008) and rendering of web pages for visually challenged users (Mahmud et al., 2007a, 2007b). The web page segmentation has been attempted using various techniques like page trees, vision based approach, densitometry based approaches etc. This work proposes a hybrid web page segmentation technique incorporating page tree and densitometry for the purpose of fine-graining the page score computation process from the page level to the intra-page segment level, which is explored in detail in Chapter III.

As the World Wide Web is diversified in nature, in terms of the content and structure, the classification of its contents has been carried out using various approaches (Asirvatham and Ravi, 2001; Castillo et al., 2007; Qi and Davison, 2009; Tsoumakas and Katakis, 2007). The classification of web contents has been studied using various techniques like Naïve Bayes (Lewis, 1998), k-NN classification (Kwon et al., 1999; Lam and Ho, 1998; Larkey and Croft, 1996), support vector machine (Sebastiani, 2002; Sun et al., 2002; Vapnik, 1999), decision trees (Apte et al., 1998). In addition to the hybrid web page segmentation technique, this thesis incorporates a web page segment classification approach using decision trees for assigning variable weights to different class of segments, while computing the score of web pages.

The score of a web page is dynamic for different users. The user specific score computation of web pages has been attempted by incorporating “user profiles”. The profile data is gathered either explicitly (Pazzani et al., 1996; Shavlik and Eliassi-Rad, 1998) or implicitly (Fox et al., 2005; Kelly and Teevan, 2003; Liu et al., 2002). The user profiles shall be maintained using keywords (Chen and Sycara, 1998; Moukas, 1997) or concepts (Gauch et al., 2007; Pretschner and Gauch, 1999). In this work, the inclusion of user profiles facilitates personalized scoring of web pages which is specific to the user’s current information requirement context.

This thesis is aimed at scoring of web contents by modeling the web pages as a collection of segments. The statement of the research problem and the specific research questions answered by this thesis are presented in the following section.

1.3 The Problem Statement

With the massive explosion of contents in the World Wide Web, the major issue now, is the “Information Overload” which makes the users to struggle in locating the relevant information resources. The research problem focused by this thesis can be stated as follows:

> *The Web Scale content scoring techniques, handles the whole page as an atomic unit which leads to un-optimized results, as the web pages have become diversified in their contents and its dynamism. The uniform treatment of intra-page components during the content scoring leads to ignoring of various structural semantics exposed by them. Moreover, the constantly evolving user specific information requirement context cast the scoring procedure a dynamic one.*

This thesis explores the above specified problem statement with the help of following research questions:

Question 1: How to fine-grain the content scoring procedure beyond the page level?

> As stated in the research problem, the web pages hold diversified contents in different sections. Hence considering the complete web page as an atomic unit

provides only a limited insight. This thesis presents a solution to this problem through a hybrid web page segmentation technique incorporating page trees and densitometry.

Question 2: How to incorporate a differential approach in content scoring for various intra-page components?

The web page segments exhibit structural diversity in terms of presentation. The scoring procedure which treats all the components in a uniform manner leads to ignoring of structural semantics exposed. This research work harnesses this structural diversity through the classification of segments into various classes. These classes are associated with the corresponding weight to incorporate a variable magnitude approach.

Question 3: How to harness the intra-segment features for content scoring?

The unique feature of the web contents is their hyperlinked nature and the presentation layer pluralism. There exist various dimensions with which this pluralism can be addressed. This research work proposes a six dimensional modeling of web page which covers multimodal components like links, images etc.

Question 4: How to address the constantly evolving user specific information requirement context?

The information requirements of each user differ due to various local and global parameters. The information requirements of users which are evolving constantly need to be included in the content scoring procedure to make it user specific. This thesis attempts the personalization by incrementally amalgamating the user profiles using a hybrid approach.

In order to answer the above mentioned research questions, this thesis presents a model termed “SCOPAS” which is aimed at modeling the various dimensions of web pages for content scoring with the help of segmentation, classification and personalization. An overview of the SCOPAS model is presented in the following section.

1.4 The SCOPAS model

This thesis proposes a multimodal modeling of web pages for content scoring, termed as "SCOPAS" (**S**emantic **Co**mputation of **Pa**ge **S**core). This section explores the objectives of the SCOPAS model and its components.

1.4.1 Research Objectives

The Objectives of this research work are as illustrated in Fig 1.1

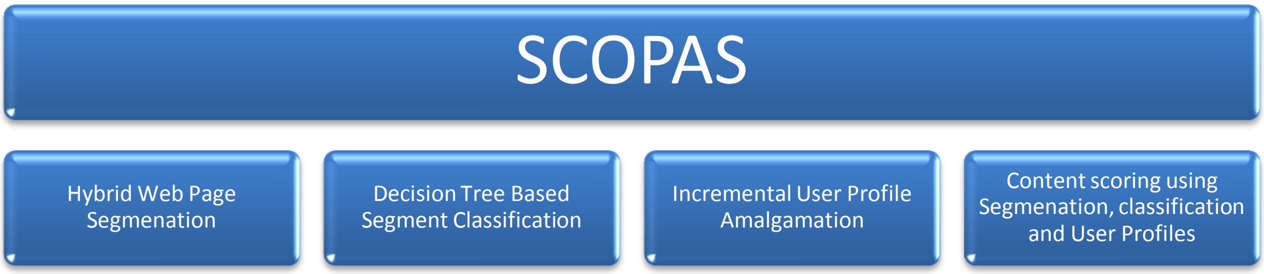

SCOPAS
Hybrid Web Page Segmentation
Decision Tree Based Segment Classification
Incremental User Profile Amalgamation
Content scoring using Segmenation, classification and User Profiles

Fig 1.1: Research Objectives

- Developing a Hybrid web page segmentation technique incorporating page tree and densitometry.
- Developing a web page segment classification model based on the Decision Trees and features extracted from the segments.
- Developing a method to build the profile of the user incrementally by action monitoring and content evaluation.
- Developing a Multimodal Web Page Model for Content Scoring Based on aforementioned Segmentation, Classification and user profiling techniques.

1.4.2 The SCOPAS Components

The SCOPAS model proposed by this research incorporates four major components as illustrated in Fig 1.2. This section provides a brief overview about the components of the proposed SCOPAS model.

The SCOPAS model proposed by this thesis approaches the content scoring problem in a multimodal manner i.e. the segments of the web pages are evaluated using a scoring procedure which involves various components of the segments like "link", "image", "visual feature" etc. These components are evaluated individually and the score of the segment is computed by fusing these different scores, using a variable magnitude approach.

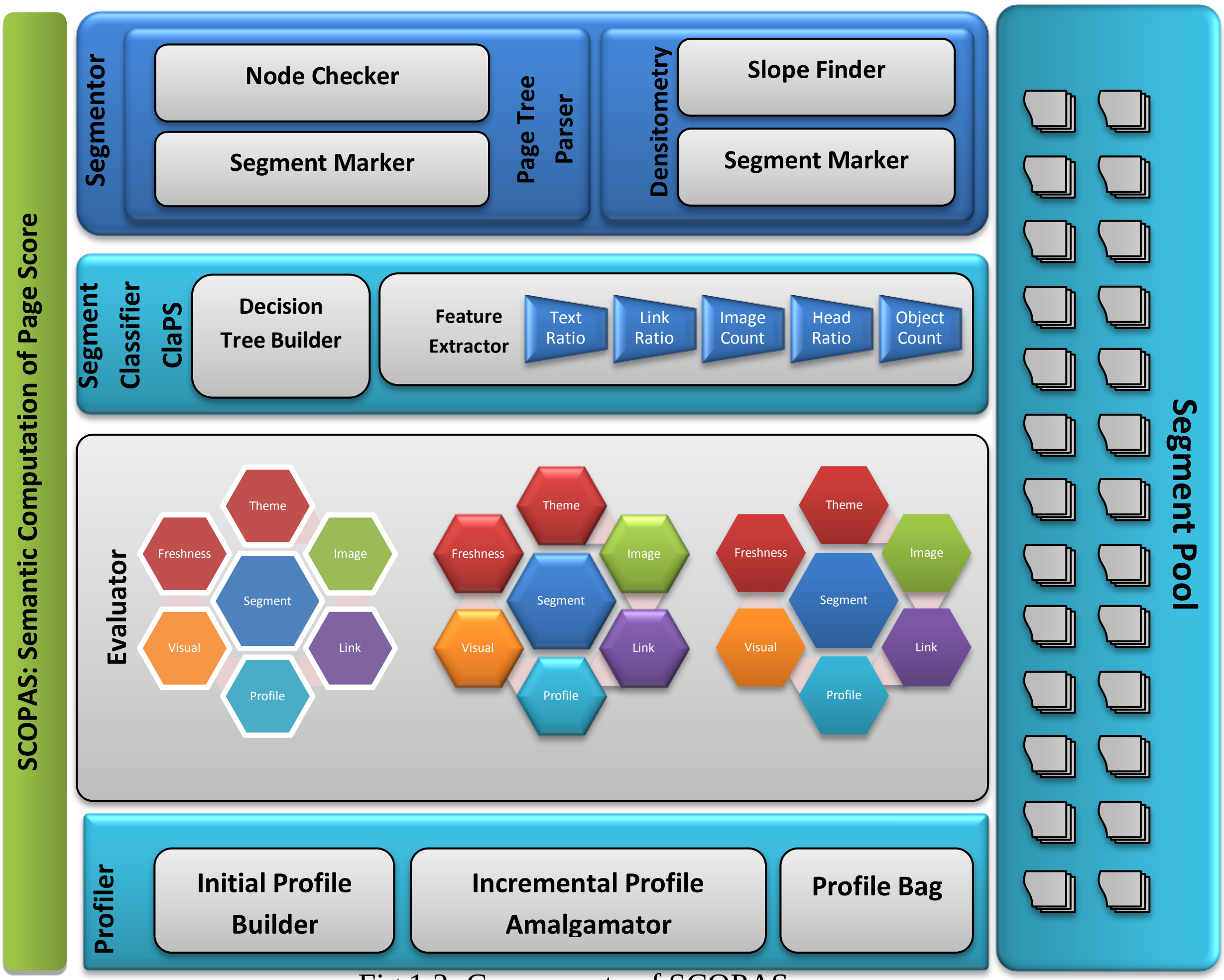


Fig 1.2: Components of SCOPAS

1.4.2.1 The Segmentor

The Segmentor component of SCOPAS performs the task of segmenting the web page into smaller blocks. These segments are built using a hybrid segmentation technique incorporating page tree and densitometry. The page tree technique works at the in-memory tree representation of the page. The densitometry computes the slope of content density among the page-components and utilizes it for marking the segment boundary. The Segmentor component identifies the segments and stores it into the segment pool. The Segmentor component is elaborated in detail in Chapter III.

1.4.2.2 The Segment Classifier

The Segment classifier component is termed as “ClaPS” (Classification of Page Segments). The role of segment classifier is to classify the segments into five different classes identified by this research work. The classification is carried out with the help of features extracted from the segments and training the decision trees based on these segments. The segment classifier is elaborated in detail in Chapter IV.

1.4.2.3 The Evaluator

The Evaluator component is responsible for computing the score of the page in a bottom-up manner by fusing the scores of individual segments. The Evaluator component computes the score in a multidimensional manner. This thesis introduces six different weight coefficients in computing the score. The score for individual coefficients are boosted with the help of "Coefficient Strength Factor". The SCOPAS-Evaluator component is explained in detail in Chapter V.

1.4.2.4 The Profiler

The profiler component incorporates personalization in the content scoring process. The personalization is achieved with the help of "Profile Bag" which is generated using a hybrid approach. The profiles are constructed in an incremental manner which involves two major steps. Initially an explicit preference collection technique is adopted with the help of FOAF (Friend Of A Friend)[2]. The profiles are incrementally amalgamated by using an implicit technique known as action monitoring. The incremental amalgamation refers to the process of enriching the initial profile bag by incorporating additional profile terms which are gathered by monitoring the user actions and evaluating the pages using various parameters like "time spent", "bookmarking". The SCOPAS profiler component is elaborated in detail in Chapter VI.

1.5 Organization of the Thesis

This thesis is aimed at web page modeling for content scoring in a multimodal manner, with the help of web page segmentation. The method proposed by this thesis incorporates a variable magnitude approach for different components of the pages, based on various structural and content semantics. The technique proposed by this thesis encompasses personalization with the help of user profiles which are amalgamated in an incremental manner by using a multidimensional approach.

[2] http://xmlns.com/foaf/spec/

The remainder of this thesis is organized as listed below:

- Chapter II presents a survey of various components of the SCOPAS model. It includes a survey on web page segmentation, classification, scoring of contents and user profiling. The survey of the above specified domains is provided with specific emphasis on the techniques utilized in this thesis.

- Chapter III elaborates about the proposed web page segmentation technique. This chapter covers the need for incorporating segmentation in the proposed model and the rules adopted for the segmentation. The page tree and densitometry components of the proposed segmentation techniques are explored in this chapter.

- Chapter IV illustrates the segment classifier component which is termed as "ClaPS" (Classification of Page Segments). This chapter covers the needs for incorporating the segment classification, the category of classification adopted by this thesis. The segment features which are utilized to classify the segments into five different categories are also defined in this Chapter.

- Chapter V explores the "SCOPAS-Evaluator" which is aimed to evaluate the score of the segments by fusing the scores computed for individual segments. Various weight coefficients like theme weight coefficient, image weight coefficient, link weight coefficient and the coefficient strength factor proposed by this thesis are explored in this chapter.

- Chapter VI is about the "SCOPAS Profiler" which is utilized to incrementally amalgamate the user profiles. The need for user profiles and the profile bag construction techniques are explored in this Chapter. Both the initial profile building and the incremental amalgamation of profile by monitoring user actions are explored in this Chapter.

- Chapter VII is broadly divided into two major sections: The first section explores the experiments conducted on the SCOPAS model and analyses the results with the help of various metrics. The second section is about the realization of the model in various application specific areas. Four different realizations of the model are explored in this chapter with details on their results analysis.

- Chapter VIII list out the conclusions derived from this research work and the future directions.

Chapter II: Motivational Work

This Chapter deals with the survey of motivational work carried out in the fields encompassed in this research work. The proposed research work entitled "Multimodal, Web Page Modeling for Content Scoring based on Segmentation and Incremental Profile Amalgamation" involves research components from various fields as illustrated in Fig 2.1.

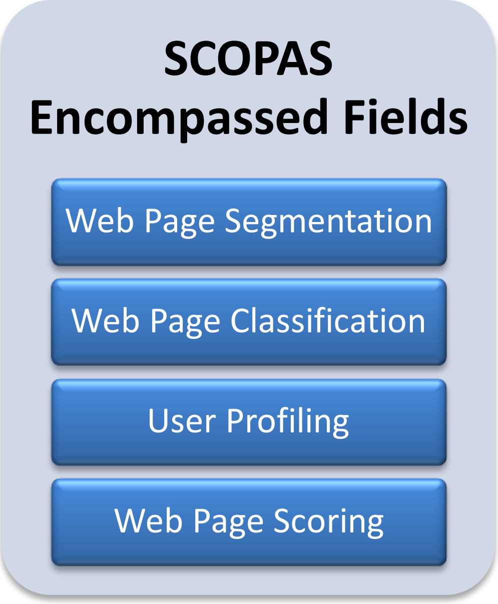


Fig 2.1 Encompassed Fields

This research work proposes a web page segmentation technique for splitting the web page into smaller components based on structural and content semantics, as one of its objectives. Hence a detailed survey of web page segmentation is provided in this chapter. The proposed model performs classification of segments, to associate different weights to segments of various classes. In order to provide details of existing classification techniques, an overview of the motivational works carried out in this regard are provided. As the proposed model incorporates personalized scoring of web pages, a survey of user profiles representation techniques and web page scoring are also provided in this Chapter. The rationales for selecting specific techniques are also discussed in this Chapter.

This Chapter is organized as follows: Section 2.1 deals with the web page segmentation by exploring various techniques and approaches. The classification of web pages is discussed in Section 2.2. The user profiling methods are elaborated in Section 2.3 which covers profile data collection techniques, profile storage approaches and profile locations. In Section 2.4, the text based and hyper link based scoring techniques for web pages are elaborated.

2.1 Web Page Segmentation

The World Wide Web has evolved from a static set of hyperlinked documents into a richly dynamic content delivery medium which has led to the fact that web pages have become diversified in nature. Different portions of the same web page might hold contents of varying nature which has opened-up an integral research question of segmenting them into smaller, relevant pieces in order to utilize them at the micro-level in various scenarios. These smaller sections of the webpage are called *segment* which is defined as follows: (Chakrabarti et al., 2008)

> "*Segment is a fragment of HTML, which when rendered, produces a visually continuous and cohesive region on the browse window and has a unified theme in its content and purpose*"

The process of identifying these segments is called segmentation. The web page segmentation is an active research topic in which a spectrum of approaches is proposed by various researchers (Cai et al., 2003; Cao et al., 2010; Chakrabarti et al., 2008; Kohlschütter and Nejdl, 2008; Liu et al., 2011; Vineel, 2009; Yesilada, 2011).

In the early days, segmentation was attempted manually (Takagi et al., 2002) which didn't evolve further due to various bottlenecks like scale of the web, dynamism of its content, level of accuracy and the cost associated with the process. All of these hindrances gave its way to the automation of segmentation process by using various approaches.

The web page segmentation process can be carried out either in a top-down manner or in a bottom-up manner. In top-down approach the process begins with considering the entire web page as one segment and subsequently its segments are identified. In the bottom-up approach, the process begins with the smaller elements (nodes of the page tree) of the page and deciding whether to include the current element under consideration into the earlier segment or make it into a new segment.

The web page segmentation is not a trivial task. The challenges associated with the web page segmentation are listed out in the following section.

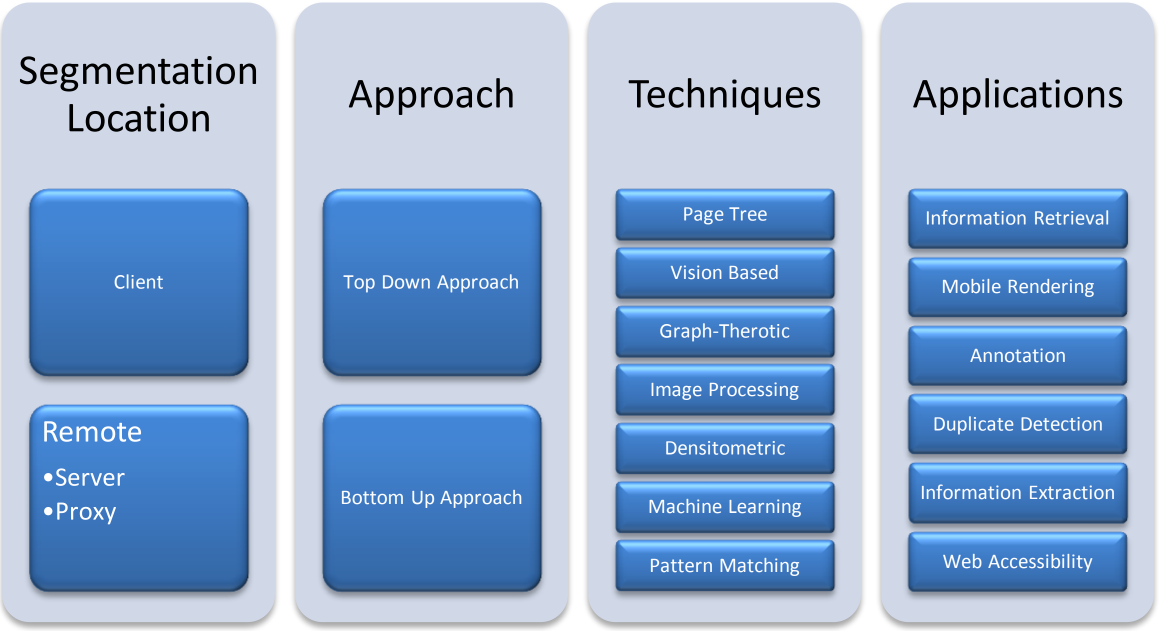


Fig 2.2. Segmentation Survey

2.1.1 Web Page Segmentation Challenges

Web page segmentation inherently poses many challenges (Fauzi et al., 2009; Yesilada, 2011).

- Each web page has its own layout which leads to difficulties in identification of the segment boundaries.
- Non-adherence of rules in the source of many web pages, which leads to errors during parsing of its layout tree.
- Requirement of a quicker response which asks for the entire process of segmentation to be completed in a swift manner.
- As web pages hold actionable blocks which could dynamically change the presentation of the web page, poses the challenge, in terms of handling the layout of such blocks.

Fig 2.2 illustrates the various dimensions adapted to systematically explain the survey of web page segmentation techniques reported in the literature.

2.1.2 Segmentation Location

The web page segmentation process can be carried out at various locations. The web page segmentation based on its location can be divided into following types:

- Client Side Segmentation
- Remote Segmentation
    - Server Side Segmentation
    - Proxy based Segmentation

2.1.2.1 Client Side Segmentation

The client side segmentation refers to the process of carrying out the segmentation in the client machine which has been explored in detail in the literature (Ahmadi and Kong, 2008; Borodin et al., 2007; Chakrabarti et al., 2008; Mahmud et al., 2007a; Milic-Frayling and Sommerer, 2002; Vineel, 2009; Yin and Lee, 2004). The client side segmentation reduces the overall load on the server. As this type of segmentation purely depends on the processing power of the client hardware which varies drastically across the spectrum, the segmentation time also varies in direct proportion to the processing power of the client hardware. If the client device is a mobile terminal then the battery power utilization for the segmentation process also needs to be taken for consideration. Due to these reasons the client side segmentation is not adopted in this research.

2.1.2.2 Remote Segmentation

In this type of segmentation the process of dividing the web page into various segments is carried out in a remote machine rather than at the client location. The remote segmentation can further be classified into server based segmentation and proxy based segmentation.

2.1.2.2.1 Server Side Segmentation

In server side segmentation, the process of identifying the segments is carried out at the server side. The server side segmentation has been studied in detail in the literature (Cao et al., 2010; Chen et al., 2005; Wang et al., 2004). The image processing based techniques have been used by Cao et al., for segmentation. In Chen et al., the segmentation is carried out by “implicit separator detection” and “explicit separator detection”. Wang et al., have used segmentation for efficient image retrieval with the help of Vision based page segmentation (VIPS). Segmentation at the server side can utilize the abundant computing power available at the server thereby making the process faster. At the same time if the number of simultaneous clients goes beyond a threshold, the server shall get overloaded which might reduce the overall performance. But this issue can be addressed by having adequate resources in the server machine.

2.1.2.2.2 Proxy Server Based Segmentation

An alternative for not carrying out the segmentation, neither at the client nor at the server is the proxy server based approach. The proxy server based approach to segment is analyzed in detail by various researchers(Baluja, 2006; Hattori et al., 2007; Xiang et al., 2007; Yang and Shi, 2009). Baluja and Hattori have utilized the segmentation for rendering of content in mobile devices, by introducing an additional layer of proxy server between the mobile device and the web server. The segmentation is carried out with the help of machine learning techniques by Baluja. The content distance based approach is adapted by Hattori. The segmentation is attempted with the help of Gestalt theory by Xiang et al. An enhanced version of Gestalt theory is utilized by Yang and Shi for segmenting the web pages.

In this type of segmentation the process is made independent of the processing power of the client hardware and at the same time the load on the server is also kept under control by relocating the segmentation process to a proxy server. The success of this type of segmentation purely depends on ensuring of proxy server up-time to the maximum and reduction in the latency of response from the proxy server.

This research work proposes a server side web page segmentation technique. The rationale for selecting server side segmentation is as follows: The client devices which are used to access the web pages ranges in their processing capability across a spectrum. If the segmentation is carried out at the client machine then the efficiency of segmentation would be heavily dependent on the processing capabilities of the client device. Another reason is the consideration of limitations on the power source available in the client side. If the segmentation is performed at the client device then the power requirements for the processing logic of segmentation should also be kept under control. The proxy based approach is not adopted in the proposed research work, as it makes an additional layer of dependency. The most important factor for carrying out segmentation at the server side is that, the segments built have to be evaluated with a variable magnitude and personalized approach before rendering the results to the users.

The web page segmentation can be carried out by following different paths. A survey of various web page segmentation techniques are explored in the following section.

2.1.3 Web Page Segmentation Techniques

The web page segmentation is explored in detail by many researchers through various approaches and this section explores these techniques. Although the list of techniques highlighted here are not exhaustive but it covers majority of approaches. A broad classification of these approaches is given below:

- Page Tree Based Segmentation
- Vision Based Segmentation
- Densitometric Segmentation
- Image Processing Based Segmentation
- Graph Theory Based Segmentation
- Pattern Matching Based Segmentation
- Machine Learning Based Segmentation
- Hybrid Approaches

The above mentioned approaches are explored in detail in this section.

2.1.3.1 Page Tree Based Segmentation

The process of segmenting a web page differs from the segmentation of traditional text documents primarily due to the inherent content structure and markup associated with the web pages. Though the web pages are constructed through simple markups, the real power of the web pages lies in their ability to represent them in memory as a tree. The Document Object Model based page tree representation has been used by many researchers to carry-out the segmentation process(Buyukkokten et al., 2001; Vineel, 2009). In DOM based segmentation, the segment boundary identification is done with the help of markup information provided by the tags.

In the page tree based segmentation (Buyukkokten et al., 2001), the Document Object Model nodes like “table”, “paragraph” are utilized in building the segments. Another approach (Lin and Ho, 2002) considers only the markup used for the “table” and utilizes an entropy based approach in segment building process. The core idea beneath usage of “table” and “paragraph” markup during segmentation is due to the fact that these elements are not only utilized for simply creating a table or a paragraph but they are primarily involved in constructing entire layout of the web page.

The DOM tree mining based approach (Vineel, 2009) characterizes the nodes of DOM tree structure, based on their “Content Size” and “Entropy”. The content size computes the textual size of the node and entropy measures the strength of “local patterns”.

The Document Object Model based segmentation has certain inherent limitations, like the usage of DOM nodes alone in the segmentation process may mislead in certain scenarios. For example in a web page, the nearby elements can be from different parent nodes altogether, and in such scenarios the utilization of DOM nodes alone would construct improper segment boundaries.

This research work proposes a hybrid segmentation technique which utilizes the page tree based segmentation as one of the components. The details of the proposed segmentation technique are provided in Chapter III.

2.1.3.2 Vision Based Segmentation

Vision Based Segmentation technique tries to solve the segmentation problem by simulating the manner in which the user understands the web page’s layout based on his/her visual perception. The pioneering work, (Cai et al., 2003) VIPS – “Vision based Page Segmentation” utilizes the visual perception of the user for segmentation. VIPS is a top-down segmentation technique. VIPS carries out segmentation in three different steps which involves Block Extraction, Separator Detection and Content Structure Construction

Initially the web page is segmented into few big blocks and the algorithm is applied recursively to find out smaller blocks within. The Degree-Of-Coherence (DoC) parameter is defined according to the visual difference. VIPS has proposed thirteen different heuristics rules and different cues like tag cue, color cue, text cue, size cue etc. After the extraction of the blocks, the visual separator detection process is carried out followed by the content structure construction.

There exist many research works (Cai et al., 2004; Saad and Gançarski, 2010; Wu et al., 2011) which have adopted the VIPS algorithm for various purposes like block level link analysis, archiving process, evaluating the visual quality of the web page etc. The research work (Akpinar and Yesilada, 2012) extends the VIPS algorithm by the incorporation of additional definitions and visual cues to enrich the block extraction phase.

2.1.3.3 Densitometric Segmentation

In the page tree based segmentation, the DOM nodes are used for segmentation and in Vision based approaches additional visual cue and heuristics were added. These approaches would underperform in the following scenarios: improper usage of tags, use of external style definitions. In order to cope with such issues, the web page segmentation has been approached with text density which is known as “Densitometric approach”. The text density is defined as the ratio between the total number of words in a segment and the height of the rendered or printed segment (Spool, 1999). An approach to segmentation, based on densitometry (Kohlschütter and Nejdl, 2008) defines the density of the text in a segment as shown in Eq. (2.1).

$$D(segment) = \frac{\text{Number of tokens in segment}}{\text{Number of lines in segment}} \qquad (2.1)$$

To compute the “Number of lines in Segment”, the content is word-wrapped (not its rendered representation) at a constant line width of specified number of characters. The decision to combine the adjacent segments x, y is taken by “block fusion” algorithm proposed by the author. The decision to fuse the segments is decided by the slope delta value computed by “block fusion” which is shown in Eq. (2.2)

$$\Delta D(x, y) = \frac{|D(x) - D(y)|}{\max(D(x), D(y))} \qquad (2.2)$$

If the slope delta is less than a threshold value then the adjacent segments are fused together. This approach falls under the bottom-up segmentation type. This Densitometric approach is utilized for analyzes of web template content and content extraction as well (Chun et al., 2012; Kohlschütter, 2009).

The segmentation technique utilized by “SCOPAS Segmentor” belongs to a hybrid category which encompasses page tree based segmentation and densitometric segmentation.

2.1.3.4 Graph Theoretic Segmentation

The web page segmentation is attempted with the help of graph theoretic approaches as well. The approach “GCuts” (Chakrabarti et al., 2008), utilizes weighted graphs and formulates to an optimization problem. The weights are used to identify whether the node should be placed in the same segment or in different segments. This leads the segmentation problem as a minimization problem. The energy minimizing cuts in graphs are used in this approach. The success of this type of segmentation heavily depends upon the weights assigned to the edges in the graph. Learning of these weights from manually labeled data is also projected in the above mentioned work.

A similar work (Liu et al., 2011) attempts the web page segmentation using graph theoretic approach using an undirected planar graph which work employs the “Gomory-Hu” tree (Gomory and Hu, 1961) algorithm.

Another approach based on graph theory (Yin and Lee, 2005), segments the web page into various categories like content, related links and advertisement. For each category, specific parameters are used in the identification process. For example for content category the parameters like elements size, location and tag type are used.

2.1.3.5 Image Processing Based Segmentation

The image processing based approach to web page segmentation treats a web page as an image and applies various techniques to slice the image into distinct segments. An approach to web page segmentation (Cao et al., 2010) using image processing approach employs iterative shrinking and dividing, to build the segments. The authors have introduced dividing conditions and the concept of dividing zone. Based on this, the image of the web page is divided into sub-images by shrinking and splitting them iteratively. The initial snapshot of the image is preprocessed with edge detection technique like “Canny”. The web page image is repeatedly divided until further division is not possible. The authors claim that, their work exhibits expansibility and good performance. The issue with this type of segmentation is the loss of semantic dimension during the process.

A similar approach (Pnueli et al., 2009) utilizes the edge detection techniques and searches for long edges, directed either in horizontal or vertical directions. After that, it looks for areas which contain information and segments them into distinct layout elements. The algorithm is designed in such way that it locates individual GUI components and utilizes the optical character recognition (OCR) for identifying the text information.

2.1.3.6 Pattern Matching Based Segmentation

The pattern matching among the elements of the web page is also harnessed to carry-out the web page segmentation. The REPS – “Repetition based Page Segmentation” algorithm (Kang et al., 2010) achieves segmentation by recognizing repetitive tag patterns called key patterns in the Document Object Model tree of the page. This algorithm detects key patterns in a page, through generating virtual nodes to properly segment the nested block in the web page. A key pattern is a repetitive pattern in a sequence that is longest and most frequent. One of the limitations of this approach is the inability to preset the number of blocks expected from the segmentation process which leads to inflexibility in segmentation since the fine-grained level of logical blocks might vary for different applications.

A general issue with this type of segmentation is that it relies only on the “repetition factor” of the code of web page, which can mislead in certain scenario like varying code elements for similar blocks.

2.1.3.7 Machine Learning Based Segmentation

The machine learning techniques are also utilized in achieving the web page segmentation process (Baluja, 2006). In this work, the page is segmented into nine distinct regions to render it conveniently in a small screen mobile device. This work converts the web page segmentation problem into a machine learning task, where the ideas of entropy reduction and decision trees are utilized. The recursive method of segmentation is employed in this approach to achieve the required result.

In addition to entropy reduction and decision tree learning, this work also includes few heuristics like bias on the size of the segment, selection of proper Document Object Model elements to mark the segment boundary.

The CSurf (Mahmud et al., 2007a) includes machine learning in the process of delivering content to the visually challenged user with the help of web page segmentation. The Geometric Clustering approach indicated in this work is based on the Mozilla’s Frame tree and semantic spatial locality (Mukherjee et al., 2003). This work defines “block rank” as the learning problem and utilizes the Support Vector Machine (SVM) for automatically annotating the web pages.

A Hierarchical Web Page Segmentation Algorithm using Machine Learning approach (Ito et al., 2008) segments the pages into hierarchical units and utilizes the Support Vector Machine (SVM) in building the segments for rendering of pages in small screen displays. The technique proposed by the author has achieved higher precision than the existing methods.

2.1.3.8 Hybrid Segmentation

Hybrid web page segmentation is the process of solving the segmentation problem by borrowing the idea from more than one type of segmentation listed above. Various researchers have utilized this hybrid segmentation method to achieve better results. An approach which combines both the Document Object Model and Visual Layout information (Hattori et al., 2007), measures the content distance using tree nodes and utilizes the web page layout information as well. This approach solved the issue of mismatch between the visual distance and the content distance in the source tree. The research work (Xie et al., 2005), initially utilizes the VIPS algorithm for identification of page segments and then employs machine learning techniques like Support Vector Machine for handling the identified segments.

Apart from the earlier approaches discussed, there exist other ways of performing the segmentation like clustering algorithms, psychology based algorithms which utilizes Gestalt theory etc. As each of the segmentation technique discussed above has its own merits and drawbacks, this thesis introduces a hybrid web page segmentation technique incorporating a page tree and densitometry technique which is explored in detail in Chapter III.

2.1.4 Applications of Web Page Segmentation

Various researchers have utilized the web page segmentation to solve specific issues. This section explores such applications of web page segmentation.

- In the field of information retrieval, web page segmentation is utilized to improve the retrieval efficiency (Bar-Yossef and Rajagopalan, 2002; Cai et al., 2004; Xie et al., 2005).
- The web pages are primarily designed to render in large screens of personal computers, which are not suitable for smaller sized screens. Hence the web page segmentation is widely used to render the page into small screen portable devices (Hattori et al., 2007; Kang et al., 2010; Xie et al., 2005).
- The web page segmentation technique is utilized for annotating the contents of the web pages (Mukherjee et al., 2003).
- The web page segmentation is effectively utilized to detect duplicates, by various researchers (Chakrabarti et al., 2008; Kohlschütter and Nejdl, 2008).
- The issue of accessibility of web contents by visually challenged users is addressed with the help of web page segmentation (Mahmud et al., 2007a).

The proposed SCOPAS model has been realized using various applications like result ranking, change detection etc. which are dealt in detail in Chapter VII.

This section explored various segmentation approaches, techniques and applications. The proposed SCOPAS model utilizes the server side, hybrid segmentation incorporating page tree and densitometry.

Apart from the segmentation, the proposed SCOPAS model involves a variable magnitude approach for scoring the web page segments by classifying them into various types. Hence, a survey of various classification types, modes and techniques are explored in the following section.

2.2 Classification of Web Pages

Classification of web contents is the process of identifying the category of web pages. Hence it becomes a web page categorization problem (Qi and Davison, 2009). Web page classification can be considered as a specialization of text classification problem (Hernández et al., 2012). The text classification is also studied in detail by various researchers (Bennett et al., 2005; Cardoso-Cachopo and Oliveira, 2003). The web page classification differs from the text classification primarily due to the semi-structured and interlinked nature of the web contents (Maan and James, 2012). As the web page contains visual markups and hyperlinks to other documents, the generic text classification techniques cannot be applied to web page classification effectively (Ong et al., 2012). This section presents a survey of web page classification through various dimensions as type, mode and technique which is illustrated in Fig 2.3.

2.2.1 Classification Types

The classification of web pages can be carried out based on two major components: the content and the structure. The main objectives of both these types of classifications are different. The content based classification is used to categorize the web pages into various classes on the basis of the semantics of their content. The structure based classification concentrates on the visual characteristics of the page for classification purpose.

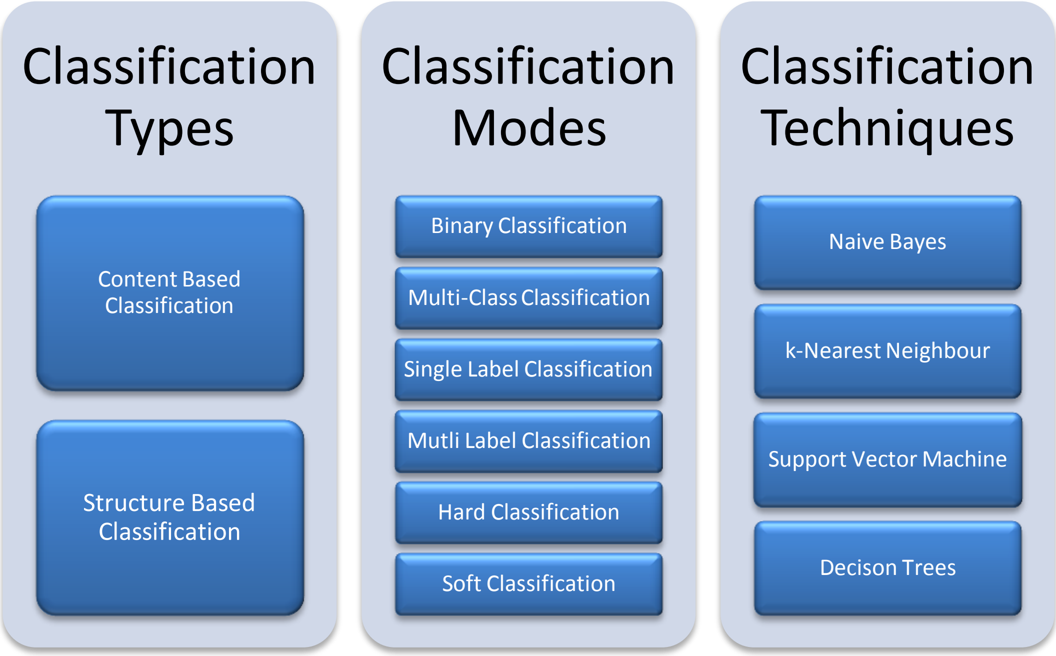


Fig 2.3. Classification Survey

2.2.1.1 Content Based Classification

The content based classification focuses on the semantics of the contents present in the web pages. The content based classification can be used for identifying the subject of the web page (Qi and Davison, 2009). It can be used to identify the function of the web page, like "home page", "about us" etc. Sentiment classification is also carried out based on the content of the web page, which is used to identify the opinion expressed in a particular web page. The content based classification is adopted for identification of web spams with the help of web topology (Castillo et al., 2007; Gyongyi and Garcia-Molina, 2005).

2.2.1.2 Structure Based Classification

The structure based web page classification utilizes the visual features of the web pages to classify. Using the visual features the page components can be classified as “advertisement” or “copyright notice” etc. (Burget and Rudolfová, 2009). There exist approaches which classify the web pages by considering their structure into “information pages”, “research pages” etc.(Asirvatham and Ravi, 2001).

This research work proposes a structure based classification of segments by utilizing various features extracted from the web page segments. The details on the classification are explored in Chapter IV.

2.2.2 Classification Modes

Various modes of classification are explored in this section. There exist many modes of classification based on the total number of classes, count of classes a web page can be assigned to. On the basis of the total number of classes there are two modes: binary classification (Goh et al., 2001) and multi-class classification (Dietterich and Bakiri, 1995, 1994; Mayoraz and Alpaydin, 1999). Based on the number of classes a web page can be assigned to, there exist two modes of classification: Single Label classification and Multi Label classification (Qi and Davison, 2009). Based on degree of association there are two types classification namely “hard classification” and “soft classification” (Wahba, 2002, 1999).

2.2.2.1 Binary Classification

In the binary classification, the number of available classes is only two. Any web page can be assigned to either of these two classes. Binary classification is comparatively simpler due to the restriction on the maximum number of classes. This research work has not utilized the binary classification, since it deals with more than two classes.

2.2.2.2 Multi-class Classification

In multi-class classification there is no restriction on the total number of classes available. Specific methods are available for multi-class classification like “pairwise coupling" (Ting-Fan Wu et al., 2004). Multiclass classification has been studied in detail in the literature, as many of the classification problems belongs to this mode (Dietterich and Bakiri, 1995, 1994; Mayoraz and Alpaydin, 1999). The multiclass classification is utilized for understanding the importance of various blocks in the pages (Song et al., 2004) by adopting the vision based segmentation process. The segment classification adopted in the proposed SCOPAS model belongs to the multiclass classification mode.

2.2.2.3 Single Label Classification

In the single label classification mode, the given web page can be assigned to only one of the available classes(Sun et al., 2002). In other words, any given web page cannot belong to more than one category at any point in time. The segment classification adopted in the proposed SCOPAS model belongs to the Single Label classification mode.

2.2.2.4 Multi Label Classification

In the multi label classification mode, the given web page can be assigned to more than one available class. The multi label classification is used in various domains like music categorization, scene classification etc. (Tsoumakas and Katakis, 2007).

2.2.2.5 Hard Classification

In the hard classification mode, a given web page is either completely assigned to a class or not assigned(Sun et al., 2002). The hard classification doesn't involve partial association process. The hard classification is adopted in those scenarios where there is no doubt regarding the membership of an item to a particular class. The segment classification proposed in the SCOPAS model belongs to the hard classification category. A given web page segment is assigned to a class completely.

2.2.2.6 Soft Classification

In the soft classification mode any given web page is assigned membership to a class with a degree of association. It is done on the basis of the probability with which the item can be associated with a particular class (Wahba, 2002, 1999). In the proposed SCOPAS model, this type of classification is not adopted in order to avoid fuzziness in the class weight score allocation process. As stated earlier, the hard classification is adopted in the SCOPAS model for the precise computation of class weight score.

2.2.3 Classification Techniques

There exists a wide array of techniques for classification such as Naïve Bayes, k-NN, SVM and Decision trees which are presented in this section with their merits and demerits. This section explores the techniques mentioned in Fig 2.2.

2.2.3.1 Naïve Bayes Classification

Naïve Bayes is a classification technique based on the probability theorem called Bayes' rule (Lewis, 1998; McCallum and Nigam, 1998). Naïve Bayes primarily works on the principle of maximum likelihood. The Naïve Bayes method is based on the conditional probability for the classification task. It uses the prior probability and posterior probability for the classification process. Naïve Bayes has been studied for hypertext categorization by Yiming Yang et al(Yang et al., 2002). Naïve Bayes classifier performs well with the limited number of classes. The performance of Naïve Bayes on web scale taxonomy is studied in the literature and improvements were suggested (Zhang et al., 2009).

2.2.3.2 k-NN Classification

The k-NN stands for "k" Nearest Neighbor classification. There are many research works which attempted the classification with the help of k-Nearest Neighbor technique(Kwon et al., 1999; Lam and Ho, 1998; Larkey and Croft, 1996). The k-NN classifier primarily works in two major steps. In the initial step, the k-nearest samples, in the training document are identified. The similarity score of the neighboring documents with respect to the test document is utilized. In the second step, the summation of these scores is used to identify the likelihood. In a study(Kwon et al., 1999), an enhanced version of the k-NN classifier is utilized to classify the web documents with the help of feature selection and term weighting.

2.2.3.3 Support Vector Machine

The Support Vector Machine (SVM) is introduced by Vapnik (Vapnik, 1999) for the classification purpose. The SVM represents the sample documents as points in the space. These points are placed in such a way that the points belonging to different categories are placed with maximum possible gap. The decision surface in Support Vector Machine is linearly separable (Sebastiani, 2002). The support Vector Machine can be applied to multiclass classification by reducing the multiclass problem into a collection of binary classification (Duan and Keerthi, 2005; Hsu and Lin, 2002; Xia et al., 2012). The Support Vector Machine, in combination with the context features of hypertext has been utilized in classification of web pages (Sun et al., 2002).

2.2.3.4 Decision Tress

Decision trees perform the classification task by dividing the instance surface recursively. Decision trees are popularly used for classification purpose (Apte et al., 1998) due to their inherent capability of being quicker and ease of use. A decision tree based automatic classification of web pages is explored for Yahoo! Japan web page collection. (Tsukada et al., 2001).

There exist many algorithms in decision trees like CLS – Concept Learning System(Hunt et al., 1966), CART - Classification And Regression Trees (Breiman et al., 1984), ID3 - Iterative Dichotomiser (Mitchell, 1997) and C4.5 (Quinlan, 1993). The C4.5 is an extension of ID3 algorithm. The C4.5 algorithm harnesses the information gain and utilizes recursion in building the decision tree.

An approach to classify the web page segments using decision tree is also explored in a study (Burget and Rudolfová, 2009) which proposes the classification of segments based on visual features. The utilization and efficiency of C4.5 classifier is established by the above work which formed the basic motivation of utilizing the decision tree algorithm C4.5 for the segment classification in SCOPAS model. The C4.5 algorithm works in two major steps. In the first step a decision tree is built and in the second step pruning of the built decision tree is carried out. The advantages of using decision tree classification lies in its ability to handle dataset with errors and managing the dataset with missing values(Mitchell, 1997; Quinlan, 1993).

For the purpose of classification of segments based on their structural characteristics, decision trees are utilized in the proposed SCOPAS model. The proposed segment classification model utilizes the J48 decision tree classifier which is based on the Quinlan’s C4.5 algorithm. The J48 is an open source implementation of C4.5 in the

Weka machine learning tool. The Weka machine learning workbench is introduced in the following section.

2.2.4 The Weka Machine Learning Workbench

This research work utilizes the Weka (Waikato Environment for Knowledge Analysis.) machine learning software "workbench" for performing the J48 based classification. Weka is a product of Waikato University (Hall et al., 2009). Weka supports a wide array of machine learning algorithms. Weka is designed in such way that either the algorithms can be directly used from Weka interface or they can be embedded in the Java code.

The attributes of the dataset are given in a file format called ARFF (Attribute-Relation File Format). The datasets can be supplied to specific machine learning algorithms in Weka either using the Graphical User Interface (GUI) or Command line mode.

The detailed metrics display like Receiver Operating Characteristics (ROC), Precision, Recall, and F-Measure is another advantage of using the Weka workbench.

In this research work, the extracted features from the segments in the web pages are supplied through an ARFF file and the J48 decision tree algorithm is utilized in classifying the web page segments into various predefined categories based on structural aspects.

In this section various classification types, modes and techniques are explored in detail and the classification technique adopted in the proposed model is also highlighted. As this research work incorporates personalization, a survey of user profiles management is provided in the next section.

2.3 The User Profiles

The proposed SCOPAS model incorporates personalized scoring of contents i.e. the scoring of contents are based on the key-terms available in the user profiles. Hence this section elaborates about various works carried out in the personalization domain. The details about the type of user profile building technique used in the proposed model are also explained in brief in this section. The information requirement of one user is different from another due to the varying context (Bennett et al., 2012; Sontag et al., 2012; Zhou et al., 2012). To deliver the user specific information, the profile of the user needs to be built and maintained in a systematic manner (Mianowska and Nguyen, 2012). In this section various techniques used for collecting the profile data, representations used for storage of profiles and profile locations are elaborated (Fig 2.4).

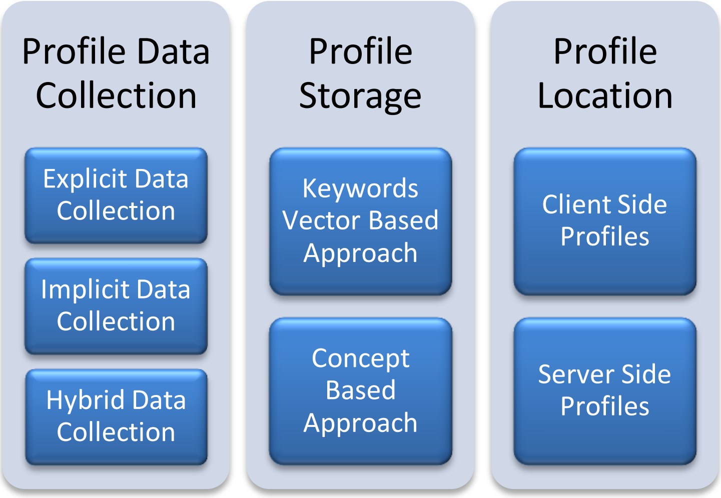


Fig 2.4: User Profiles Survey

2.3.1 Profile Data Collection

The first step in providing personalized services is to collect the profile data of the user. The profile data of the user can be collected in three different ways (Gauch et al., 2007). They are implicit data collection, explicit data collection and hybrid method. All these approaches have their own merits and demerits. Various works carried out in the profile data collection are discussed in this section.

2.3.1.1 Explicit Data Collection

In the explicit profile data collection method, the user has to provide the details explicitly. Many research works are carried out using explicit profile data collection (Shavlik and Eliassi-Rad, 1998; Shavlik et al., 1998). In the explicit profile data collection technique, the user has to create the preferences explicitly by filling the data in the relevant forms or rating the contents. Explicit user feedback based link recommendation attempts to find the interesting links based on the feedback provided by the user on an available link (Pazzani et al., 1996). The explicit profile building with the help of Artificial Neural Networks is explored in Wisconsin Adaptive Web Assistant (Shavlik and Eliassi-Rad, 1998; Shavlik et al., 1998). The issues with explicit profile data collection includes the increased load on the user, improper details provided by the user etc.

2.3.1.2 Implicit Data Collection

In the implicit profile data collection, the user need not specify any details explicitly. The required data are collected by monitoring the user interaction with the web page. The profile data collection can be carried out in an implicit manner using a wide variety of techniques (Iglesias et al., 2012; Kelly and Teevan, 2003). In the implicit profile building process the parameter “time” i.e. the amount of time spent on a page is taken into consideration (Claypool et al., 2001; Fox et al., 2005; Goecks and Shavlik, 2000).

The implicit profile data can be collected through various modes apart from the time spent such as browsing history, search logs, desktop agents etc. (Adar et al., 1999; Barrett et al., 1997; Chien, 2000; Liu et al., 2002; Pretschner and Gauch, 1999). In the browsing history based data collection the user’s past web navigation pattern is analyzed for building the profile. In the case of search logs, the queries entered by the user plays an important role in collecting the user profile data.

The desktop agents are installable client side tools which provide information regarding user interactions with the system. The drawback with the implicit profile data collection mechanism through the agents is the requirement of installation of a client side tool to monitor the user interactions.

2.3.1.3 Hybrid Data Collection

Both the explicit and implicit profile data collection techniques have their own pros and cons as stated above. The hybrid data collection method involves the positive aspects of both the implicit and explicit profile data collection techniques. A comparative study among the various profile data collection techniques have been carried out using the search activities of various users (Quiroga and Mostafa, 1999). The results of the study showed that, the explicit profile data collection achieved 63% precision, implicit profile data collection achieved only 58% precision. The study confirmed that the hybrid approach combining both the implicit and explicit techniques were able to provide an encouraging precision value of 68%.

This research work proposes a hybrid model of profile building by gathering the explicit preferences from the user directly and gathering the implicit preferences by action log monitoring. The profile of the user is incrementally amalgamated in the proposed SCOPAS model.

This section explored three different types of profile data collection techniques. Another important aspect with respect to the user profile, apart from the profile data collection technique, is the profile storage method which is explored in the following section.

2.3.2 Profile Storage and Representation

The profile data gathered through any of the techniques specified in the previous section has to be represented and stored in a standard manner, for further action. This section explores two different approaches for profile representation. They are the Keyword based approach and Concept based approach.

2.3.2.1 Keywords Based Approach

The keywords based user profile storage, functions on the basis of collection of keywords which expresses the user interest. The different keywords in the profile can be given a varying weight so that the priority among the keywords can be set. The keywords based profiles can be built either using the implicit or explicit feedback mechanisms. (Chen and Sycara, 1998; Moukas, 1997). The primary issue with the keywords profile is the context disambiguation as the same keywords shall mean different things. In order to overcome this difficulty separate keyword vectors have been used for different interests (Chen and Sycara, 1998).

The keywords based profiles can incorporate both positive and negative keywords. The presence of positive keywords denotes the user’s interest towards that term and the presence of negative keywords denotes the disinterest of users towards the term.

2.3.2.2 Concepts Based Approach

The concepts based profiles are different from the keyword based profiles in the representation of the concepts instead of simple terms. In line with the keyword based profile, the concept based profiles can also represent both the positive and negative preferences (Leung and Lee, 2010). The concept based profiles utilizes a concept hierarchy like "Open Directory Project" for their representation (Pretschner and Gauch, 1999; Trajkova and Gauch, 2003). In the concept based approach multilevel hierarchical concepts are used from the reference hierarchy.

The proposed SCOPAS model maintains the profiles with the combination of Keyword vector and Concept hierarchy from Open Directory Project (ODP). This multi-dimensional approach facilitates incorporating the advantages of both the Keyword based representation and Concept based representation.

2.3.3 The Profile Location

The user profiles represented with the techniques specified in the previous section can be stored either in the Client side or in the Server side. In the server side storage of user profiles, the gathered profile data is sent to the server machine for persistence (Kamba et al., 1997). In the case of client side profile representation technique, the gathered profile data is stored in the client machine itself which can be utilized for various tasks like search result re-ranking, user specific rendering etc. (Fredrikson and Livshits, 2011; Guha et al., 2009; Teevan et al., 2005; Toubiana et al., 2010; Xu et al., 2007).

In this research work, the user profiles are stored in the server side with the initial data provided by the user explicitly. The profile is updated constantly by monitoring the user action through a client side component and the gathered data is sent to the server for further action.

2.3.4 The FOAF

FOAF stands for "Friend Of A Friend". FOAF is a promising semantic web technology (Li Ding et al, 2005). FOAF can be used to represent the user interest in different dimensions and it can also be used to link the profiles of related users. FOAF is a XML based representation. There exists many research works on utilizing FOAF in enhancing the web scale information systems (Adamic et al., 2003; Golbeck et al., 2003; Grimnes et al., 2004).

FOAF documents are distributed across the World Wide Web. Studies were conducted to retrieve the FOAF documents from various sources for extracting the user information from the FOAF document and fusion of information (Ding et al., 2005). The FOAF documents are machine readable as the representation is given in XML.

The FOAF specification includes various terms which are primarily grouped into categories like “Core” and “Social Web” terms. The “Core” group includes terms like “name”, “age”, “title” etc. In the “Social web” group there are terms like “topic_interest”, “topic”, “weblog” etc.

The proposed user profile component in the SCOPAS model utilizes the FOAF for representing user profiles. It incorporates terms from both the core group and the social web group. Various FOAF terms are utilized in the proposed model to capture the user’s interest in a multidimensional way. These captured profile data are constantly enriched by user action logging and incrementally updating the profiles.

This section explored various techniques for managing the user profiles through various dimensions like storage techniques, data collection methods etc. A brief introduction to the FOAF based profile handling technique, which is employed in the proposed SCOPAS model is also highlighted in this section. The proposed model incorporates these user profiles in scoring the web contents. Various approaches to score the web pages are discussed in the following section.

2.4 The Web Page Scoring

With the massive explosion in the quantity of contents available in the World Wide Web, locating the relevant content has become the pivot around which the entire web information system revolves around. The user’s information requirement is expressed in the form of search queries, for which the relevant contents have to be retrieved from the colossal World Wide Web with the help of various scoring techniques. This section explores various web page scoring techniques using text based approaches and hyperlink based approaches.

2.4.1 Text Based Scoring

The relevance of a document with respect to the user supplied query is computed by analyzing the textual contents of the web page using various methods. This section explores important text based scoring techniques: Boolean method and Vector Space Model.

In the case of text based scoring, each page is represented as a collection of terms (Ponte and Croft, 1998; Robertson et al., 1995; Salton, 1968). Each word in the page is considered as a term (Robertson et al., 1995). Instead of considering the terms alone, the stemmed versions of the terms are also considered in some approaches for the purpose of scoring (Porter, 1980). The score of the page is computed by considering the term's "intra page weight" and "inter page weight" among the complete index.

2.4.1.1 Boolean Method

The Boolean method considers the page either completely relevant or completely irrelevant (Baeza-Yates and Ribeiro-Neto, 1999). The Boolean methods follow the principle of "all or nothing". There is no idea of partial relevancy in the Boolean method of scoring. The contents retrieved by the Boolean method are done on the basis of exact match. The major advantages of Boolean method are simplicity of the implementation and intuitiveness. Though the Boolean method is easy to implement, the biggest drawback of the Boolean method is the lack of ranking among the retrieved document collection. Another issue with the Boolean method is that for some queries the result list gets overloaded, but for some queries it doesn't return any result at all, due to the exact match condition (Cooper, 1988).

2.4.1.2 Vector Space Model

As indicated in the previous section, the non-ranked retrieval performed by the Boolean method falls short in huge sized corpus like World Wide Web. The Vector Space Model provides a solution to this problem by incorporating ranked relevance, among the retrieved pages. The Vector Space Model considers both the page and the query as a vector (Salton, 1971; Salton et al., 1975). The similarity among the query and the page are computed as the distance between the vectors representing the page and the query.

The Vector Space Model employs the ideas of "term frequency" (tf) and "inverse document frequency" (idf) for ranking the pages against the user supplied query (Jones, 1972). Term frequency refers to the number of times a term appears in a page. The inverse document frequency is computed as the logarithmic value of total number of pages in the index divided by the number of pages containing the term. The weights of the terms are computed as the product of “term frequency” and “inverse document frequency”. The distance between the page and the search terms is computed using functions like “Cosine Similarity” (Baeza-Yates and Ribeiro-Neto, 1999).

One of the drawbacks of Vector Space Model is that the lengthier documents get ranked higher due to higher term frequency value. In order to overcome this drawback, the document length normalization techniques have been proposed (Singhal et al., 1996). The length normalization is carried out by dividing the term frequency by the maximum number of times the particular term appears in any document in the complete collection. A recent study has concentrated on the usage of Vector Space Model in combination with other techniques like Ontology based information retrieval (Castells et al., 2007). In the proposed SCOPAS model, a variation of term frequency and inverse document frequency is utilized to compute the score of terms at the fine-grained segment level which is termed as “Inverse Segment Frequency (ISF)”. The Inverse Document Frequency (IDF) works at the page level whereas the proposed Inverse Segment Frequency works at the intra-page segment level. The SCOPAS model uses the Inverse Segment Frequency (ISF) in computing the segment score in a multimodal manner which is explored in detail in Chapter V of this thesis.

2.4.2 Hyperlink Based Scoring

The text based scoring techniques explored in the previous section focuses on the statistical properties, to compare the query and the document. The web pages can be scored by considering parameters which are outside of the page like links pointing to that page (Zhu and Gauch, 2000). There are studies conducted which focused on the anchor text of the links pointing to the web page in computing the relevance score of the web page (Craswell et al., 2001; Eiron and McCurley, 2003).

One of the most popular link based scoring algorithm is “Page Rank” which computes the score of the page by measuring the sources linking to that page (Brin and Page, 1998b). The context sensitive versions of “Page Rank” is also explored to rank the pages based on the context in which they appear (Haveliwala, 2003). The page scoring, based on the user’s link navigation pattern using “Hierarchical Navigation Path” is explored in a recent study (Li et al., 2012).

The relevance scoring of web pages based on “Hub Pages” and “Authoritative Pages” are explored in HITS- Hypertext Induced Topic Search algorithm (Kleinberg, 1999). The HITS algorithm is query dependent. The HITS algorithm generates two scores for a web page as “Hub Score” and “Authoritative Score”. Improvements over the HITS algorithm are proposed for increasing the precision (Li et al., 2002). In addition to this context based scoring of contents is also explored by various researchers (Bouramoul et al., 2011; Fernández et al., 2011; Gupta, 2012).

In the proposed SCOPAS model, the score of the page is computed by fusing the scoring of various segments of the page. The scores of segments are computed using a multidimensional approach. In the proposed model, the link texts present in the pages are utilized as one of the component while scoring.

In this Chapter, the survey of various motivational works carried out in the domain which are encompassed in the SCOPAS model is explored. The survey included web page segmentation, page classification, user profile handling and web page scoring. Different approaches in each of the domain are presented, with emphasize on the related technique proposed by this research work. In the case of web page segmentation, a hybrid segmentation technique incorporating page trees and densitometry is used in this research work. For classification of web page segments based on various structural parameters, the decision trees with C4.5 algorithm are used. For user profiling, this research work proposes a multimodal technique involving explicit and implicit profile data collection and FOAF based representation. For the purpose of content scoring, a multidimensional approach is proposed which involves a variable magnitude approach. These techniques are elaborated in the following Chapters of this thesis.

Chapter III: The SCOPAS – Segmentor

This research work aims towards building a web page model for scoring the contents with a multimodal approach, on the basis of personalization through user profiles. The scoring of web pages is achieved using a bottom-up approach in this research work. The page is initially split into various segments and each segment is scored individually using a multidimensional, variable magnitude methodology. The scores of the individual segments are fused together to compute the score of the page. This Chapter focuses on the segmentation component of the proposed research work. The segmentation component is termed as "SCOPAS Segmentor". The SCOPAS Segmentor is elaborated in this Chapter by giving details on the need for segmentation, the rules proposed for the segmentation, the type of segmentation technique and the mathematical representation of the segmentation technique proposed by this research work.

3.1 Need for Segmentation

The proposed SCOPAS model computes the score of a web page by dividing it into various segments. The reasons for carrying out the segmentation process systematically to score the web pages are discussed.

The segmentation plays an important role in scoring the relevancy of a web page, as the complete web page focusses not only on a single topic rather multiple topics. As it is known, in a page each part of the page might focus on different topic, if the score of the page is computed with a uniform measure, it might lead to un-optimized results. In such a scenario, the various sections of the page need to be given segment specific weight coefficient.

To achieve the same this work incorporates a segment classification technique which classifies the segments, based on various structural attributes. The segments belonging to different classes are given varying importance based on the structural semantics. In order to proceed with such a variable magnitude approach, the web page segmentation becomes a mandatory component while scoring the web page.

3.2 Segmentation Category

A detailed survey of web page segmentation is explored in Section 2.1 of the previous Chapter. The category of the segmentation proposed by this research, as shown in Fig 3.1 is elaborated, in this section.

The web page segmentation can be carried out either in the client side or in a remote location like the application server or the proxy server. The segmentation proposed by this research work is carried out in the server side. The rationale behind this decision is as follows: The web page segmentation alone is not the objective of this research work. The segmentation is a component in the overall scoring process. The scoring is carried out in the server side and hence the segmentation process cannot be ported to the client side. As the input to the scoring component is received from the Segmentor component, it has to be carried out in the server side.

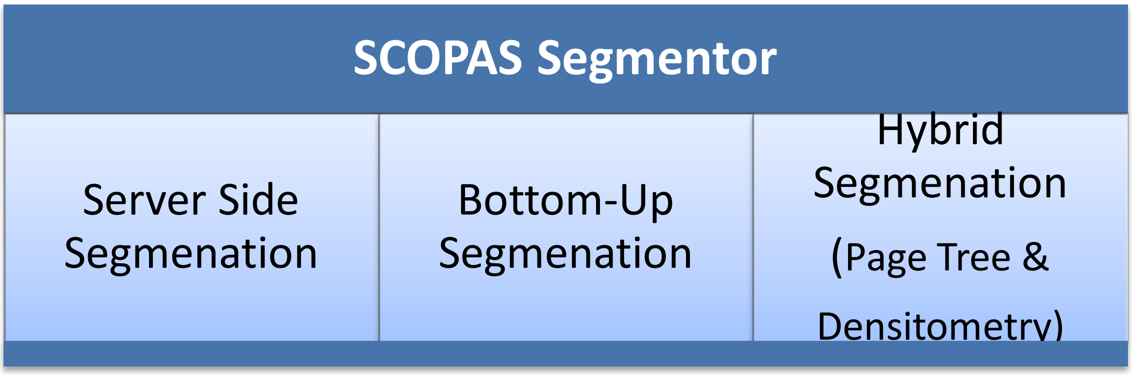


Fig 3.1: Segmentation Category

The segmentation approach adopted in this research work belongs to the bottom-up type. The segments are constructed in a bottom-up manner starting with the leaf nodes of the page tree. The nodes are added to the candidate segments incrementally to mark the segment boundary based on a threshold value.

The segmentation technique followed in this research work falls under the hybrid web page segmentation. The hybrid web page segmentation technique utilizes more than one type of segmentation techniques. In this research work, the hybrid segmentation technique incorporates the “web page tree technique” and “densitometry technique”. The details about the individual segmentation approaches and the benefits of the hybrid approach followed in this research work are explored in the following section.

3.3 Hybrid Segmentation

The components of the hybrid segmentation technique proposed by this research work are page tree and densitometry based segmentation techniques. This section illustrates the page tree and densitometry based segmentation techniques and the specific features of the proposed hybrid technique. The reasons for combining the page tree and densitometry techniques are also discussed in this section.

3.3.1 Page Tree Technique

The page tree based segmentation was introduced in detail in Section 2.1.3.1 of the previous Chapter. The page tree technique of web page segmentation works on the basis of Document Object Model (DOM) nodes of the web page. Rather than carrying out the segmentation on the text version of the web page, the page tree based segmentation technique operates on the in-memory tree representation of the page. By utilizing the tree representation, the segmentation technique becomes efficient as it can harness various tree related operations like “traversal”, “searching”. In this research work, the page tree technique is a component in the hybrid segmentation technique because of its ability to parse the Document Object Model tree of the web page. By parsing the Document Object Model tree of the web page, the nodes in the page tree are traversed starting with the leaf level nodes and segregating them into block level and non-block level nodes. The process is explained in detail through its mathematical representation in Section 3.5.

3.3.2 Densitometry Technique

The densitometry based web page segmentation technique was introduced in detail in Section 2.1.3.3 of the previous Chapter. The densitometry based web segmentation utilizes the density of text, for splitting the web page into various segments. The density of the text is computed by wrapping it in fixed width lines. The ratio of number of tokens to number of lines gives the text density. The tokens are chosen by their word boundary or the tag boundary. The change in density of the candidate segments is harnessed in marking the segment boundary. If the change in density is less than a specified threshold value, the candidate segments are merged, otherwise they are considered as individual segments. The densitometry technique is applied on the nodes identified using the page tree technique as illustrated in the previous section.

The complete segmentation process incorporating both the page tree and densitometry technique is explained in detail in Section 3.5. The advantages of attempting the web page segmentation using the hybrid approach are discussed in the next section.

3.3.3 Advantages of Hybrid Segmentation

The hybrid web segmentation technique proposed by this research work, inherits the advantages of component techniques, as illustrated in this section. The web pages can be processed at various layers of abstraction like the mark-up representation, in-memory tree representation. The markup representation is flat and text based. The in-memory representation is hierarchical and tree based. The page tree segmentation technique works at the in-memory representation layer and the densitometry technique operates at the markup layer. As the proposed model incorporates the page tree and densitometry techniques, the web page segmentation exploits both of these in-memory and text based representation.

The flat, text representation alone fall short in scenarios when the portions of the page are dynamically built. By using the combination of page tree and densitometry based techniques the benefits posed by accessing the web page at both these layers are harnessed in the proposed hybrid segmentation technique.

The proposed hybrid web page segmentation technique operates on two basic rules which are explained in the following section.

3.4 The Segmentation Rules

The web page segmentation technique proposed in this research work is based on two fundamental rules: the “Segment Mutual Exclusion rule” and “Comprehensive Segmentation rule”. These two rules of segmentation and the reasons for incorporating these two rules in the proposed segmentation technique are given in this section.

3.4.1 The Segment Mutual Exclusion Rule

Rule I: The Segments identified are non-overlapping

The first rule proposed by this research work for the Segmentor, to follow is the "Segment Mutual Exclusion". The segment mutual exclusion states that, the segments are identified so that they are non-overlapping. In other words, the resultant segments of the page should obey the mutual exclusion rule. The intersection of individual segments should produce the NULL value as shown in Eq. (3.1).

$$\forall\left(\mathrm{s_i},\ \mathrm{s_j}\right) \in P:\ \mathrm{s_i} \cap \mathrm{s_j} = \text{NULL}\ ;\ \mathrm{i, j}\ =\ \left(1\ ..\ \mathrm{k}\right)\ \ \&\quad i \neq j \tag{3.1}$$

In (3.1), the segments are indicated as $\mathrm{s_i}$,$\mathrm{s_j}$. The complete web page is indicated as "P". According to the Segment Mutual Exclusion rule, the boundaries of segment "$\mathrm{s_i}$" and "$\mathrm{s_j}$" are marked such that, they are non-overlapping. The values of indexes "i" and "j" are ranging from one to "k". In Eq.(3.1), "k" indicates the total number of segments in the web page. The condition $i \neq j$ is added so that the segment doesn't get compared with itself.

The "Segment Mutual Exclusion" rule makes sure that, there exist no overlapping boundaries for the segments of the web page. The reason for having non-overlapped boundaries is that, the score of the page is computed by fusing the segment scores. If some portions of the page appear in more than one segment then it would make the page score inconsistent.

3.4.2 The Comprehensive Segmentation Rule

Rule II: The Segmentation incorporates all parts of the web page.

The second rule proposed by this research work for the Segmentor, to follow is the "Comprehensive Segmentation Rule". This rule states that the segmentation should incorporate all parts of the web page. The rationale behind Comprehensive Segmentation rule is that, no part of the web page should be left by the Segmentor while building the segments. The Comprehensive Segmentation Rule is represented as shown in Eq. (3.2).

$$\boldsymbol{P} = \bigcup_{j=1}^{k} \boldsymbol{s}_j \tag{3.2}$$

In (3.2), the segments are denoted by "S_j" and the web page is denoted as "P". The index "j" ranges from one to "k". The value of "k" represents the total number of segments present in the web page.

The SCOPAS – Segmentor is designed in a manner that it follows both the "Segment Mutual Exclusion Rule" and "Comprehensive Segmentation rule". The combinatorial effect of both these rules makes sure that the segmentation is performed such that any part of the web page is allocated to only one segment and no part of the web page is left without allocation. The hybrid web page segmentation technique which is based on both of these rules is explained in the following section.

3.5 The Segmentation Process

The role of SCOPAS-Segmentor is to split the web page into various segments. During the segment building stage, the two rules derived in the previous section are followed, so that the segmentation covers the complete web page and the segment boundaries are non-overlapping. The steps involved in the proposed web segmentation technique are cleansing the page tree, identification of block level and non-block level nodes, applying the threshold condition, finding the text density in the case of block-level nodes, computing the change in the density slope and marking the segment boundary. Each of the above mentioned steps are described in detail in this section.

3.5.1 Page Tree Cleansing

The SCOPAS – Segmentor begins the segmentation process by parsing the web page tree. As stated in Section 3.3.3, web pages can be processed at different layers of abstraction. The proposed hybrid segmentation technique incorporates the page tree based segmentation and densitometry based segmentation. The segmentation process begins with the page tree based approach.

The first step in the proposed hybrid web page segmentation technique is cleansing the page tree of the web page. As stated in Section 2.1.1 of the previous Chapter, one of the challenges with the web page segmentation is the non-adherence of rules while building the web pages (Fauzi et al., 2009; Yesilada, 2011). There exist many common errors in the construction of web pages like missing the closing tag, improper nesting of the tags etc. As the segmentation technique works by constructing the Document Object Model tree of the web page, these errors need to be corrected before proceeding further for efficient handling of the page tree.

The source of the web page is analyzed with existing toolkits like “HTML Tidy” (Raggett, 1998) for cleansing the web page for errors like improper tag paring, errors in attributes specification etc. The cleansed version of the web page is used for further processing like block level and non-block level node identification which is explored in the following section.

3.5.2 Block Level and Non-Block Level Node Segregation

After cleansing the web page, the tree representation of the web page is built by the page tree parser. The nodes in the web page’s Document Object Model tree can be broadly divided into two major categories. They are “block level nodes” and “Non-Block level nodes”. The non-block level nodes are otherwise known as “inline nodes”.

The block level nodes shall contain other nodes as child nodes. The block level nodes generally associate a line break with them. The presence of block level nodes signifies a new block in the web page. On the other hand, the inline elements simply hold the contents. The Document Object Model tree of the web page is parsed and the block level and non-block level nodes are segregated as shown in Eq. (3.3).

$$\Omega = \{\eta, \xi\} \tag{3.3}$$

In Eq. (3.3), Ω represent the complete set of nodes containing the subsets of block level and non-block level nodes. The block level nodes are represented as η and the non-block level nodes are represented as ξ . The block level and non-block level nodes identified by this research work are parallel with the W3C standard block level and “in-line” element specification[3].

The rationale for segregating the block level and non-block level nodes is that, different procedures are involved in the segmentation of these two types of nodes.

[3] http://www.w3resource.com/html/HTML-block-level-and-inline-elements.php

3.5.3 The Segmentation Criteria

This section explores the segmentation criteria adopted for the block level and non-block level nodes. The non-block level nodes are considered in the decreasing order of their depth. The non-block level nodes are checked against a threshold value which can be customized based on different applications. If the node size is above the specified threshold then it is considered as a separate segment otherwise it has to be merged with the previous segment as shown in Eq. (3.4)

$$\forall_{i=1;\, j=1}^{n} \omega_j = \left\{ \begin{array}{cc} \xi_i & if\, \psi(\xi_i) \geq \delta_{NB}; i++, j++ \\ \omega_{j-1} \cup \xi_i & otherwise; i++ \end{array} \right\} \quad (3.4)$$

In Eq. (3.4), ω_j represent the segments built from the web page, ξ_i indicate the non-block level nodes. The function $\psi(\xi_i)$ is used to compute the size of the node. If the node size is beyond a threshold value δ_{NB} then the specific node is considered as a separate segment. Otherwise the node is merged with the predecessor segment. The value of index "i" ranges from one to "n" which indicates the number of non-block level nodes. The index "j" is used to denote the segments and its value starts with one. As the number of segments is not known before hand the terminating value of "j" is not specified in Eq. (3.4). When the threshold condition is satisfied the values of both the indexes "i" and "j" are incremented so that the next node starts with a new segment. If the threshold condition is not met, then only the value of "i" is incremented so that the next node would get added to the same segment.

For the block level nodes, the segmentation begins by checking the node's size against the threshold value δ_B as shown in Eq. (3.5).

$$\forall_{i=1;\, j=|\omega|+1}^{n} \omega_j = \left\{ \begin{array}{cc} \eta_i & if\, \psi(\eta_i) \geq \delta_B; i++, j++ \\ \omega_{j-1} \cup \eta_i & otherwise; i++ \end{array} \right\} \quad (3.5)$$

In Eq. (3.5), the variables used are equal to (3.4) except for the following changes. The threshold condition is modified from δ_{NB} to δ_B. Apart from this, the index "j" which is used to indicate the segments, increments its value by one to the number of segments already identified using Eq. (3.4).

In addition to this, another major change in approach with the block level nodes is that the densitometry based segmentation is applied as a subsequent step. The reason for including densitometry based segmentation for block level nodes is that they can hold other blocks inside them. The densitometry based segmentation component is explained in the following section.

3.5.4 The Densitometry for Block Level Nodes

The densitometry based web page segmentation was introduced in Section 2.1.4.3 of Chapter II. This section explains in detail about the proposed densitometry based segmentation component. In the previous section, the segmentation criteria got applied to both the block level and non-block level nodes. For the block level nodes, the densitometry technique is also applied because of the fact that the block level nodes shall contain other nodes in them. Due to this containership property of the block level nodes, their size tends to be larger than the non-block level nodes. In order to avoid having blocks with very large size, the densitometry technique is applied to the block-level nodes so that the level of granularity of the block can be maintained efficiently.

In this work, the text density of block level nodes has been computed and the change in density slope is utilized in marking the segment boundaries. The uniqueness of this approach is that it computes the density at the fine-grained segment level rather than at the complete page level. This facilitates efficient marking of segment boundaries.

In order to compute the text density, the source of the block level nodes above the specified threshold, is placed in lines of fixed length. The length of the line can be kept as a customizable key. The “terms” in the node contents are derived with the help of markup tags and word boundaries. Each “word” in the block level nodes is considered as a “term” and the presence of tags also mark the term boundary. The density is computed as the ratio of number of terms to the number of lines the node contents occupies, as shown in Eq. (3.6).

$$\forall_{j=1}^{\sigma} \varepsilon[\Re(\eta_i)]_j = \left\{ \frac{tc([\Re(\eta_i)]_j)}{\sigma([\Re(\eta_i)]_j)} \right\} \tag{3.6}$$

In Eq. (3.6) $\varepsilon[\Re(\eta_i)]_j$ denotes the density of the jth line of ith block level node η_i. The function $\Re(\eta_i)$ is the content fetcher used to get the content of the specified line. The function tc is used to compute the term count. The σ is used to compute the number of fixed length lines occupied. The ratio of value returned by tc to σ is computed as the text density.

3.5.4.1 Segment Slope Computation

The content density of the candidate segments are computed as shown in (3.6). The slope value between the content densities is the factor which decides the segment boundary. The segment slope is computed as the ratio of difference between the content densities of successive candidate segments to maximum content density among the candidate segments as shown in Eq. (3.7).

$$\Delta\varepsilon([\Re(\eta_i)]_j,[\Re(\eta_i)]_{j+1}) = \frac{\left|\varepsilon\{[\Re(\eta_i)]_j\} - \varepsilon\{[\Re(\eta_i)]_{j+1}\}\right|}{\max(\varepsilon([\Re(\eta_i)]_j,[\Re(\eta_i)]_{j+1}))} \quad (3.7)$$

In Eq. (3.7), the $\Delta\varepsilon([\Re(\eta_i)]_j,[\Re(\eta_i)]_{j+1})$ indicates the slope value between the candidate segments "j" and "j+1". The $\max(\varepsilon\{[\Re(\eta_i)]_j,[\Re(\eta_i)]_{j+1}))$ returns the bigger value among the content densities of the candidate segments.

This slope valued computed using Eq. (3.7) is utilized in the following section to decide the segment boundary.

3.5.4.2 Marking the Segment Boundary

The slope value computed in the previous section is compared against a threshold value to decide the segment boundary.

If the calculated slope value is greater than a threshold value λ then $[\Re(\eta_i)]_j$ and $[\Re(\eta_i)]_{j+1}$ are considered separate blocks ω', ω'' otherwise they are fused together (ω) as shown in Eq. (3.8).

$$\begin{Bmatrix} \omega = [\Re(\eta_i)]_j \cup [\Re(\eta_i)]_{j+1} & if\, \Delta\varepsilon\{[\Re(\eta_i)]_j, [\Re(\eta_i)]_{j+1}) < \lambda \\ \omega' = [\Re(\eta_i)]_j & \\ & otherwise \\ \omega'' = [\Re(\eta_i)]_{j+1} & \end{Bmatrix} \tag{3.8}$$

The value of the threshold limit λ can be customized based on different applications. Smaller the value of λ , the granularity of the segments would also be finer. This threshold value can be utilized to adjust the size of the segments built by the proposed hybrid segmentation technique. At the end of this process, the complete set of segments of the web page is built by merging the segment identified using Eq. (3.4) and Eq. (3.8) as shown in Eq. (3.9).

$$\Omega' = \{\omega_1, \omega_2, \omega_3 .. \omega_n\} \tag{3.9}$$

In Eq. (3.9), ω_i indicates the segments identified. The Ω' is the set holding the complete collection of segments.

3.5.5 The Segment Pool

The segments built by the proposed hybrid segmentation technique are represented using “Segment Pool”. The segment pool is a structure which facilitates efficient accessing of the built segments which are built.

The specific segments from the segment pool can be accessed by providing the “segment index”. The segment index is a unique number assigned to each of the segments built by the above specified process, so that they can be accessed faster and in an efficient manner. The segment pool has two major components. They are “Live Segment Pool” and “Offline Segment Pool”. The live segment pool holds the Document Object Model representation of the segment for using them immediately.

The offline segment pool is utilized for storing the segments in the secondary memory for future use. The offline segment pool stores the web page segments in a “Serialized” format which is the process of converting the in-memory representation of data into a format which can be stored in a file. When the segments need to be processed, they can be accessed from the secondary memory and un-serialized for further processing. The segment pool functions as the warehouse of segments using the proposed hybrid web page segmentation technique. As such the segment pool can be utilized in other applications like mobile rendering, segment based search engine repositories etc. which makes the handling of web pages simple and efficient.

This Chapter has elaborated in detail about the web page segmentation component “SCOPAS – Segmentor” of this research work. The Segmentor component is explained in detail from various dimensions. This Chapter started with explaining the necessity of incorporating the segmentation in the proposed content scoring model. The details on the type of segmentation technique proposed by this research work are also explored by focusing on the component techniques of the proposed hybrid segmentation. The “Segment Mutual Exclusion Rule” and the “Comprehensive Segmentation Rule” are illustrated with their usage. The segmentation process is elaborated in detail and the segmentation criteria for block level and non-block level nodes are illustrated in this Chapter. The live and offline components of segment pool which are utilized for immediate access and for the future usage respectively, are explored. As the result of segmentation process, the segment pool is filled with the segments of the web page which need to be evaluated individually using a multimodal, variable magnitude approach. In order to have differential weight for various segments, the classification of segments is carried out using a list of features. The classification of segments is explained in detail in the following Chapter.

Chapter IV: The ClaPS – Segment Classifier

This Chapter illustrates the Segment Classification component of the SCOPAS model, which is termed as ClaPS (Classification of Page Segments). The method proposed for the classification of various segments, based on their structural characteristics is explained in detail in this Chapter. This Chapter is organized as follows: The need for classifying the segments are listed out in Section 4.1. The category of classification followed in this research work is explored in Section 4.2.The various classes of segments identified by this research work are elaborated in Section 4.3. Various features like “text ratio”, “link ratio” which are proposed in this research work for the purpose of classifying the segments are defined in Section 4.4. The decision tree based methodology adopted for the classification is narrated in Section 4.5.

4.1 Need for Segment Classification

This section explores the rationale for incorporating a segment classification technique in the content scoring process which is the main objective of this research work.

The web page contains segments which exhibits an array of structural properties. Some segments may be purely text based, some may contain only images and some of the segments may have hyperlinks as their major component. During the content scoring phase, if all of the segments are given equal weights then it indicates that the structural diversity exposed by the web pages is not harnessed properly. For example, if a query term appearing in an image oriented segment and a query term appearing in a simple text segment are given uniform weights then it will lead to ignoring of the structural semantics.

In order to capture the structural semantics embedded in the web pages, a segment classification technique based on structural features extracted from the segments has been proposed. This classification of segments into various categories enables the content scoring component to incorporate the “segment class weight” while fusing the scores of segments to compute the score of the web page. The classification category followed in this research work is discussed in the following section.

4.2 Classification Category

A survey of various classification techniques was provided in Section 2.2. This section highlights the classification category and the approaches adopted in this research work which is highlighted in Fig 4.1.

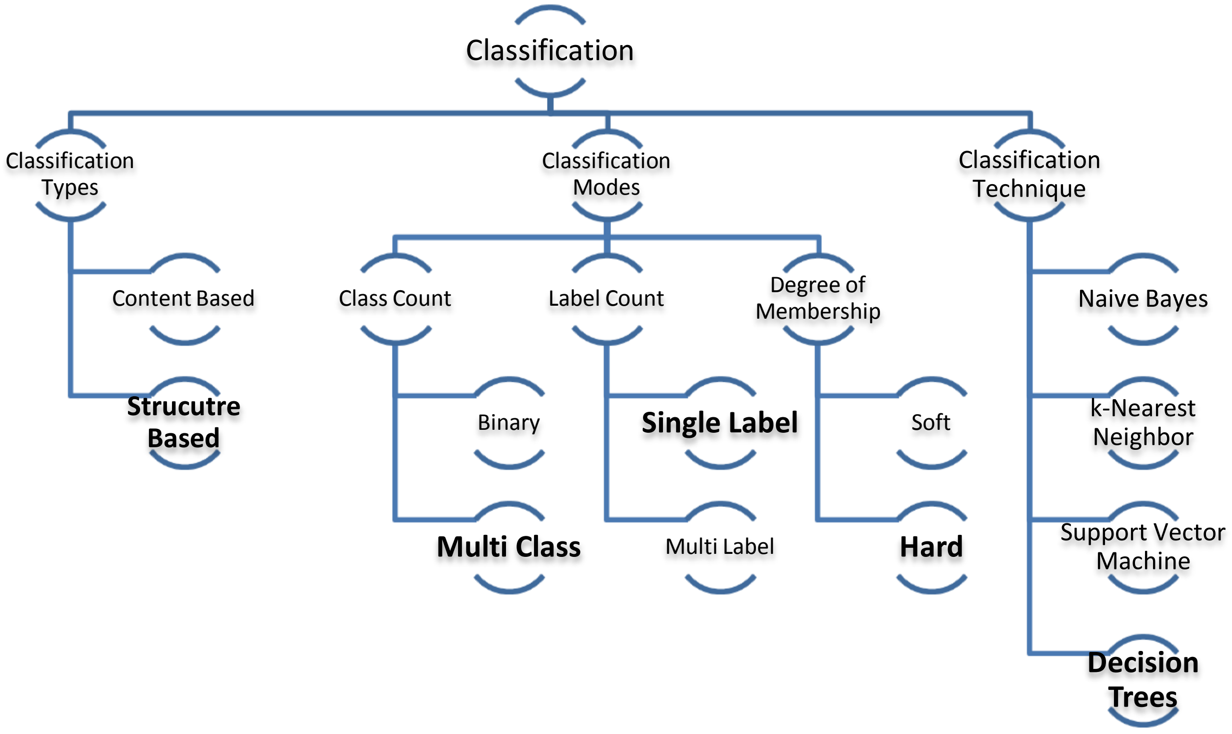


Fig 4.1: Classification Category

The classification type followed in this research work belongs to the “Structure Based Classification”. The rationale for choosing this type is that, the segments need to be given differential weights based on their structural semantics. Hence the classification is carried out using various structure based features.

This research work introduces five different types of segment classes. Since the count of segments is more than two, the proposed classification approach falls under the “multi class” category.

The segments are classified in such a manner that any given segment shall be assigned only one class at an instance. Hence the classification method of ClaPS is categorized as “Single Label” classification. When a segment is assigned a class label, it is assigned with complete membership. Since the “degree of membership” of a segment to a class is not included, the proposed method belongs to "Hard classification" category. The reason for not including the degree of membership is that, if a segment is not firmly classified to a specific type then the content scorer cannot incorporate the corresponding segment class weight in the relevance scoring procedure.

4.3 ClaPS Segment Classes

The proposed segment classification technique (ClaPS) incorporates five different classes of segments as shown in Table 4.1. The classes of segments are based on the web page entities i.e. the elements present in the segments.

Segment Type	**Description**
Simple Text Segment	The segments which majorly consists of descriptive text rather than other web page entities.
Navigation Segment	The segments which majorly consists of hyperlinks rather than other web page entities.
Image Segment	The Segments which hold images as their major part.
Head Segment	The Segment which is a header.
A/V Segment	The Segment which holds audio / videqo contents as their major part.

Table 4.1: Segment Classes

The segment classes are assigned such that, it would cover various structural characteristics exhibited by web pages.

Definition 4.1:

Simple Text Segment: A segment is classified as Simple Text Segment if and only if the simple text tokens overweigh all other tokens present in the segment under consideration.

Definition 4.2:

Navigation Segment: A segment is classified as Navigation Segment if and only if the tokens in the hypertext overweigh all other tokens present in the segment under consideration.

Definition 4.3:

Image Segment: A segment is classified as Image Segment if and only if the segment is filled with image(s), comparing with all other types of tokens present in the segment under consideration.

Definition 4.4:

Head Segment: A segment is classified as a Head Segment if and only if the tokens in the heading element overweigh all other tokens present in the segment under consideration.

Definition 4.5:

A/V Segment: A segment is classified as Audio Video segment if and only if the segment is filled with audio / video, comparing all other tokens present in the segment under consideration.

The segment class weight which is assigned during the content scoring, based on the class of segment in which the query term is found, is set as a variable parameter. The segment class weight is assigned in an incremental manner in the following order: Simple Text Segment, Navigation Segment, Image Segment, Head Segment and A/V Segment. The rationale for this approach is that if the query term is found with an audio/video component then it is assigned more score than the other segments. Similarly the head segment gets more preference than the other segments like Simple Text, Navigation Segment etc. The segment class weight scoring is explained further in Section 5.2.4.

The classification of segments into any one of the above specified segments is done with the help of various parameters extracted from the characteristics of the segments, which is explained in the following section.

4.4 The ClaPS Features

This section defines five different features, extracted from the web page segments, for the purpose of classifying the segments into the classes which are defined in the previous section. The five features computed by ClaPS are illustrated in Fig 4.2.

ClaPS - Segment Features

Text Ratio

Link Ratio

Image Count

Head Ratio

Object Count

Fig 4.2: The ClaPS Features

The Table 4.2 introduces these five features with a short description.

Feature	Description
Text Ratio	A feature based on the quantity of simple text in the segment
Link Ratio	A feature based on the quantity of hyperlink text in the segment
Image Count	Number of images in the segment
Head Ratio	A feature based on the quantity of heading text in the segment
Object Count	The number of Object and Embed elements in the segment

Table 4.2: Segment Features

4.4.1 Text Ratio

This section defines the "text ratio" feature which is based on the quantity of text content present in the web page segment.

Definition 4.6:

Text ratio feature is defined as the ratio of the number of simple text tokens to total number of tokens in the segments.

$$TR(\omega) = \frac{count(SimpleText(\omega))}{count(Tokens(\omega))} \qquad (4.1)$$

The text ratio computation is shown in Eq.(4.1). The text ratio is a real value computed by dividing the count of simple text token by total number of tokens in the segment. The simple text token refers to content without additional structural features. The total number of tokens incorporates both text and non-text tokens.

4.4.2 Link Ratio

This section defines the "link ratio" feature which is based on the quantity of hyperlink text content present in the web page segment.

Definition 4.7:

Link Ratio is defined as the ratio of the number of tokens in hypertext anchors to total number of tokens in the segment.

$$LR(\omega) = \frac{count(AnchorText(\omega))}{count(Tokens(\omega))} \qquad (4.2)$$

The link ratio computation is shown in Eq. (4.2). The link ratio is a real value computed by dividing the number of tokens in the hypertext anchors by the total number of tokens in the segment.

4.4.3 Image Count

This section defines the "image count" feature which is extracted based on the quantity of image elements present in the web page segment.

Definition 4.8:

Image Count is defined as the total number of image elements present in the segment under consideration.

$$IC(\omega) = Count(img(\omega)) \qquad (4.3)$$

4.4.4 Head Ratio

This section defines the "Head Ratio" feature which is extracted based on amount text content present in the head elements.

Definition 4.9:

Head Ratio feature is defined as the ratio of number of tokens in heading elements to total number of tokens in the segments.

$$HR(\omega) = \frac{count(HeadText(\omega))}{count(Tokens(\omega))} \qquad (4.4)$$

The Head Ratio computation is shown in Eq. (4.4). It is computed by dividing the number of tokens in the head elements by the total number of tokens in the segment.

4.4.5 Object Count

This section defines the "object count" feature which is extracted based on the number of multimedia elements present in the web page segment.

Definition V:

The Object count is defined as the total number of "embed" and "object" elements present in the segment under consideration.

$$EC(\omega) = Count(embed(\omega)) \qquad (4.5)$$

The "embed" or "object" elements are used to incorporate multimedia elements in the web page.

Based on these features the segments in the segment pool are classified into any one of the classes specified in Table 4.1. The actual classification process has been carried out with the "Decision Trees" with the help of C4.5 algorithm.

4.5 The Classification Technique

The web page segment classification approach of this research work is based on the Decision Trees. A survey on decision trees for the classification purpose was presented in Section 2.2.3.4. This section explores the decision tree based classification approach to classify the web page segments into classes listed in Section 4.3 based on the features described in Section 4.4.

The Decision tree is used in this work due to its ability to perform classification in an enhanced manner for datasets with limited number of features(Quinlan, 1993). In this work the count of features using which the classification is carried out is five. This manageable count of features favors the decision trees. In addition to this, the decision trees can handle missing values in the feature file (a file holding all the attribute values). This research work utilizes the C4.5 (Quinlan, 1993) classification algorithm for recursively portioning the instance space and assigning the class labels for the web page segments.

This classification of web page segments into the predefined classes on the basis of features introduced in this research work, is carried out with the help of Weka (Hall et al., 2009) machine learning software workbench. A formal introduction to the Weka machine learning software workbench was provided in Section 2.2.4.

The Weka tool requires features to be given in the form of Attribute Relation File Format (ARFF). The format of the ARFF is shown in Fig 4.3.

TR	**LR**	**IC**	**HR**	**OC**	**Class**
FV$_1$	FV$_2$	FV$_3$	FV$_4$	FV$_5$	[STS /IS / HS / NS / AVS]

Fig 4.3: ARFF Entry Format

In Fig 4.3, first five columns of the top row indicate features Text Ratio, Link Ratio, Image Count, Head Ratio and Object Count. The last column in the top row is the class to which the segments need to be assigned. The FV$_i$ (Feature Value) in the bottom row of the Fig 4.3 indicate the corresponding Feature Values.

The last column in the bottom row will hold any one of the class identifiers. In Fig 4.3, STS stands for Simple Text Segment, IS stands for Image Segment, HS stands for Head Segment, NS stands for Navigation Segment and AVS stands for the Audio Video Segment.

The J48 decision tree classifier of the Weka tool, which is the open source variant of the C4.5 algorithm, is utilized in performing the classification. In order to perform the classification tasks, the features are extracted from the segments of the segment pool. As stated earlier in Section 3.5.5 the segment pool consist of two different components. They are “Live Segment Pool” and “Offline Segment Pool”. The features are extracted from the live segment pool directly.

In the case of “Offline Segment Pool”, the segments are fetched from the secondary storage in the serialized format. The segments are un-serialized before performing the feature extraction process. The feature extractor scans each segment and populates the values in the corresponding fields.

As decision tree is a machine learning tool, it has to be trained before labelling the novel data. Hence the Attribute Relation File Format populated with the features extracted from segments from the segment pool is passed to the decision tree learner. The learning process builds the decision tree based on the training dataset. The decision tree built is pruned further for efficient processing.

As the result of training process, the decision model is built. This decision model is utilized for classifying the novel web page segments into the classes proposed in this work. The detailed empirical validation of the proposed web page segment classification model is provided in the Chapter VII.

At the content scoring stage, the segment is classified using the decision model built during the learning stage. The classifier returns the segment class as output, which is used to incorporate the segment class weight by the page scorer component.

This Chapter explored the web page segment classification (ClaPS) component of the SCOPAS model. This chapter explored the necessity of performing the classification in the overall page scoring process. The classification method followed in this research work belongs to the multiclass, structure based, single label, hard classification category. This Chapter defined five different segment classes on the basis of their properties. Five features of segments are also defined in this research work for carrying out the segment classification. This works utilizes the decision tree based classification technique using J48 classifier of the Weka machine learning framework.

The web page scoring component, “SCOPAS Evaluator” which computes the score of the page by fusing the segment scores is explored in detail in the following Chapter.

Chapter V: The SCOPAS – Evaluator

This Chapter explains the content scoring component, "SCOPAS Evaluator". The primary objective of this research work is to provide a web content scoring mechanism with the help of segmentation, segment classification and personalization which is achieved by modeling the web page. The technique proposed in this work for scoring the web pages with respect to the context of the user's information requirement is carried out through a multimodal, variable magnitude approach. This has been achieved by introducing various weight coefficients for evaluating the segment's relevancy through "search queries" and "user profiles".

The user profiler component which is used to build the profile of the user in an incremental manner is explored in detail in Chapter VI. The reason for explaining the "SCOPAS Evaluator" before "SCOPAS Profiler" is that the profile building process iteratively uses the components of SCOPAS Evaluator for extracting the user profile terms. Hence the SCOPAS profiler component is dealt, following the SCOPAS Evaluator component.

This Chapter is organized as follows: the role of SCOPAS –Evaluator is explained in Section 5.1. The Section 5.2 illustrates the segment scorer component MUSEUM (Multidimensional SEgment EvalUation Method) which measures the relevancy of the segment by incorporating various intra-segment features in a multimodal manner with the help of six "weight coefficients". The page score computation which is done by fusing the scores of individual segments is described in Section 5.3. The section 5.4 incorporates the "segment class weight" which is computed with the help of the web page segment classifier component (ClaPS). The page score is enriched by adopting "segment class weight" which is a process introduced in this research for boosting the score, based on structural characteristics exhibited by different components of the web page.

5.1 The Role of SCOPAS Evaluator

The SCOPAS Evaluator computes the score of the page by splitting the page into various segments. For the purpose of segmenting the page, a hybrid web page segmentation technique incorporating "Page Trees" and "Densitometry". The page segmentor component is explained in detail in Chapter III.

As the result of segmenting the pages, the segment pool is filled with the segments derived from the webpage. The segments shall be directly used through “Live Segment Pool” or accessed from the “Offline Segment Pool” which requires un-serializing before further utilization. The segments in the segment pool are fetched by the classifier component (ClaPS) for performing a structure based classification, with the help of decision trees, as illustrated in Fig 1.2 of Chapter I.

The role of SCOPAS Evaluator is to fetch the segments from the segment pool and pass them to the “Segment Scorer” which is termed as “MUSEUM” (Multidimensional SEgment EvalUation Method). The scores of the segments are computed by MUSEUM with the help of various weight coefficients.

The SCOPAS Evaluator fetches the user profile specific keywords from the “SCOPAS Profiler”. The profile specific terms are incorporated into the page scoring process, to facilitate the user specific scoring of web pages.

The SCOPAS Evaluator computes the page score by fusing the scores returned by the MUSEUM and incorporating the “segment class weight” which indicates the segment class specific coefficient to be multiplied with the page score.

5.2 The Segment Scorer – MUSEUM

As stated in the previous section, the “SCOPAS – Evaluator” computes the relevance score of the web page, by fusing the individual segment scores.

The scores of the segments are computed by the Segment Scorer termed “MUSEUM” (Multidimensional SEgment EvalUation Method). The web page segment scorer – MUSEUM computes the score by a multimodal manner as illustrated in Fig 5.1.

5.2.1 The MUSEUM Coefficients

The relevance score of a web page segment $\omega(\mathrm{s_i})$ is computed with six different weight coefficients as represented in Eq. (5.1)

$$\omega(\mathrm{s_i}) = (E, M, L, R, F, V) \tag{5.1}$$

Where

E = Theme Weight Coefficient

M = Image Weight Coefficient

L = Link Weight Coefficient

R = Profile Weight Coefficient

F = Freshness weight Coefficient

V = Visual Weight Coefficient

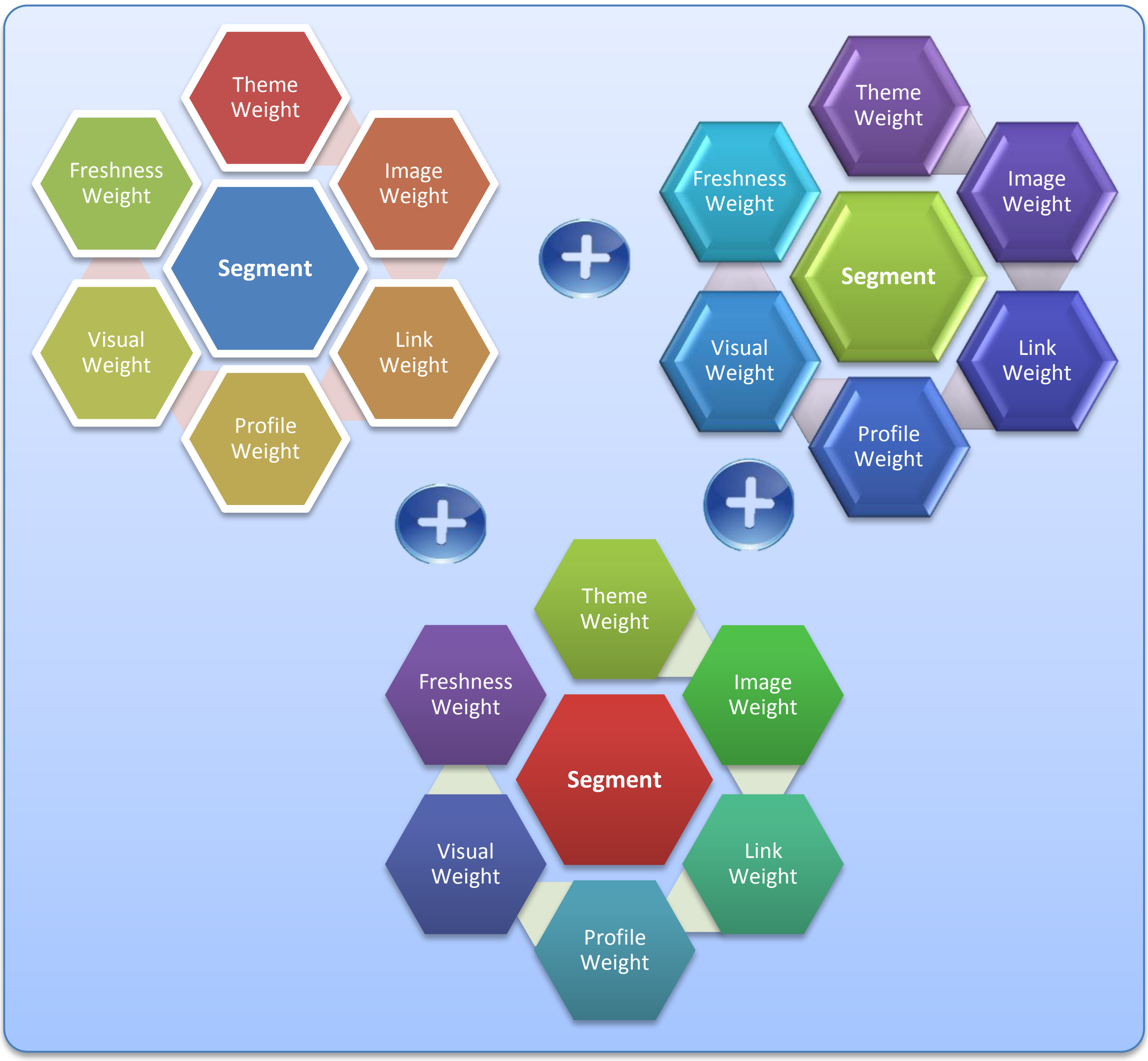


Fig 5.1: The MUSEUM – Segment Scorer

These weight coefficients are designed in such a manner that the web page segment scorer encompasses all intra segment components. The core idea of introducing six different coefficients is that, the terms appearing in the body-text of the web page should be treated differently from the terms appearing in the hypertext anchors or image attributes. Similarly the text with plain formats should be treated differently from the text appearing with the special visual feature. This variable magnitude approach proposed in this work for computing the web page segment score is further enhanced by the incorporation of "segment class weight". The weight coefficients are computed by analyzing the specific intra segment structural features using the user's current information requirement context which is represented by the combination of query terms entered by the users and the profile keywords fetched from the "SCOPAS Profiler". Each of the weight coefficients is computed in an independent manner as described in this section.

5.2.1.1 Theme Weight Coefficient

The multidimensional segment scoring model MUSEUM incorporates the "theme" of the web page as one of the weight coefficients. The theme of the web pages is captured with the help of "title" of the web page (Craven, 2003; Dawson, 2004; Noruzi, 2007). The incorporation of title is proven to improve the retrieval efficiency by earlier studies (Ogilvie and Callan, 2003; Xue et al., 2007). The MUSEUM incorporates the theme of the web page through title in the segment scoring procedure, as one of the weight coefficients.

The theme weight coefficient is computed by measuring the frequency of the terms appearing in the segment, which are part of the title of the web page. If the segment consists of terms from the page title, then the theme weight coefficient is assigned as the number of terms matching between the title and segment terms. The theme weight coefficient $\omega_E\left(\mathrm{s_i}\right)$ is computed as shown in Eq.(5.2).

$$\omega_E\left(\mathrm{s_i}\right)=\begin{cases}\forall\ \mathrm{e_j}\in\ \mathrm{E},\ \mathrm{q_k}\in\ \mathrm{Q},\ 0\le \mathrm{j}<\left|E\right|,\ 0\le \mathrm{k}<\left|Q\right|\ :\ |\mathrm{s_i}\cap e_\mathrm{j}\,|+|\mathrm{syn(s_i)}\cap \mathrm{syn(e_j)}\,|\,/2 & \mathit{if}\,|\mathrm{syn(s_i)}\cap \mathrm{syn(e_i)}\,|>0; \\ 0 & \mathit{otherwise}\end{cases} \tag{5.2}$$

Where

E = The set of terms present in the web page title

$\mathrm{e_j}$ = Specific terms in the title

Q = Set of terms in the query

$\mathrm{q_k}$ = Individual elements of the query-terms

The index “j” is used to navigate through the set “E”. The value of “j” ranges from zero to |E| -1. The index “k” is used to navigate through the set “Q”. The value of “k” ranges from zero to |Q| -1.

The function “syn()” is used to fetch the synonyms of the terms. The synonyms component is incorporated so that the linguistic variations of the terms are considered for evaluation, when the term is not directly present in the segment.

The theme weight coefficient is computed as the number of terms in the segments matching the terms in the title. In order to boost the direct presence of the title terms in the segment, the weight coefficient is incremented by one for direct presence and it is incremented by 0.5 for the synonyms of the terms, from the title. If neither the direct presence nor the synonyms of the title terms are detected, then the theme weight coefficient is set to zero.

5.2.1.2 Image Weight Coefficient

The presence of image related to the query term, signifies the relevance of score of the web page, which is explored by earlier studies (Hu, 2008). The MUSEUM segment scorer incorporates the image weight coefficient as a component in its six dimensional segment scoring procedure.

The description about the images is provided in the metadata of the images in the web pages to handle the scenarios when the image is not properly loaded in the client machine (Antonacopoulos et al., 2001).[4] This metadata of the image is harnessed by the MUSEUM segment scorer to compute the image weight coefficient. The image weight efficient $\omega_M\left(s_{\mathrm{i}}\right)$ is computed as shown in Eq.(5.3).

$$\omega_M\left(s_{\mathrm{i}}\right)=\begin{cases}\forall\ \mathrm{m}_{\mathrm{j}}\in \mathrm{M},\ \mathrm{q}_k\in\ \mathrm{Q},0\le \mathrm{j}<|M|,\ 0\le \mathrm{k}<|Q|\ :\ |\mathrm{q}_{\mathrm{k}}\cap(m_{\mathrm{j}})|+|\mathrm{syn}(\mathrm{q}_k)\cap(\mathrm{syn}(\mathrm{m}_j))|/2 & \mathit{if}\,|\mathrm{syn}(\mathrm{q}_{\mathrm{k}})\cap(\mathrm{syn}(\mathrm{m}_{\mathrm{j}}))|>0; \\ 0 & \mathit{otherwise}\end{cases} \tag{5.3}$$

Where

M = Set of meta-data text terms

$\mathrm{m_j}$ = Individual terms of the metadata text

[4] The metadata associated with the image shall be fetched from attributes like “alt”.

Q = Set of terms in the query

q_k = Individual elements of the query-terms

The index "j" is used to navigate through the set M. The value of "j" ranges from zero to the size of the set |M|-1. The index "k" is used to navigate through the set "Q". The value of "k" ranges from zero to |Q| -1.

The "image weight coefficient" $\omega_M\left(s_i\right)$ is computed as the frequency of terms from the query "Q" appearing in the image metadata text set "M". The synonyms of the terms are computed with the help of "syn ()" function. The synonyms of the terms are given half the weight as the direct presence of the term "q_k".

The image weight coefficient is assigned the value of zero in the following two scenarios: for the segments with no images; for the segments without query terms or their synonyms in the image metadata.

5.2.1.3 Link Weight Coefficient

The leading factor which distinguishes flat text from the web pages is the presence of hypertext anchors in the web pages with links to other web contents. The presence of query-terms in the hypertext anchors gets more significance than the query-terms in the simple text (Chauhan and Sharma, 2007; Pant, 2003).

The MUSEUM segment scorer incorporates the "link weight coefficient" as one of the component in the variable magnitude segment scoring procedure. The link weight coefficient $\omega_L\left(s_i\right)$ is computed as shown in Eq. (5.4).

$$\omega_L\left(s_i\right)=\begin{Bmatrix}\forall\, l_j \in L,\ q_k \in Q, 0 \le j<|L|,\ 0 \le k<|Q| \ :\ |q_k \cap l_j|+|\mathrm{syn}(q_k)\cap \mathrm{syn}(l_j)|/2 & \mathit{if}\, |\mathrm{syn}(q_i)\cap \mathrm{syn}(l_i)|>0; \\ 0 & \mathit{otherwise}\end{Bmatrix} \tag{5.4}$$

In (5.4), the set of terms in hypertext anchors of the web page segment is represented as "L". The individual elements of the hypertext anchor-terms are indicated as "l_j". The query term set is represented as "Q" as explained in the previous section.

The index "j" is used to iterate through the hypertext anchor-terms set "L". The value of "j" ranges from zero to size of the set |L|-1. Similarly the index "k" is used to navigate through the query-terms set.

The "link weight coefficient" $\omega_L\left(s_i\right)$ is computed as the frequency of query-terms appearing in the hypertext anchor-text, including the synonyms. The synonyms of the terms are computed by the "syn()" function. In parallel with the previous weight coefficients, the synonyms of the terms are given half the weight comparing with the direct presence of the terms.

The link weight coefficient is assigned the value of zero in the following the two scenarios: for segments with no hyperlinks; for segments without query-terms or their synonyms in the hypertext anchors.

5.2.1.4 Profile Weight Coefficient

The MUSEUM scorer incorporates user specific information requirement in the segment scoring procedure. The user specific scoring is performed with the help of profile terms which is amalgamated in an incremental manner by the "SCOPAS Profiler". The SCOPAS Profiler is elaborated in detail in Chapter VI.

The incorporation of personalization improves the overall retrieval experience of the web users (Matthijs and Radlinski, 2011). Hence the MUSEUM scorer incorporates the profile weight coefficient $\omega_R\left(s_i\right)$ as shown in Eq.(5.5).

$$\omega_R\left(s_i\right)=\begin{cases}\forall\ \mathrm{r}_j\in\mathrm{R}, 0\le \mathrm{j}<\left|R\right|:\ |\mathrm{s}_i\cap(r_j)|+|\mathrm{s}_i\cap\mathrm{syn}((\mathrm{r}_j))|/2 & \mathit{if}\,|\mathrm{syn}(\mathrm{s}_i)\cap\mathrm{syn}((\mathrm{r}_j))|>0;\\ 0 & \mathit{otherwise}\end{cases} \tag{5.5}$$

Where

R = Profile-bag which consisting of profile terms of the user
r_j = Individual terms in the profile bag

The index "j" is used to iterate through the profile bag "R". The value of "j" ranges from zero to the size of the profile bag |R| -1. The segment for which the score is computed is indicated as "s_i" in Eq. (5.5).

For the direct presence of the profile terms in the segment, the score is incremented by one and for the synonyms of the profile terms it is incremented by 0.5. The rationale for following this approach is same as explained in Section 5.2.1.1.

The profile weight coefficient is set to zero, if none of the terms from the profile or their synonyms are present in the segment under consideration. The profile weight coefficient hence would return different scores for different users for the same segment. This facilitates the evaluation of segments in terms of user's specific information requirement context by supporting the query terms with the profile-bag terms.

The profile-bag terms are derived using a multidimensional approach by using a semantic representation technique which is explored further in the next Chapter. The profile keywords are gathered in a hybrid manner by receiving the "explicit terms" from the users directly and deriving the "extended profile terms" from the resources specified by the users.

5.2.1.5 Freshness Weight Coefficient

This section explores another weight coefficient proposed by the MUSEUM scorer, "freshness weight coefficient". The temporal dimensions of information retrieval have been studied by researchers for enhancing the retrieval process. (Anagnostopoulos et al., 2010; Bar-Ilan, 2002; Cambazoglu et al., 2007; Dai and Davison, 2010; Sato et al., 2003).

The core idea of incorporating a weight coefficient for freshness is that the fresh contents related to the query terms should be given higher priority than the stale contents. The freshness is considered as an important metric in the web scale retrieval systems (Na Dai and Brian D. Davison.2010). The freshness weight coefficient $\omega_{\mathrm{F}}(\mathrm{s}_{\mathrm{i}})$ is computed as the sum of two different components as shown in Eq. (5.6).

$$\omega_{\mathrm{F}}(\mathrm{s}_{\mathrm{i}}) = \omega_{\mathrm{FA}}(\mathrm{s}_{\mathrm{i}}) + \omega_{\mathrm{Fs}}(\mathrm{s}_{\mathrm{i}}) \tag{5.6}$$

In (5.6), the freshness weight coefficient $\omega_{\mathrm{F}}(\mathrm{s_i})$ is computed as the sum of, “actual freshness weight” $\omega_{\mathrm{FA}}(\mathrm{s_i})$ and the “synonym freshness weight” $\omega_{\mathrm{Fs}}(\mathrm{s_i})$. The actual freshness weight coefficient $\omega_{\mathrm{FA}}(\mathrm{s_i})$ is computed as shown in Eq. (5.7).

$$\omega_{\mathrm{FA}}(\mathrm{s_i}) = \begin{Bmatrix} \forall\ \mathrm{t_j} \in \mathrm{T},\ \mathrm{q_k} \in \mathrm{Q},\ 0 \le \mathrm{j} < |T|,\ 0 \le \mathrm{k} < |Q|\ :\ |\mathrm{q}_k \cap s_{\mathrm{i}}| & if(\mathrm{t_{(0..j-1)}} \cap \mathrm{q_k}) = NULL \\ 0 & otherwise \end{Bmatrix} \tag{5.7}$$

Where

T = Temporal snapshot set of the segments
t_j = Individual snapshots of the set “T”
Q = Set of terms in the query
q_k = Individual elements of the query-terms

In Eq. (5.7), the freshness weight coefficient is computed by considering the previous snapshots of the segment in the temporal dimension. The weight is assigned as the frequency of query terms appearing in the segment freshly. If the terms in the query are present only in the current snapshot of the segment and not in the earlier temporal versions, then the “actual freshness weight” coefficient is computed by measuring the frequency of term appearances. If the above condition is not met then the coefficient $\omega_{\mathrm{FA}}(\mathrm{s_i})$ is assigned a value of zero.

The “Synonym freshness weight” coefficient $\omega_{\mathrm{Fs}}(\mathrm{s_i})$ is computed as shown in Eq. (5.8).

$$\omega_{\mathrm{FS}}(s_{\mathrm{i}}) = \begin{Bmatrix} \forall\ \mathrm{t_j} \in \mathrm{T},\ \mathrm{q_k} \in \mathrm{Q},\ 0 \le \mathrm{j} < |T|,\ 0 \le \mathrm{k} < |Q|\ :\ |\mathrm{syn(q_k)} \cap syn(s_{\mathrm{i}})|/2 & if(\mathrm{t_{(0..j-1)}} \cap \mathrm{syn(q_k)}) = NULL \\ 0 & otherwise \end{Bmatrix} \tag{5.8}$$

In Eq. (5.7) and Eq. (5.8), the temporal snapshot set of the segment is indicated as “T”. The collection of snapshot in the temporal dimensions is termed as “Page Evolution Track”. The page evolution track is built by the web crawlers visiting the page at frequent interval of time which can be decided by the page’s dynamism frequency. The value of the index "j" ranges from zero to the size of the set |T| -1. The query terms are represented as "Q" and the index "k" is used to navigate through the set "Q". In Eq. (5.8), the function "syn()" is used to compute the synonyms.

Both $\omega_{\mathrm{FA}}(\mathrm{s}_{\mathrm{i}})$ and $\omega_{\mathrm{Fs}}(\mathrm{s}_{\mathrm{i}})$ are added together to compute the overall freshness weight coefficient $\omega_{\mathrm{F}}(\mathrm{s}_{\mathrm{i}})$. By incorporating the freshness as a component in the segment scoring procedure, the MUSEUM scorer boosts the relevance score of the terms which are added to the web content freshly. This weight component would be beneficial while evaluating the web pages which changes dynamically. In a dynamic page, the contents which are added freshly get more significance if they hold the content to satisfy the user's information requirement.

The last coefficient in the MUSEUM scorer, visual weight coefficient is explored in the following section.

5.2.1.6 Visual Weight Coefficient

Another dimension used in the MUSEUM scorer for computing the score of the segment is the "Visual Weight". This section explores the method used for computing the visual weight coefficient.

Visual weight coefficient is based on the visual cues added to terms in the web pages which serve as an important parameter in correlating its relevance. More the importance of the term, stronger the visual cue. The visual weight coefficient $\omega_{\mathrm{V}}(s_i)$ computation is based on the variable magnitude approach. Different visual cues are given different weight which shall be customized based on the application specific requirements. The weight allocation for visual cues in the page ω_{V} is as shown in Eq. (5.9).

$$\omega_{\mathrm{V}} = \left\{ \left[\mathrm{v}_1,\ \omega_{\mathrm{V1}}\right],\ \left[\mathrm{v}_1,\ \omega_{\mathrm{V2}}\right],\ \ldots\ \left[\mathrm{v}_n,\ \omega_{\mathrm{Vn}}\right]\right\} \tag{5.9}$$

In Eq. (5.9), the visual cues are represented as "v_{i}". The weight associated with a visual cue "v_{i}" is indicated as ω_{Vi} .

The "visual weight coefficient" $\omega_V(s_i)$ is based on the frequency of query-terms appearing in segment with specific visual markup. Each item in the visual weight array is given different weights as shown in Eq. (5.9).

While calculating the Visual weight coefficient, the weight for the visual cue as shown in Eq. (5.9) is multiplied with the number of query-terms matching with content having that particular visual cue, as represented in Eq. (5.10).

$$\omega_V(s_i) = \begin{Bmatrix} \forall v_i \in V, q_i \in Q : |q_i \setminus v_i| * \omega v_i & if\ |q_i \setminus v_i| > 0 \\ 0 & otherwise \end{Bmatrix} \tag{5.10}$$

In Eq. (5.10), the set of available visual cues in that page is defined as "V". The individual items of the set "V" are represented as "v_i". The query-terms are represented as the set "Q". The operation "$|q_i \backslash v_i|$" computes the count of the query-terms with the specific visual cue "v_i". This count is multiplied by the weight associated for that visual cue. If none of the terms in the query are matching with particular visual features the "visual weight component" is assigned to zero.

This section elaborated about the computation of all the six weight coefficients proposed in this research work. The "Coefficient Strength Factor (CSF)" which is used to incorporate the component specific boosting of score, is explained in the following section.

5.2.2 The Coefficient Strength Factor

The MUSEUM scorer computes the score of each segment in a multidimensional manner through six different coefficients as described in the previous section. The different weight coefficients proposed by the MUSEUM scorer are treated with a variable magnitude approach, with the help of Coefficient Strength Factor (CSF).

The Coefficient Strength Factor (CSF) is applied to individual weight coefficients as shown in Eq. (5.11).

$$\begin{aligned} \omega(\mathrm{s}_i) = (\lambda_F * \omega_F(s_i)) + (\lambda_E * \omega_E(s_i)) + (\lambda_L * \omega_L(s_i)) + \\ (\lambda_V * \omega_V(s_i)) + (\lambda_M * \omega_M(s_i)) + (\lambda_R * \omega_R(s_i)) \end{aligned} \tag{5.11}$$

As there are six weight coefficients, the strength factor also consists of six different components as shown in Eq. (5.11). The components of the CSF are tabulated in Table 5.1.

CSF Component	Description
λ_F	The Freshness Coefficient Strength Factor
λ_E	The Theme Coefficient Strength Factor
λ_L	The Link Coefficient Strength Factor
λ_V	The Visual Coefficient Strength Factor
λ_M	The Image Coefficient Strength Factor
λ_R	The Profile Coefficient Strength Factor

Table 5.1: The CSF – Components

The components of Coefficient Strength Factor listed in Table 5.1 are multiplied with corresponding weight coefficients to increase or decrease the magnitude of the corresponding coefficient. The rationale for this approach is that, if a particular coefficient is not applicable for a page or it has to be suppressed, and then the corresponding co-efficient strength factor can be set to zero. If certain coefficients need to be boosted then higher values can be assigned.

In order to have control over the coefficient strength factor, a maximum bound for the sum of strength factors is set as six which is equal to the number of weight coefficients. By multiplying the coefficient strength factor with the corresponding weight coefficients and summing up all the six components, the segment score $\omega(\mathrm{s_i})$ is computed.

The segment score normalization is done by utilizing the "Inverse Segment Frequency" which is explained in the following section.

5.2.3 The Inverse Segment Frequency

The segment score computation procedure which is explained in this Chapter so far, focused only on the intra segment features. In order to incorporate inter-segment features for normalized score computation procedure, this research work proposes "Inverse Segment Frequency" which is explained in this section.

The Vector Space Model is used to compute the relevance score with respect to a query which involves "term frequency" and "inverse document frequency" (Salton, 1971; Salton et al., 1975). The term frequency computes the number of times the query-terms appears in a page. The "inverse document frequency" harnesses the characteristics of the entire collection of pages in the corpus. The reasons for having term frequency and inverse document frequency in the Vector Space Model are to make sure that the scoring is not skewed towards lengthier documents and to incorporate the interpage features during the computation of page score.

This work extends the idea of "inverse document frequency" from the page level to the segment level by introducing "Inverse Segment Frequency (ISF)". The "Inverse Segment Frequency" of term with respect to a page, $isf(t,P)$ is computed as shown in Eq. (5.12).

$$isf(t,P) = \log \frac{|P|}{\begin{vmatrix} \{s : t \in s\} & if(tf(s,t) \neq 0) \\ 1 & otherwise \end{vmatrix}} \tag{5.12}$$

In Eq. (5.12), the count of segments in the web page is represented as |P|. The function $tf(s,t)$ computes the term frequency of "t" in the segment "s".

The inverse segment frequency $isf(t,P)$ is computed by dividing the number of segments in the page |P| by the count of segments in which the term appears, and applying logarithmic value to it. The inverse segment frequency is incorporated into the segment score computation procedure as shown in Eq. (5.13).

$$\omega(\mathrm{s_i}) = \begin{Bmatrix} (\lambda_F * \omega_F(s_i)) + (\lambda_E * \omega_E(s_i)) + (\lambda_L * \omega_L(s_i)) + \\ (\lambda_V * \omega_V(s_i)) + (\lambda_M * \omega_M(s_i)) + (\lambda_R * \omega_R(s_i)) \end{Bmatrix} * isf \tag{5.13}$$

In Eq. (5.13), the segment score $\omega(\mathrm{s_i})$ is computed by multiplying the "Inverse Segment Frequency" which is derived, as explained in this section. The segment score shown Eq. (5.13) incorporates the sum of products of individual weight coefficients multiplied by the corresponding coefficient strength factor. Further the computed value is multiplied by the "Inverse Segment Frequency" in order to normalize the segment score.

If the "Inverse Segment Frequency" is not included in the score computation procedure, then the lengthier segments with increased count of term occurrences would skew the segment score in favour of longer segments. In order to avoid such an adverse effect the "Inverse Segment Frequency" is incorporated in the score computation procedure.

This research work has an additional layer of segment score enhancement with the inclusion of "Segment Class Weight" which is discussed in the following section.

5.2.4 The Segment Class Weight

This research work encompasses a segment classification component (ClaPS) which was elaborated in Chapter IV. The segment score computed in the previous section is enhanced by "Segment Class Weight" which is described in this section.

The Segment Classifier component of this work classifies the segments of the page into five different classes as follows: the Simple Text Segment, Navigation Segment, Image Segment, Head Segment and A/V Segment. The segments of the web page are classified into any of these classes using a decision tree based approach. The classification performed in this research work belongs to the "Structure Based Classification". The rationale for classifying the segments of the web page is to associate different weights to the segments belonging to various categories. The classification of segments is carried out by extracting various features from the segments and to train a decision tree based on these features.

The MUSEUM segment scorer computes the score of the segment using the "Segment Class Weight". The "Segment Class Weight" is nothing but a way to increase or decrease the segment score, based on the class of the web page segment. For each of the segment classes the "Segment Class Weight" is provided as a customizable key. The value of "Segment Class Weight" is not rigidly fixed by this research work; rather it is designed in such a way that the value for "Segment Class Weight" can be decided, based on the domain specific requirements.

The score of a segment $\omega(s_i)$ is updated by incorporating the "Segment Class Weight" as shown in Eq. (5.14).

$$\omega(s_i) = \omega(s_i) * \Psi(s_i) \tag{5.14}$$

In Eq. (5.14), the “Segment Class Weight” is indicated as $\Psi(s_i)$. The $\Psi(s_i)$ is multiplied with the segment weight derived in Eq. (5.13).

This section explored in detail the segment scorer component MUSEUM which computed the score of the segment by following a multidimensional, variable magnitude approach. The score of the segments are computed with the help six weight coefficients and strength factors for corresponding coefficients. The scoring of segments is further enriched by introducing the segment class weight which facilitated a differential approach in computing the scores of segments belonging to various classes.

The overall score of the page is computed by fusing the scoring of the individual segments which is explained in the following section.

5.3 The Page Score Computation

The MUSEUM scorer computed the scores of the individual segments. Using a multidimensional, variable magnitude approach, the score of the page is computed by combining the score of the individual segments which is computed by a multidimensional, variable magnitude approach. The page score computation from the segment scores is illustrated in this section.

The final relevance score of the page with respect to the user’s query and the profile terms is computed as shown in Eq. (5.15).

$$PS = \sum_{i=1}^{k} \left| count(s_i, t) * \omega(s_i) \right| \tag{5.15}$$

In Eq. (5.15), the score of the page PS is computed by the sum of product of segment scores and the term frequency in the corresponding segment. In Eq. (5.15), the frequency of occurrence of the query-terms in a segment is represented as $count(s_i, t)$. The score of the individual segments is indicated as $\omega(s_i)$. The index “i” is used to navigate through the set of segment scores. The value of the index “i" ranges from one to “k” which is the count of segments in that page.

The page score PS computation involves multiplying the segment score by the number of occurrences of the term. The reason for this multiplication is to boost the scores of segments where either the user profile terms or the query-terms appears more number of times comparing with the other segments of the page.

By this procedure the score of the page is computed as the sum of products of the individual segment scores and the term frequency.

This chapter elaborated in detail about the SCOPAS evaluator component of this research work. The SCOPAS Evaluator's role is to compute the score of page by splitting it into smaller segments. The scores of these web page segments are computed with the help of six different weight coefficients. The specific coefficients are boosted or scaled-down with the help of "Coefficient Strength Factor". Further the segment score is either increased or decreased with the help of "Segment Class Weight" which assigned differential score for segments of various classes. The overall score of the page is computed by the fusing the scores of individual scores together.

This research work incorporates personalized scoring with the help of the user profiles. The user profiles are built in an incremental manner by "SCOPAS Profiler" which is explained in the following Chapter.

Chapter VI: The SCOPAS Profiler

This Chapter focusses on the personalization component which is termed as "SCOPAS Profiler". The overall objective of this work is to provide a personalized content scoring model using a multimodal approach with the help of web page segmentation. The relevance scoring of web pages using a variable magnitude approach is illustrated in the previous Chapter which computed the score of a page in a bottom-up manner by merging the scores of individual segments. The scoring of web page segments are carried out using six weight coefficients. One of the coefficients involved in the scoring process is the "Profile Weight Coefficient". This profile weight coefficient requires the profile of the user as input, which is built by the SCOPAS Profiler component.

The profiler component which is used to amalgamate the user profiles in an incremental manner is explored in this Chapter. The organization of this Chapter is as follows: the need for user profiles in the content scoring process is explained in Section 6.1. The role of the profiler component is narrated in Section 6.2. In section 6.3 the category of user profiling technique adopted by this research work is discussed. The two profile building scenarios are listed out in Section 6.4. The initial profile representation is explained in Section 6.5 and the incremental profile amalgamation is illustrated in Section 6.6.

6.1 Need for User Profiling

This section lists out various needs for incorporating the profile of the user in the page score computation process.

The World Wide Web hosts the largest collection of information in a distributed manner. The information retrieval domain has shifted its focus from the problem of "information scarcity" to "information overload". In the early days of web, it was difficult for the users to find the information that they need, but these days the users have to filter out the required information from a large collection of unrelated set of data.

Generally, the web surfing sessions starts with typing keywords in the search engine to locate the required page rather than directly typing the address in the browser. This kind of entering the keywords has become a common task these days. The same keyword supplied by two different users shall mean different things. For example the keyword “Cricket” generally points to the sport Cricket but when supplied by an entomologist it has a higher probability of representing the insect cricket. In order to differentiate between these representations, the query-terms need to be supported with the user profile information.

These user specific information requirements are handled by the “SCOPAS Profiler” component. The role of the “SCOPAS profiler” component is illustrated in the following section.

6.2 Role of SCOPAS Profiler

The information requirements for different users are not unique as stated in the previous section. The SCOPAS profiler component is responsible for providing personalization in the web content evaluation process.

The profile of the user is amalgamated in an incremental manner using a hybrid approach. The SCOPAS profilers objectives are as follows: Building the profile of the user in an incremental manner by utilizing the local and global context data. The user profile building process is supported by the components of the “SCOPAS Evaluator” to extract the keywords from the resources given by the user.

The role of SCOPAS profiler is to provide the user specific profile terms to the segment score computation process. The SCOPAS profiler builds the profile in two major scenarios. In one of the scenarios, the explicit method is used to collect the data from the user and in another scenario the profile data is gathered in an incremental manner by action monitoring. The incremental profile data collection gathers the profile terms from various dimensions. The SCOPAS profiler represents the profile of the user with semantic profile representation technique which is explored further in this Chapter.

The category of the user profiling technique adopted by the SCOPAS profiler component is illustrated in the following section.

6.3 The User Profiler Category

The profile of the user can be built using various methods. A survey of user profiling domain was presented in Chapter II. The major aspects to be considered with the profile building process of the user are the profile data collection techniques, the profile storage methods and location of the user profiles. This section highlights the specific methods adopted by the "SCOPAS profiler" as illustrated in Fig 6.1.

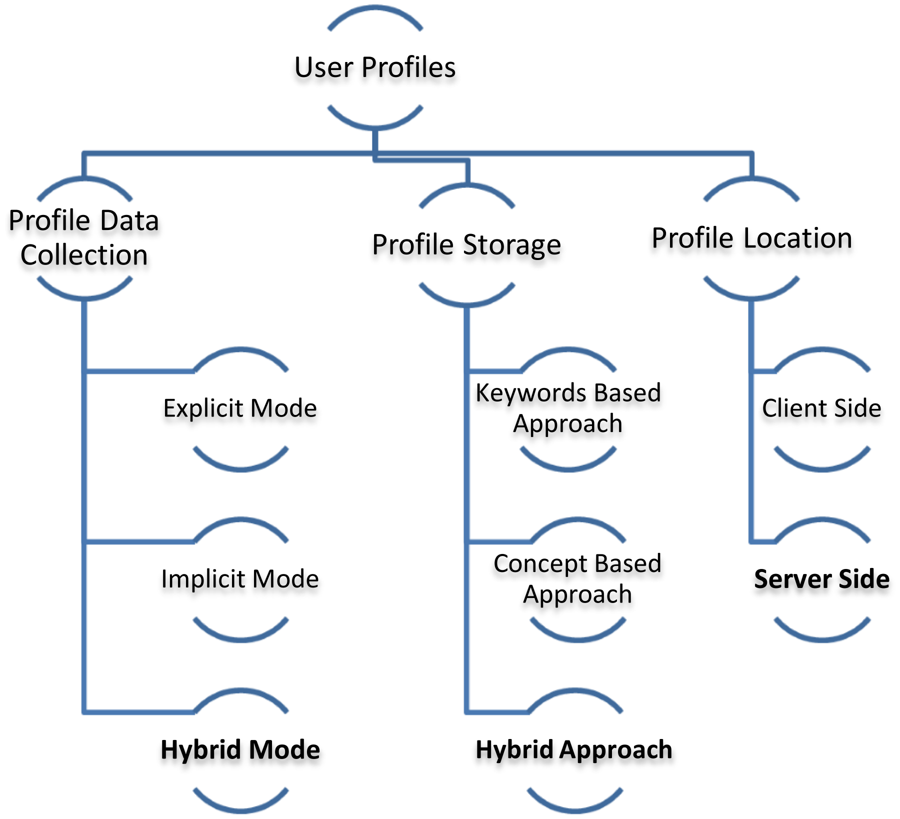


Fig 6.1: The SCOPAS Profiler – Category

One of the important aspects of the user profile handling is the way in which the profile data is gathered from the user. There are three modes of collecting the data from the users: the explicit mode, implicit mode and hybrid mode. The explicit mode requires the user to enter all the profile data explicitly and in the implicit mode the profile data is gathered by monitoring the user actions. The hybrid approach is the combination of both the explicit and implicit modes of gathering the data.

The SCOPAS profiler follows the hybrid mode of gathering the profile data from the user. The SCOPAS profiler adopts the explicit mode while gathering the initial profile data and the incremental enhancement of the profile is done using the implicit mode. Hence the SCOPAS profiler falls under the hybrid profile data collection mode.

The profile data storage is done in three ways as illustrated in Fig 6.1. In the keyword based approach, the profile of the user is represented as a vector of keywords. In the concept based approach, instead of the keywords, the concepts from a hierarchy is utilized to represent the profile. The hybrid approach utilizes both the keywords and concepts to represent the profiles. The SCOPAS profiler utilizes the hybrid profile storage technique by adopting both the keywords and the concepts. The concepts utilized in the SCOPAS profiler are chosen from the Open Directory Project (ODP) concept hierarchy.

The user profiles can be stored either at the client side or at the server side. In the client side profiles, the entire personalization process is carried out in the client machine. In the case of server side profiles, the profiles are directly accessible to server side components. The SCOPAS profiler falls under the server side profile category.

The rationale for choosing the server side profile is that the segment scoring procedure is carried out on the server side which requires the profile data. Hence the profile data gathered for the user in an incremental manner is stored in the server side. For gathering the data from the user by action monitoring, a client side component is utilized for monitoring the user actions and submits the data to the SCOPAS profiler located at the server side.

The two scenarios of user profiling i.e. the initial profile building and the incremental profile amalgamation are explored in the following section.

6.4 The User Profiler - Scenarios

The profile data gathering of the user is carried out using two different scenarios by the “SCOPAS Profiler” which is explored in this section. The two scenarios of the user profile building process are illustrated in Fig 6.2.

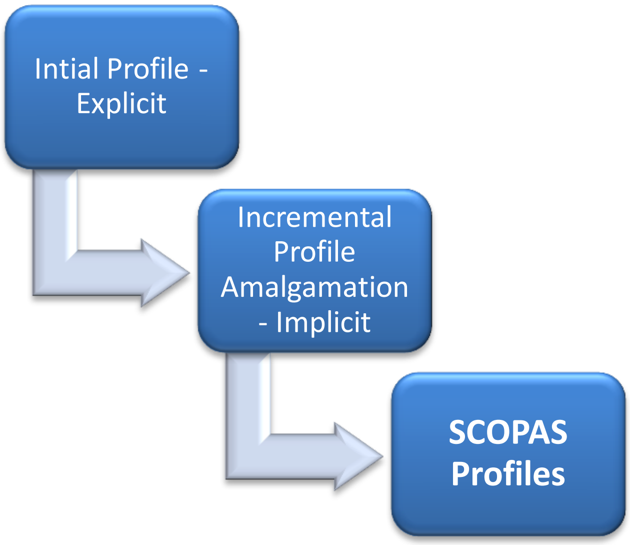

Intial Profile - Explicit
Incremental Profile Amalgamation - Implicit
SCOPAS Profiles

Fig 6.2: The User Profiler Scenarios

The SCOPAS profiler gathers the profile data by a combination of both the implicit and explicit techniques. Initially the profile data of the user is gathered in an explicit mode where the user has to provide the preference details explicitly to the SCOPAS profiler. The initial data gathering consists of both the personal and workplace related preferences which are explored further in Section 6.5. The initial profile representation of the user is based on a semantic technique called FOAF which was introduced in Section 2.3.4.

As the profile of the user is not a static entity, the SCOPAS profiler incorporates an incremental profile amalgamation component which enriches the user profiles by monitoring the actions performed by the user. The incremental profile amalgamation procedure involves various components which are explored in Section 6.6.

The profile of the user has the data provided by the user at the initial instance and the data gathered with the incremental profile amalgamation technique. The profile keywords supplied to the "SCOPAS Evaluator" is a combination of the terms gathered using both the above specified scenarios.

The initial profile building scenario which is carried out in an explicit manner is explained in the following section.

6.5 Initial Profile Building

This section focusses on the initial profile building process which requires the user to provide the data explicitly for the first time. The data provided by the user during this stage serves as the “seed data” for further enriching the profile by fetching the extended profile terms.

In the initial profile building stage the data is collected from the user in a multidimensional manner. Each dimension fetches the keywords, which represents the user’s preferences in different aspects. These profile data are represented using the FOAF ontology based representation which is explored in the following section.

6.5.1 The FOAF Representation

An introduction to the FOAF (Friend Of A Friend) was presented in Section 2.3.4. FOAF is a promising semantic web technology which is harnessed by many studies in enhancing the web scale information systems (Adamic et al., 2003; Ding et al., 2005; Golbeck et al., 2003; Grimnes et al., 2004). This section explores the reasons for selecting FOAF and the various fields provided by FOAF which are utilized by the SCOPAS profiler.

6.5.1.1 The Reasons for Choosing FOAF

The reasons for selecting FOAF in the user profile representation in this research work are as follows: Being a XML based technique the FOAF is readable by both machines and humans which make it a convenient structure for representing the profiles, as the data in the profile need to be read by the machines. Another reason for utilizing FOAF is that the profiles of users with similar interest shall be merged in a easy manner. The rich collection of fields supported by the FOAF is another important reason for the choice of FOAF as the profile representation technique.

The various FOAF fields utilized by the SCOPAS Profiler, to represent the user profile data in a multidimensional manner is explored in the following section.

6.5.1.2 The FOAF Fields

The FOAF specification involves various fields. Among them the SCOPAS profiler utilizes four fields to capture the profile data of the user through various dimensions.

The reason for selecting these four fields is that they capture the user's interest from both personal and work-related dimensions. In this section the four fields utilized by the SCOPAS profiler are described as illustrated in Fig 6.3.

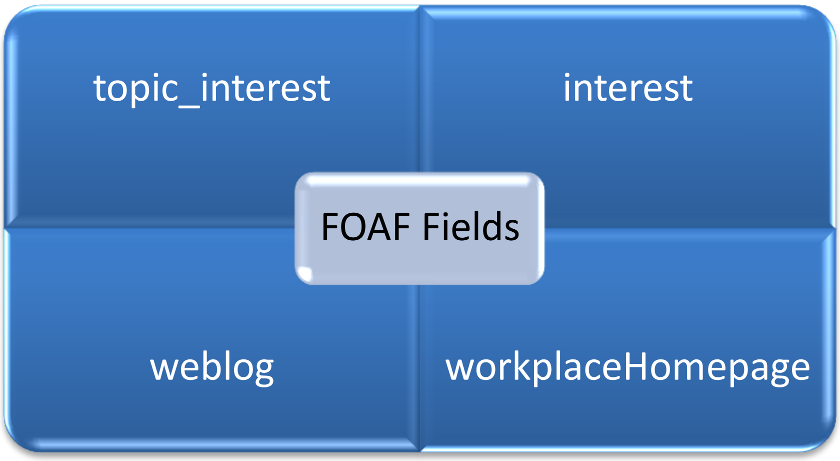


Fig 6.3: FOAF Fields

The four fields placed in Fig 6.3, "topic_interest", "interest", "weblog" and "workplaceHomepage" gathers the information requirements of the user using different contexts. The descriptions about each of these fields are tabulated in Table 6.1.

The "topic_interest" field is used to hold the interest of the user in specific topics. The "interest" field is used to point to resources in which the user is interested. The resources are identified through URIs. The "weblog" field points to the "blog" maintained by the user. In case, if the user is not having a blog then this field is utilized to point to the blog(s) which the user visits regularly.

The "workplaceHomepage" is used to point to the place where the user is working. This field is utilized so that the work dimension related profile terms are also maintained in the "profile bag" of the user. The "profile bag" is used to indicate the collection of all the profile terms gathered through the four fields specified above.

Field	Description
foaf:topic_interest	The "thing" of interest to the user
foaf:interest	A page about a topic of interest to this person.

foaf:weblog	A weblog of some “thing” / person
foaf: workplaceHomepage	A workplace homepage of some person

Table 6.1: The FOAF Fields and Description

The method proposed by this research work to extract the profile terms from the resources identified by the fields specified in Table 6.1 are explored in the following section.

6.5.2 Profile Terms Extraction

Among the fields specified in the previous section, except the “topic_interest” field all the other fields provide a pointer to a resource. Hence it becomes necessary that these resources provided by the user, need to be parsed, to extract the profile specific keywords. This section explores the method proposed by this research work to extract the profile terms from these resources.

In the Eq. (6.1), the profile bag Ω is shown as a combination of four components. In Eq. (6.1), α denotes the topic_interest, β denotes the “interest”, δ denotes the “weblog” and ε denotes the“workplaceHomepage”.

$$\Omega = \left\{ \begin{matrix} \alpha \\ \beta \\ \delta \\ \varepsilon \end{matrix} \right\} \qquad (6.1)$$

The topic_interest field which holds the interest terms directly is searched in the Open Directory project for expanding the key terms, as shown in Eq. (6.2).

$$\Omega' = \left\{ \begin{matrix} \Gamma(\alpha) = \{\lambda_1, \lambda_2 \ldots \lambda_n\} \\ \beta \\ \delta \\ \varepsilon \end{matrix} \right\} \qquad (6.2)$$

The reason for searching the “topic_interest” in the Open Directory Project is to broaden the profile term bag with the concept hierarchy provided in the Open Directory Project. In Eq. (6.2), the retrieval of concepts from the Open Directory Project is shown as $\Gamma(\alpha) = \{\lambda_1, \lambda_2 \ldots \lambda_n\}$. Each λ_i represents the entities retrieved from the concept hierarchy. By this process, topic_interest which holds the "things" the user is interested in, is expanded further for better context disambiguation.

As stated earlier, except the "topic_interest" all other fields hold the resource pointers in the form of URIs. Hence for all other fields, the resource pointers are resolved to fetch the corresponding resources and evaluated for fetching the profile terms. In evaluating the resources pointed by the other three fields, a variation of the MUSEUM scorer is applied. As elaborated in Chapter V, the MUSEUM scorer evaluated the score of the page with respect to a query. In this keyword extraction process, there are no queries. So, the query-term component is replaced in two different ways: utilizing the terms extracted by "topic_interest" or utilizing the resource titles as key terms.

Using the above specified approach, the segments with top "N" score are utilized in fetching the key terms. The value of "N" is provided as a customizable parameter which shall be assigned value based on the average segment count in the entire corpus. In order to follow such a mechanism for key term extraction, the web page is split into various segments using "SCOPAS Segmentor" as shown in Eq. (6.3). The individual segments of the web page pointed by the fields are shown as $\{\omega_1, \omega_2 ... \omega_n\}$.

$$\Omega'' = \left\{ \begin{array}{c} \Gamma(\alpha) = \{\lambda_1, \lambda_2 ... \lambda_n\} \\ \Psi(\beta) = \{\omega_1, \omega_2 ... \omega_n\} \\ \delta \\ \varepsilon \end{array} \right\} \qquad (6.3)$$

In each segment ω_i, the terms are extracted using the procedure explained above. This work adopts one more layer of enhancement to this process by analyzing the contents of the segments with the help of "Content Analysis" services. The content analysis service analyses any unstructured content and fetches the key terms from them. This research work has adopted the Yahoo! Content Analysis service for the purpose. The process of merging the content analysis service score and the multidimensional approach based score proposed by this research work is as shown in Eq. (6.4).

$$\Psi(\beta) = \{\forall_{i=1..n} \psi(\omega_i) \oplus \vartheta(\omega_i)\} \qquad (6.4)$$

In (4), $\psi(\omega_i)$ represent the segment score calculated and $\vartheta(\omega_i)$ represent the Content Analysis score. The content analysis would output a weighted array with extracted terms and their weight. The $\oplus$ operator indicates the fusion of scores generated by both the components.

The same procedure is repeated for the URIs represented by other two fields. The representation for the "weblog" component is as shown in Eq. (6.5).

$$\Omega''' = \left\{ \begin{matrix} \Gamma(\alpha) = \{\lambda_1, \lambda_2 ... \lambda_n\} \\ \Psi(\beta) = \{\omega_1, \omega_2 ... \omega_n\} \\ \Theta(\delta) = \{\nu_1, \nu_2 ... \nu_n\} \\ \varepsilon \end{matrix} \right\} \tag{6.5}$$

The weblog term extraction is indicated as $\Theta(\delta) = \{\nu_1, \nu_2 ... \nu_n\}$ in Eq. (6.5). Hence the symbol Ω''' denotes the enriched profile bag.

The representation for the "workplaceHomepage" is as shown in Eq. (6.6).In Eq. (6.6), the "workplaceHomepage" term extraction is shown as $\Phi(\varepsilon) = \{\kappa_1, \kappa_2 ... \kappa_n\}$.

$$\vec{\Omega} = \left\{ \begin{matrix} \Gamma(\alpha) = \{\lambda_1, \lambda_2 ... \lambda_n\} \\ \Psi(\beta) = \{\omega_1, \omega_2 ... \omega_n\} \\ \Theta(\delta) = \{\nu_1, \nu_2 ... \nu_n\} \\ \Phi(\varepsilon) = \{\kappa_1, \kappa_2 ... \kappa_n\} \end{matrix} \right\} \tag{6.6}$$

The enriched profile bag of the user after extracting terms from all the four fields specified above is represented as $\vec{\Omega}$. The final profile bag representation $\vec{\Omega}$ is the built by merging all of the above specified intermediate representations Ω', Ω'' and Ω'''. This initial profile serves as the personalization component for the new user. This profile is incrementally enriched by monitoring the user actions which is explored in the following section.

6.6 Incremental Profile Amalgamation

This work adopts the dynamic user profiles. The initial profile of the user which is built by the procedure elaborated in the previous section is enhanced in an incremental manner. The incremental profile amalgamation approach enhances the user profile through monitoring of user actions, in terms of various factors. The incremental profile amalgamating factors are given variable weights for selecting the resources to extract the profile terms.

6.6.1 The Action Monitoring Factors

This research work incorporates four action monitoring factors while incrementally building the profiles, as illustrated in Fig 6.4. The factors utilized for selecting the resources for profile term extraction are bookmarking, time threshold, persistence and hard-copying. Each of these factors point to one or more resources which would be evaluated using the procedure narrated in 6.5.2. Each of these action monitoring factors are given different weights which is decided based on the scenario.

The resources are extracted, segmented and evaluated for identifying the keywords using Eq. (6.4). The profile-bag would be enhanced with the terms extracted using each of the resources identified by all the four factors. The incremental profile building features are explored in this section.

6.6.1.1 Bookmarking

Bookmarks serves as an important factor in deciding the user interest. The bookmarks for improving the retrieval efficiency have been carried out earlier (Vallet et al., 2010). The "SCOPAS Profiler" utilizes the bookmarks for enhancing the profile bag.

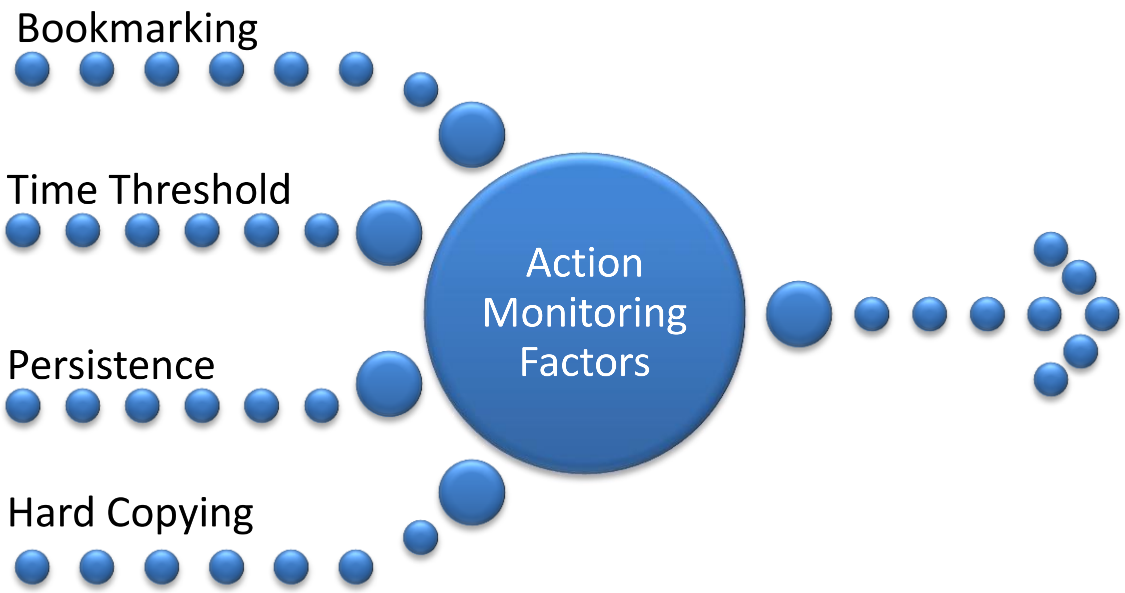


Fig 6.4: Four Action Monitoring Factors

The bookmarks hold a URI which is resolved and evaluated using the procedure specified in the Section 6.5.2. The "Bookmark" factor handling is represented as shown in Eq. (6.7).

$$\overrightarrow{\Omega_E} = \Omega \bigcup \{\forall_{i\ =\ 1..n} score(\tau_i) * \vartheta_1\} \quad (6.7)$$

In Eq. (6.7), each bookmark is specified as τ_i . The enhanced profile bag is represented as $\overrightarrow{\Omega_E}$. The $score(\tau_i)$ computes the score of the segments in the web page pointed by each bookmark. In Eq. (6.7), ϑ_1 indicates the weight boost given to the bookmarking factor. All the four factors are given different boost values to associate a variable magnitude approach.

6.6.1.2 Time Threshold

The amount of time the user has spent in a web page serves as an indicator for the interest level of the user towards that web page. If the user has stayed in a web page for more than a threshold time limit then it can be inferred that the user is interested in the topic represented by that web page. This is harnessed by the SCOPAS Profiler during the incremental amalgamation of the user profiles.

The SCOPAS profiler collects all the pages in which user has stayed more than a threshold limit and those pages are segmented and evaluated as described earlier. The "time threshold" factor utilization for enhancing the profile bag is as shown in Eq. (6.8).

$$\overrightarrow{\Omega_E} = \overrightarrow{\Omega_E} \bigcup \left\{ \begin{array}{l} \forall_{i\ =\ 1..n} score(\tau_i) * \vartheta_1 \\ \forall_{i\ =\ 1..n} score(\mu_i) * \vartheta_2 \end{array} \right\} \quad (6.8)$$

In Eq. (6.8), each page in which user has stayed more than a threshold limit is indicated as μ_i . The time threshold boost value is represented as ϑ_2 .

6.6.1.3 Persistence

Another factor utilized in the incremental profile amalgamation process is the "Persistence" which refers to the user's act of saving a web page to their hard disk. The persistence is an important factor is deciding the user's interest towards a web page. If the user is saving the web page in the system then it indicates the increased interest level of the user towards the contents specified in that page.

The SCOPAS profiler enhances the profile bag by extracting the profile terms from all the web pages which are saved by the user for future references. The persistence factor computation is as shown in Eq. (6.9).

$$\overrightarrow{\Omega_E} = \overrightarrow{\Omega_E} \cup \begin{Bmatrix} \forall_{i=1..n} score(\tau_i) * \vartheta_1 \\ \forall_{i=1..n} score(\mu_i) * \vartheta_2 \\ \forall_{i=1..n} score(\sigma_i) * \vartheta_3 \end{Bmatrix} \qquad (6.9)$$

In Eq. (6.9), σ_i indicates the pages saved by the user to their system and ϑ_3 indicates the persistence boost value.

6.6.1.4 Hard-Copying

Another factor utilized in incremental profile building is “hard-copying”. The “hard copying” refers to the act of user printing a web page. If the user is printing a web page then it indicates the user’s increased interest level towards that page. The SCOPAS profiler monitors the tasks of user printing the web pages and collects all the URIs which were printed. These URIs are resolved and profile terms are extracted from them by segmentation and segment score computation. The “hard-copying” factor of the incremental profile amalgamation is represented as shown in Eq. (6.10).

$$\overrightarrow{\Omega_E} = \overrightarrow{\Omega_E} \cup \begin{Bmatrix} \forall_{i=1..n} score(\tau_i) * \vartheta_1 \\ \forall_{i=1..n} score(\mu_i) * \vartheta_2 \\ \forall_{i=1..n} score(\sigma_i) * \vartheta_3 \\ \forall_{i=1..n} score(\varepsilon_i) * \vartheta_4 \end{Bmatrix} \qquad (6.10)$$

In Eq. (6.10), ε_i represent the pages printed by the user. The boost value used for the “hard-copying” factor is represented as ϑ_4.

The initial profile of the user is enhanced by all of the above mentioned factors. This research work has introduced an additional layer of enhancement by introducing boost values. The boost values assigned to four factors can be different. The values of boost factor and the threshold are given as a customizable key which can be set based on the domain specific requirement, where the model is applied. The incremental profile enhancement task is carried out at regular intervals of time so that the different factors specified above contribute to the updation of user profile bag by incorporating new profile terms.

This Chapter focused on the “SCOPAS Profiler” component which involves two major scenarios in building the user profiles. The initial profile of the user is gathered with an explicit data collection approach which involved various FOAF fields. The initial profile is enhanced through monitoring the user action with four factors which include, bookmarking, time threshold, persistence and hard-copying. The resources selected by these factors are evaluated with a multidimensional process explored in the previous Chapter and key terms are extracted. The major benefit of adopting this hybrid approach is that the profile of the user is made to reflect the user’s requirements in an evolving manner.

Chapter VII: SCOPAS Experiments and Model Realization

This Chapter elaborates about various experiments conducted on the SCOPAS model which aimed towards building a multimodal content scoring for web pages using segmentation and user profile amalgamation.

The conceptual methods proposed in the previous Chapters need to be evaluated for their effectiveness through empirical evidence. The efficiency of both the specific components of the model and the overall efficiency of the model are analyzed through a set of experiments conducted which are presented in this Chapter. As the proposed SCOPAS model is a generic content scoring model, the domain specific efficiency of the model is proven with the help of realization of the model with specific applications.

The empirical validation of the components of the model is provided with the help of various metrics. Specific metrics are chosen to prove the efficiency of the various components. This Chapter is organized as follows: It is broadly divided into two major sections, the experiments on the SCOPAS components and the realization of the model in specific domains. The experimental setup which is utilized as a test-bed for the SCOPAS model is explored in Section 7.1.1. The analysis of the segmentation component, SCOPAS-Segmentor is provided in Section 7.1.2 through various metrics. In section 7.1.3, the analysis of the classification component, ClaPS is explored. The analysis of the "SCOPAS-Profiler" is presented in Section 7.1.4.

As stated above the efficiency of the model is proven with the help of domain specific realizations and it is discussed in Section 7.2. The "SCOPAS-Rank" which is a realization of the model for "re-ranking" the search engine results is explained in Section 7.2.1. The CaSePer which is a realization of the model in the "Change Detection" domain is explained in Section 7.2.2. The realization of the SCOPAS model in the mobile rendering domain termed "MORPES" is explained in detail in Section 7.2.3. A web page summarization approach using SCOPAS components is elaborated in Section 7.2.4.

7.1 SCOPAS Experiments

The specific components of the SCOPAS model are empirically validated with the help of various metrics. This section begins with explaining the experimental setup utilized for conducting the experiments. The analysis of segmentation component, classification component and the user profile component are also provided in this section.

7.1.1 The Experimental Setup

Both the hardware and software context of the experimental setup which is used as the test bed for conducting the experiments on various components is presented in this section.

The model's implementation is carried out with the help of Open Source Software stack. The software stack includes Linux, Apache, MySql and PHP. The rationale for choosing the open source software stack is to make the implementation both vendor and architecture neutral. The above specified software components can be installed on all types of hardware without focusing much on the architecture.

The hardware configuration of the system includes a Quad Core processor, 8 GB of main memory and 128 Mbps leased line internet connection. With respect to client side components the hardware configurations are not restricted, as the real time web access scenarios and the spectrum of client side hardware are very large in size.

With respect to the datasets, experiments were conducted using both the standard datasets and the custom built collection of pages collected with the help of a "Lucene" based crawler.[5] The crawler is provided with a set of seed URLs and "revisit interval flag" (RIF). The seed URLs are the locations from where to start the crawl and the revisit interval flag holds the time interval between the successive visits to the same page by the crawler. The revisit interval flag is assigned a value based on the type of page. In the experiments, three different categories of seed URLs were used. They are "academic", "personal" and "news". The rationale for having pages from various categories is to make the data set enriched with pages of different categories rather than a monotonous collection of a single category of pages.

[5] http://lucene.apache.org/core/

For academic category the RIF is set as 48 hours, for personal as 24 hours and for news as one hour. The rationale behind such an approach is that the frequency of modification gradually reduces from the news web pages to personal to academic. The snapshots of the pages are stored in the “Page Evolution Track” introduced in Chapter V which holds the snapshots of pages across a temporal dimension. The rationale for incorporating the “Page Evolution Track” is that the multimodal content scoring model involves “freshness” as a component which requires the temporal predecessors of the current page in order to compute the freshness weight coefficient. The average page evolution track length of the pages crawled in the corpus is 4.6. The length of the Page Evolution Track refers to the number of temporal predecessors available in the corpus.

7.1.2 The Segmentation Analysis

The SCOPAS model incorporates a web page segmentation component termed “SCOPAS Segmentor” which was elaborated in detail in Chapter III. The role of “SCOPAS-Segmentor” is to split the web pages into segments with a hybrid segmentation technique incorporating the “page tree” and “densitometry” as two components. This section discusses in detail about the efficiency of the segmentation process carried out by the “SCOPAS-Segmentor”.

The segmentation is analyzed with the help of two metrics namely, “Adjusted Rand Index” (ARI) and “Normalized Mutual Information” (NMI). Both these metrics are explained, before proceeding to the analysis of SCOPAS segmentor component. These metrics are chosen from the cluster correlation domain, as segmentation can also be considered as a clustering problem by comparing two sets of segments.

7.1.2.1 Segmentation Metrics

The metrics used for the evaluation of the segmentation process are as illustrated in Fig 7.1.

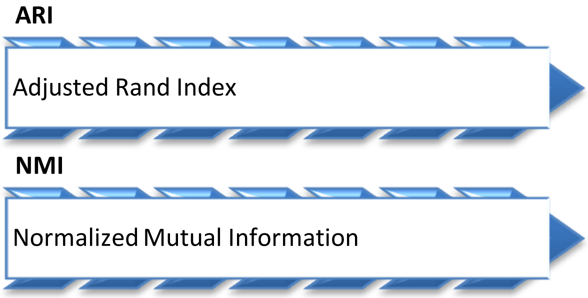


Fig 7.1: The Segmentation Metrics

These metrics are chosen from the clustering domain and utilized to evaluate the clustering efficiency. The segmentation problem is casted into a clustering problem by measuring the correlation between the two set of segments generated from a web page. For this purpose, the segmentation is initially carried out manually for the set of pages for which the segmentation accuracy is going to be measured. Following this manual segmentation process, the same set of pages got segmented by the “SCOPAS-Segmentor”. The result sets built by the above said processes are compared with the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI).

These two metrics are chosen to evaluate the performance of “SCOPAS-Segmentor”, as other popular studies (Chakrabarti et al., 2008; Kohlschütter, 2009) on web page segmentation have utilized these metrics in their evaluation process. The results of the segmentation process are compared using the above specified metrics with these works.

7.1.2.1.1 Adjusted Rand Index (ARI)

One of the metrics which is used to evaluate the efficiency of the segmentation component of the SCOPAS model is Adjusted Rand Index (Hubert and Arabie, 1985). It is an improvement over the “RAND Index”. The Rand Index is used to measure the similarity among two data clustering. The computation of RAND Index is as explained below.

Consider a set with “n” elements $\Omega = \{\lambda_1, \lambda_2 ... \lambda_n\}$ and two partitions of the same set $\Omega_A = \{\alpha_1, \alpha_2 ... \alpha_m\}$ and $\Omega_B = \{\beta_1, \beta_2 ... \beta_o\}$. The partition Ω_A divides the set into “m” subsets and the partition Ω_B divides the set into “o” subsets. The value of “Rand Index” is computed as shown in Eq. (7.1).

$$RI = \frac{a+b}{a+b+c+d} \tag{7.1}$$

Where

“a” indicates count of elements in Ω which are in the same set Ω_A and Ω_B

“b” indicates the count of elements in Ω which are in same set Ω_A and different set Ω_B

“c” indicates the count of elements in Ω which are in different set Ω_A and same set Ω_B

“d” indicates the count of elements in Ω which are in different sets Ω_A and Ω_B

The Rand Index (RI) is assigned a value which ranges from zero to one. The value of RI closer to zero indicates disagreement and a value closer to one indicates perfect agreement.

One of the major issues with the simple Rand Index is that the random arrangement of elements doesn’t guarantee a value closer towards zero. In order to solve this problem the Adjacency Rand Index is adopted in this evaluation process instead of simple Rand Index.

The “Adjusted Rand Index (ARI)” is an enhanced version of Rand Index which is assigned a maximum value of one and for the random sets it assigns zero. The Adjusted Rand Index is computed with the help of a “Contingency Table” which is shown in (7.2).

Confusion Matrix

$$\begin{array}{c|cccc|c} \Omega_A \setminus \Omega_B & \beta_1 & \beta_2 & . & \beta o & \mathit{Sum} \\ \hline \alpha_1 & n_{11} & n_{12} & . & n_{1o} & a_1 \\ \alpha_2 & n_{21} & n_{22} & . & n_{2o} & a_2 \\ . & . & . & . & . & . \\ \alpha_m & n_{m1} & n_{m2} & . & n_{mo} & a_m \\ \hline \mathit{Sum} & b_1 & b_2 & . & b_o & \end{array} \tag{7.2}$$

The elements of the contingency table which is in the form of a matrix, represent the number of objects common between the sets α_i and β_j .

The Adjusted Rand Index is computed using the values specified in the contingency table (7.2), as shown in Eq. (7.3).

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \quad (7.3)$$

In Eq. (7.3), the values for all the variables are fetched from the contingency table (7.2). The value of Adjacency Rand Index computed with the above procedure is used in the analysis of segmentation component of this research work.

Another metric used for the analysis of segmentor component, the Normalized Mutual Information (NMI) is explained in the following section.

7.1.2.1.2 Normalized Mutual Information (NMI)

The “Normalized Mutual Information” metric was developed by Strehl and Ghosh. (Strehl and Ghosh, 2003). It is utilized for measuring the accuracy of the segmentation carried out by the SCOPAS-Segmentor. The Normalized Mutual Information is defined as the mutual information between two partitioning’s. The Normalized Mutual Information is normalized by the geometric mean of the entropies.

The Normalized Mutual Information is computed as shown in Eq. (7.4)

$$NMI(\Omega_A, \Omega_B) = \frac{I(\Omega_A, \Omega_B)}{\sqrt{H(\Omega_A) H(\Omega_B)}} \qquad (7.4)$$

In Eq. (7.4), the Normalized Mutual Information (NMI) computation comparing two partitions Ω_A and Ω_B are given. In Eq. (7.4), $I(\Omega_A, \Omega_B)$ denotes the mutual information between Ω_A and Ω_B. The function $H(\Omega_A)$ denotes the entropy of Ω_A and $H(\Omega_B)$ denotes the entropy of $H(\Omega_B)$.

The segmentation accuracy is directly proportional to the value of Normalized Mutual Information. Higher the value of NMI, better the segmentation accuracy is. The value of one indicates the best results.

This section explored the two segmentation metrics, “Adjusted Rand Index” (ARI) and “Normalized Mutual Information” (NMI). The analysis results of SCOPAS-Segmentor with respect to two these metrics are presented in the following section.

7.1.2.2 The SCOPAS Segmentor Result Analysis

This section explores the SCOPAS-Segmentor result analysis. In order to establish the efficiency of the segmentation process various experiments were conducted on the SCOPAS Segmentor. The segmentation problem is casted into a clustering problem where the comparison is made between the actual segments and the segments built by the SCOPAS-Segmentor.

In order to get the actual segments from the user’s perspective, the set of pages in the collection are segmented manually. These segment boundaries identified by the user are compared with the segment boundaries computed by the SCOPAS – Segmentor with the help of the two metrics specified in the previous section.

The segmentation carried out by the SCOPAS segmentor is compared with the existing segmentation approaches for efficiency and accuracy. Two previous studies are chosen where the segmentation is measured in terms of Adjusted Rand Index and Normalized Mutual Information (Chakrabarti et al., 2008; Kohlschütter, 2009). One of the study (Chakrabarti et al., 2008) has utilized the Graph theoretic approach for segmenting the pages. Another study (Kohlschütter, 2009) is based on the densitometry based segmentation technique.

Segmentation Method	Adjusted Rand Index	Normalized Mutual Information
WordWrap	0.35	0.62
TagGap	0.41	0.58
Block Fusion	0.60	0.75
Gcuts	0.60	0.76
SCOPAS	**0.67**	**0.77**

Table 7.1: SCOPAS – Segmentor Result Analysis

The results of the experiments conducted on the SCOPAS model is provided in the Table 7.1 along with the other two studies mentioned earlier. The Chart which compares the segmentation performance using the above specified metrics is shown in Fig 7.2.

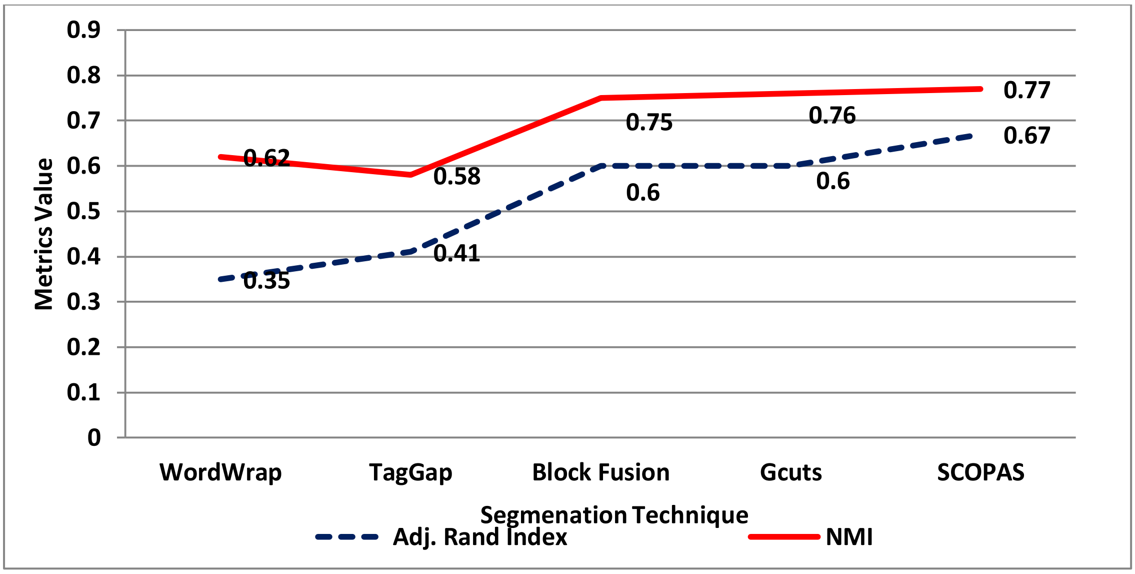


Fig 7.2: SCOPAS Segmentor – Result Analysis

It can be observed from Table 7.1 and Fig 7.2 that five different types of techniques “WordWrap”, “TagGap”, “Block Fusion”, “Gcuts” and “SCOPAS” are compared for the accuracy and efficiency of segmentation. Out of these five methods, the “WordWrap” and “TagGap” are considered as the baseline models. The size of the dataset used for this comparison is in-line with the size of the dataset used by the other two studies (Chakrabarti et al., 2008; Kohlschütter, 2009). The rationale for this approach is that comparison among the experiments conducted with datasets of considerable size difference would lead to skewing of metrics towards either the larger or smaller datasets.

The chart clearly illustrates that the “SCOPAS – Segmentor” is having a better metrics value compared with other works. The Adjusted Rand Index value of 0.67 and the Normalized Mutual Information value of 0.77 serve as the empirical evidence to the efficiency and accuracy of the “SCOPAS-Segmentor” component.

This section explored the performance analysis of the SCOPAS-Segmentor component with the help of two critical metrics. The Segment classifier component analysis is presented in the following section.

7.1.3 The Classification Analysis

The SCOPAS model incorporates a segment classifier component (ClaPS) in order to associate a variable magnitude approach while computing the segment scores.

As stated in Chapter IV, the segment classifier ClaPS falls under the multi-class, single label, hard classification category. The ClaPS utilizes the “Decision trees” to classify the web page segments built using the segmentor component. The segments are classified into five different categories: Simple Text segment, Navigation segment, Image Segment, Head Segment and Audio / Video segment. The segments of web page are labeled into any one of the above specified classes. The classification of these segments is carried out with the help of five different segment features: Text Ratio, Link Ratio, Image Count, Head Ratio and Object Count which are extracted from the segments.

The web page segment classifier component “ClaPS” is evaluated with the help of various metrics which are explored in the following section.

7.1.3.1 Classification Metrics

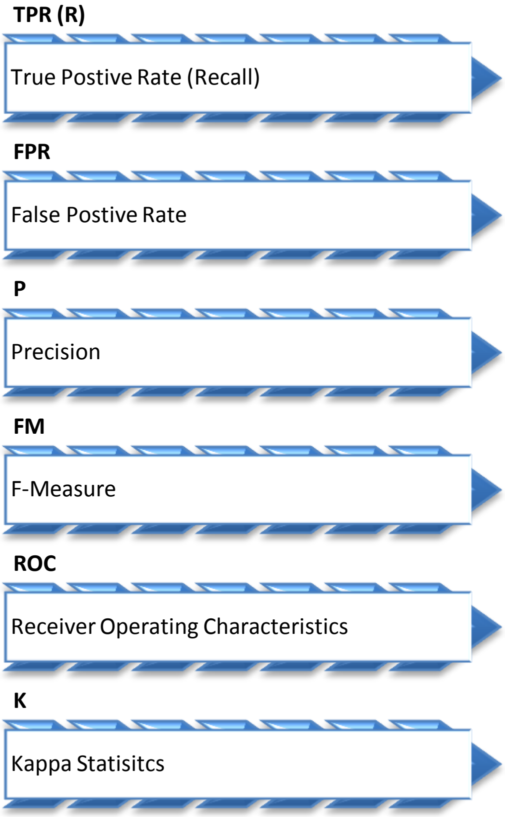


Fig 7.3: Classification Metrics

The multi-class, single label and hard classification technique followed in the SCOPAS model is evaluated with the help of various metrics which are discussed in this section. The metrics which are utilized in the result analysis process of ClaPS are as illustrated in Fig 7.3.

To explain some of the metrics defined above, a “Confusion Matrix” is defined which holds various measures used in analyzing the classification results (Kohavi and Provost, 1998).

The confusion matrix is as shown in Table 7.2. The following metrics are defined from the confusion matrix

		Classifier Output	
		Negative	Positive
Actual Output	Negative	**a** (True Negative)	**b** (False Positive)
	Positive	**c** (False Negative)	**d** (True Positive)

Table 7.2: Confusion Matrix

7.1.3.1.1 True Positive Rate (TPR)

The “True Positive Rate” (TPR) or “Recall” is defined as the fraction of, true positives (d) divided by the sum of false negative (c) and true positive as shown in Eq. (7.5).

$$TPR = \frac{d}{c+d} \tag{7.5}$$

7.1.3.1.2 False Positive Rate (FPR)

The “False Positive Rate” (TNR) is defined as the fraction of, false positives (b) divided by the sum of true negatives (a) and false positives (b), as shown in Eq. (7.6).

$$FPR = \frac{b}{a+b} \tag{7.6}$$

7.1.3.1.3 Precision (P)

The “Precision” is defined as the fraction of, true positives (d) divided by the sum of false positives (b) and true positives (d), as shown in Eq. (7.7).

$$P = \frac{d}{b+d} \tag{7.7}$$

7.1.3.1.4 F-Measure

F-Measure is computed as the harmonic mean of Precision and Recall. The F-Measure computation is as shown in Eq. (7.8).

$$FM = \frac{2.P}{P+R} \tag{7.8}$$

In Eq. (7.8), “P” represents the Precision value and “R” represents the “Recall” value. As the F-Measure incorporates both the precision and recall in its computation, it is considered as an important metric in the evaluation process.

7.1.3.1.5 Receiver Operating Characteristics

The Receiver Operating Characteristics (ROC) is a measure computed by plotting the fraction of True Positive Rate (TPR) versus the False Positive Rate (FPR). The ROC curve is plotted by placing the False Positive Rate in the X-axis and placing the True Positive Rate in the Y-axis. The sampling is carried out at various cut-off levels. The ROC – Area Under the Curve (AUC) is an important metric in evaluating the machine learning based classification (Bradley, 1997).

The “ROC – AUC” is utilized for measuring the classification in this research work. The Area Under the Curve for individual segment classes is computed. Larger the AUC value better the classification. The ROC area is assigned a value between 0.5 and 1.0. The value of 0.5 indicates a useless test and in perfect case the value is assigned as 1.0.

7.1.3.1.6 Kappa Statistics

Another critical measure used in analyzing the performance of the classifier is the Kappa Statistics. The Kappa Statistics is the measure of agreement among various raters. The Kappa Statistics is computed as shown in Eq. (7.9).

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{7.9}$$

In Eq. (7.9), P(A) indicates the proportion of times that the raters agree and P(E) represents the “chance” or random agreement. As the Kappa Statistics incorporates the chance agreement in the computation it is considered to be an efficient measure for the classification tasks.

The Kappa Statistics will be assigned a value which ranges from zero to one. The value of zero for the Kappa statistics indicates no agreement among the raters and the value of one indicates a perfect agreement scenario.

The above mentioned metrics are utilized in the analysis of the SCOPAS-Classifier “ClaPS”. The classification accuracy is evaluated using a technique called “Stratified Cross Validation with 10 Folds” which is explained in the following section.

7.1.3.2 Stratified Cross Validation With 10 Folds

The Cross Validation is a technique utilized to measure the accuracy of a prediction, using an independent dataset. As the proposed web page segment classification model utilizes the decision tree based approach which is a machine learning technique, its accuracy need to be predicted.

The decision tree based classification requires a training set using which the model is built. This model is used in classifying the instance data. The model built in the training phase is tested with a test data set. This testing process is carried out using Stratified Cross Validation with 10 Folds.

The Stratified Cross Validation with 10-Folds method breaks the data into 10 different sets. Each set would be of the size “n” divided by ten. Then training is conducted on nine datasets and test is conducted on the remaining one set. The procedure is repeated 10 times by iterating through all the ten data sets and the mean of accuracy is computed.

The stratified Cross Validation with 10 Folds serves as a good mechanism for measuring the accuracy of the classification procedure because of the repeated test and training conducted on the partitioned data set.

Using these metrics, the performance of classification of the segments by the ClaPS is measured.

7.1.3.3 The ClaPS Result Analysis

This section explores the result analysis of the experiments conducted on the ClaPS component. As stated in Chapter IV, the classification of web page segments is carried out using the Decision trees and C4.5 algorithm. The "Weka" machine learning software workbench is utilized to train the decision trees using the features extracted from the segments. This work proposes five features from the web page segments to perform the classification task. The segments are classified into any one of the predefined five segment classes, as the model follows the hard classification category.

The features are extracted from the segments identified by the "SCOPAS Segmentor" component and given as input to the J48 algorithm which is an open source implementation of C4.5 algorithm, in Weka machine learning software workbench. The results of the training data set are tabulated in Table 7.3.

Segment Class	TPR	FPR	Precision	F-Measure	ROC- AUC
Simple Text	0.923	0.011	0.923	0.923	0.951
Image	0.895	0.012	0.944	0.919	0.92
Navigation	0.964	0.027	0.931	0.947	0.981
Head	1	0.025	0.913	0.955	0.988
Audio Video	0.85	0.012	0.944	0.895	0.924
Weighted Average	**0.931**	**0.019**	**0.931**	**0.93**	**0.956**

Table 7.3: Class-wise Results of Training Data Set

In Table 7.3, the five different metrics of classification, True Positive Rate (TPR), False Positive Rate (FPR), Precision, F-Measure and Receiver Operating

Characteristics – Area Under the Curve are tabulated for the training data set. The training data set consisted of 5300 segments from which the 26,500 feature values are extracted. The Attribute-Relation File Format (ARFF) is built using these values and supplied to the decision tree based classifier.

It can be observed from Table 7.3 that the weighted average of True Positive Rate is computed as 0.931 which indicates excellent performance. Similarly the values of Precision and F-Measure are observed as 0.931 and 0.93 respectively which are the indicators of good performance. The ROC – Area Under the Curve which is a critical measure for evaluating the classification performance is observed as 0.956. As the value of ROC is closer towards one which is the maximum limit for the ROC, it indicates an efficient classification. The class-wise results for training data set are illustrated in Fig 7.4.

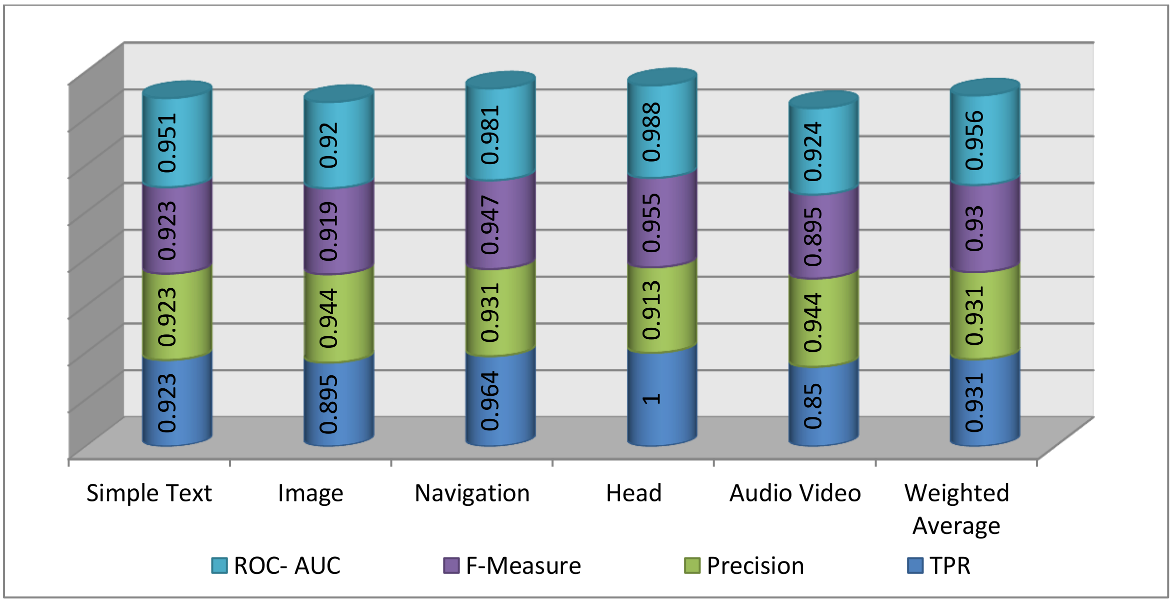


Fig 7.4: Class-wise Results for Training Data

The metrics listed in Table 7.3 are the metrics for individual segment classes. The overall efficiency of the training set classification is shown in Table 7.4.

Measure	Value
Correctly Classified Instances	93.069 %
Incorrectly Classified Instances	6.93 %

Kappa statistic	0.912
Mean absolute error	0.035
Root mean squared error	0.16

Table 7.4: Classification Results for Training Set

In Table 7.4, the value of correctly classified instances (93.069%) indicates the accuracy of classification. The “Kappa Statistics” which is another important measure for classification task is computed as 0.912. This value of 0.912 for Kappa Statistics, which is closer towards one, indicates a better level of agreement.

After the completion of the training process, the Classifier is tested with the test data set. The results of data set are as shown in Table 7.5.

Measure	Value
Correctly Classified Instances	92.307%
Incorrectly Classified Instances	7.692%
Kappa statistic	0.8992
Mean absolute error	0.0383
Root mean squared error	0.1706

Table 7.5: Classification Results for Test Set

In Table 7.5, the value of correctly classified instances (92.307%) indicates the accuracy of classification. The “Kappa Statistics” is computed as 0.8992. The value of 0.8992 which is closer towards one indicates a better level of agreement. For individual segment classes, the predication values across various sessions are as shown in Table 7.6.

With the Training data set, a decision tree model has been built which is used to test the instance data. The experiments were setup in sessions. In each of the sessions, 800 segments are supplied to the model for classification. The prediction values for each class of segments are tabulated in Table 7.6. Totally, ten sessions of experiments were conducted and the mean across all the ten sessions is used as indicator for classification performance. The last row of the Table 7.6, the mean across all the 10 sessions are shown for individual segment classes. It can be observed that the mean prediction values are in the range of 0.88 to 0.93, all of which are closer towards one which empirically validates the accuracy of the classification process proposed by the web page segment classifier, ClaPS.

Session	Simple Text	Navigation	Image	Head	Audio / Video
1	0.876	0.912	0.978	0.876	0.890
2	0.914	0.834	0.980	0.881	0.923
3	0.856	0.898	0.910	0.912	0.945
4	0.801	0.881	0.920	0.934	0.967
5	0.890	0.878	0.930	0.941	0.910
6	0.910	0.890	0.941	0.951	0.967
7	0.932	0.880	0.932	0.961	0.963
8	0.930	0.940	0.931	0.911	0.991
9	0.940	0.881	0.912	0.941	0.912
10	0.930	0.812	0.934	0.934	0.910
Mean	**0.898**	**0.880**	**0.936**	**0.924**	**0.937**

Table 7.6: Mean Prediction Values For Segment Classes

The proposed decision tree based classifier is compared with other models of classification as a comparative measure. The comparison of ClaPS against other classification methods is shown in Table 7.7.

In Table 7.7, the proposed web page segment classification approach ClaPS, is compared with three other classification approaches namely Conjunctive Rule, KStar and Radial Basis Function.

Metrics / Method	Conjunctive Rule	KStar	Radial Basis Function	ClaPS - J48
Correctly Classified	47.524	90.099	92.01	**93.06**
In-Correctly Classified	52.475	9.901	7.99	**6.93**
Kappa Statistics	0.321	0.874	0.937	**0.942**
Mean Absolute Error	0.223	0.044	0.019	**0.035**
Root Mean Squared Error	0.336	0.169	0.14	**0.166**
TP Rate	0.475	0.901	0.95	**0.96**
FP Rate	0.139	0.023	0.01	**0.01**

Precision	0.327	0.905	0.95	**0.931**
F-Measure	0.36	0.902	0.93	**0.93**
ROC - AUC	0.728	0.912	0.925	**0.956**

Table 7.7: ClaPS Comparative Analysis

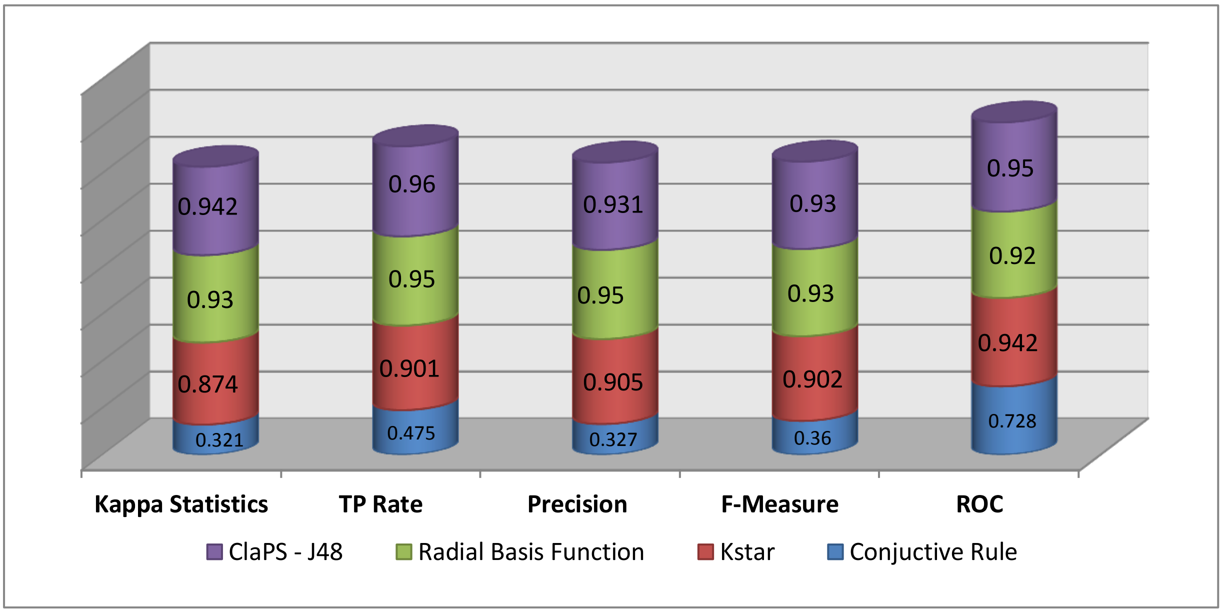


Fig 7.5: Classification Comparative Analysis

The comparison of Kappa Statistics, True Positive Rate (TPR), Precision, F-Measure and ROC – Area Under the Curve measures are given in Fig 7.5.

Apart from this, the proposed SCOPAS model incorporates personalization with the help of user profiles. The evaluation of SCOPAS – Profiler is provided in the following section.

7.1.4 The SCOPAS – Profiler Analysis

The SCOPAS model incorporates a component for user profiles which are built in a multidimensional manner. The SCOPAS Profiler was elaborated in detail in Chapter VI. This section analyses the SCOPAS – Profiler component.

The SCOPAS profiler compiles the “Profile-Bag” with the help four parameters namely “interest”, “item_interest”, “weblog” and “workplaceHomepage”. Among these four parameters, except the “interest”, all other parameters point to resource identifiers. The resource identifiers are resolved and the pages pointed by them are extracted and evaluated to compile keywords.

The analysis of SCOPAS-Profiler is carried by measuring the contribution of the individual parameters towards compiling the keywords. The experiments were conducted with various groups of users which covered the user's skill level from novice to pro. The mean of number of keywords extracted from the individual parameters in each of the group is utilized for measuring the contribution of individual parameters towards profile building. The mean of the keyword count for each user in different groups is indicated as MTIT (Mean of Terms from Item Interest), MTWB (Mean of Terms from Weblog) and MTWH (Mean of Terms from Workplace Homepage). The mean of terms extracted from various sources are as tabulated in Table 7.8.

Group ID	**MTIT**	**MTWB**	**MTWH**
1	13.12	17.32	8.45
2	16.45	18.21	9.47
3	18.45	19.53	11.76
4	14.32	14.38	12.45
5	12.76	16.54	15.35
6	11.28	12.67	12.12
7	22.34	25.23	18.43
8	11.78	23.54	14.34
9	16.43	18.12	16.12
10	14.12	15.13	12.12
11	18.75	19.24	13.13
12	14.68	16.32	14.78
13	13.65	18.43	14.21
14	12.79	16.51	15.31
15	11.33	14.71	14.12

Table 7.8: Mean Profile Terms Count

The chart given in Fig 7.6 illustrates the comparative analysis of profile terms count extracted from three sources as stated above.

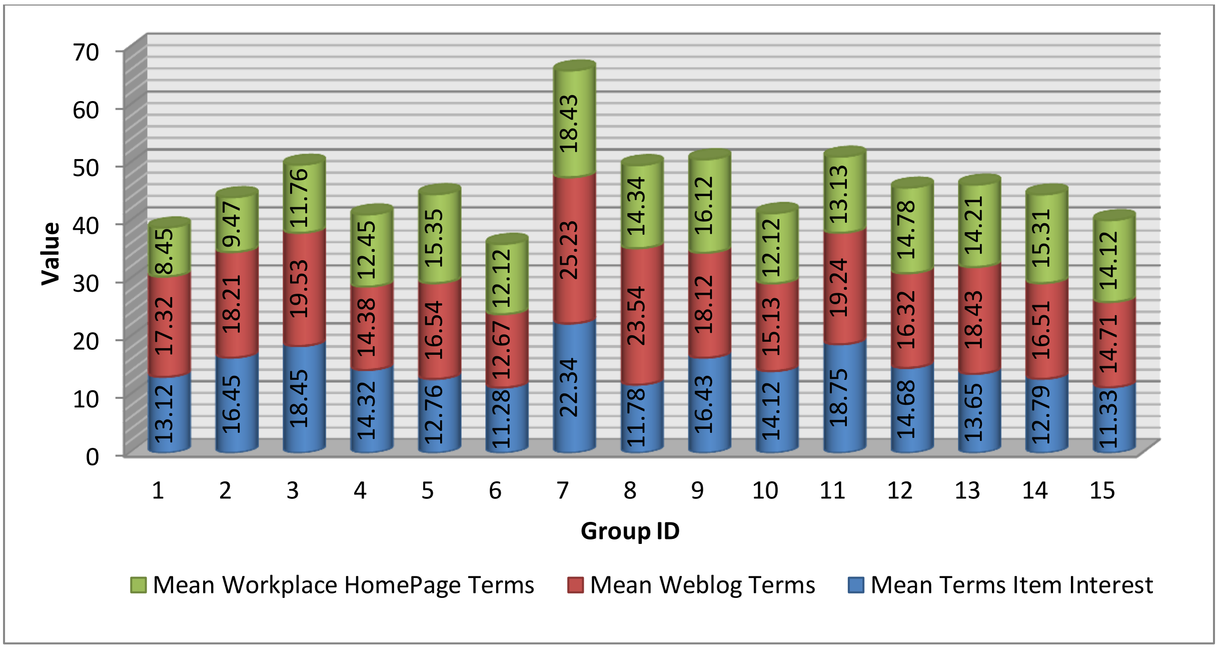


Fig 7.6: Comparison of Term Count from Different Sources

It can be observed from the chart (Fig 7.6) that the mean of terms extracted from the Weblog is larger than the other two sources. The rationale for this behavior is that the other sources are simply pointed by the user whereas the weblog is the page maintained by the users themselves. Hence the increased count of terms extracted from the weblog serves as an added advantage in the representation of the user's preferences.

The SCOPAS profile terms extracted by the model are validated with the user's expectations. The user's ratings on specific profile terms are gathered and the efficiency of the SCOPAS-Profiler is evaluated. For each term extracted by the model, the user specific rating as shown in Table 7.9 is assigned.

Rating	Description
0	Completely Irrelevant
1	Partially relevant
2	Completely relevant

Table 7.9: Profile Term Rating

The mean of rating by user's for profile terms across different user groups is as shown in Table 7.10.

Group ID	CIC (%)	PRC (%)	CRC (%)
1	5.4	30.8	63.8
2	3.2	27.5	69.3
3	5.8	35.8	58.4
4	7.3	37.1	55.6
5	3.1	28.3	68.6
6	4.2	30.1	65.7
7	8.1	32.3	59.6
8	2.1	45.3	52.6
9	3.8	44.3	51.9
10	2.8	26.7	70.5
11	3.5	27.3	69.2
12	4.5	23.8	71.7
13	3.1	22.4	74.5
14	1.3	20.8	77.9
15	2.9	38.8	58.3

Table 7.10: User Rating for Profile Terms

The Chart in Fig 7.7, illustrates the user rating for the profile terms. It can be observed from the chart that, "Completely Irrelevant Count" (CIC) is negligible for the profile terms extracted by the model which proves the efficiency of the profile terms extraction.

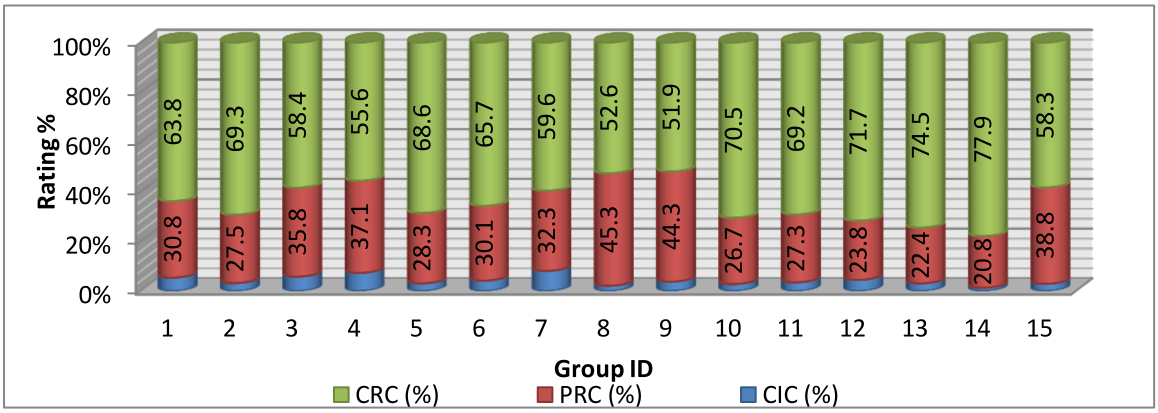


Fig 7.7: User Rating for Profile Terms

In Fig 7.8, the mean of Completely Relevant terms is computed as 64.51%, "Partially Relevant Count" (PRC) is computed as 31.42% and "Completely Irreverent" is

4.07%. The sum of completely relevant (CRC) and partially relevant terms is 95.92% which validates the accuracy of the model in extracting the user profile terms.

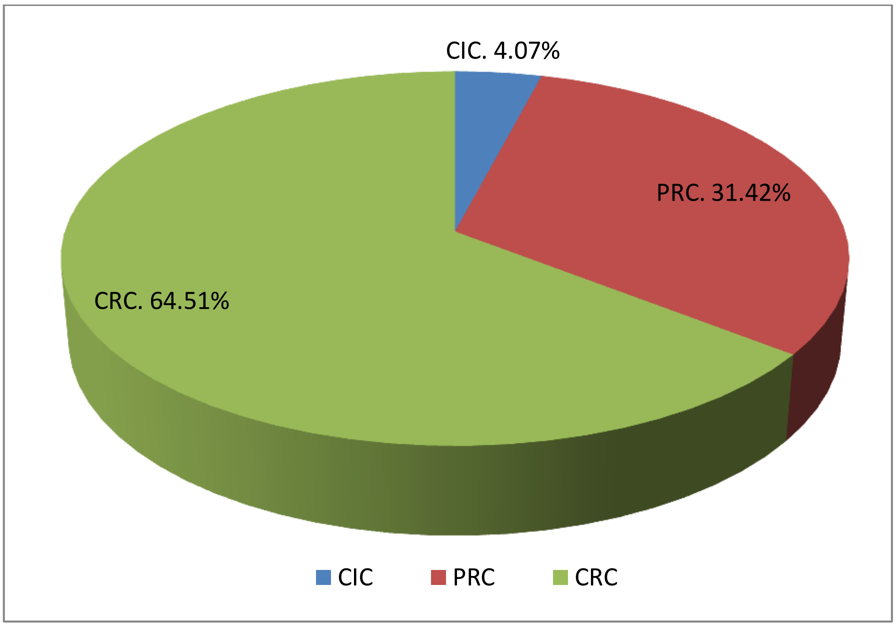
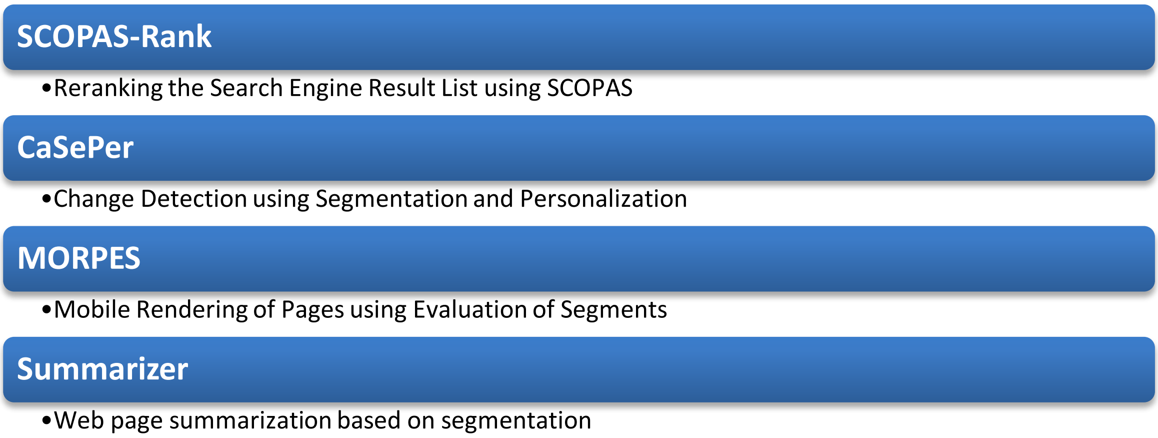


Fig 7.8: Mean of Profile Term Rating

As stated earlier in this Chapter the SCOPAS model is evaluated in two major steps. One is to evaluate the individual components and another is to evaluate the realization of the model in domain specific implementations. This section explored the direct evaluation of these components and the evaluation of model realization of SCOPAS for specific applications is explored in the following section.

7.2 Model Realization

The SCOPAS model is realized with various domain specific objectives which are explored in this section. The domain specific realizations of the SCOPAS model are as illustrated in Fig 7.9. The four realizations mentioned in Fig 7.9 namely, “SCOPAS-Rank”, “CaSePer” and MORPES, utilizes the ideas proposed by the SCOPAS model to achieve a specific goal. These realizations utilize the components of the SCOPAS model. This section explores various aspects of these realizations and their evaluation. The experimental result analysis serves as evidence to the efficiency of the SCOPAS model as well.



SCOPAS-Rank

- Reranking the Search Engine Result List using SCOPAS

CaSePer

- Change Detection using Segmentation and Personalization

MORPES

- Mobile Rendering of Pages using Evaluation of Segments

Summarizer

- Web page summarization based on segmentation

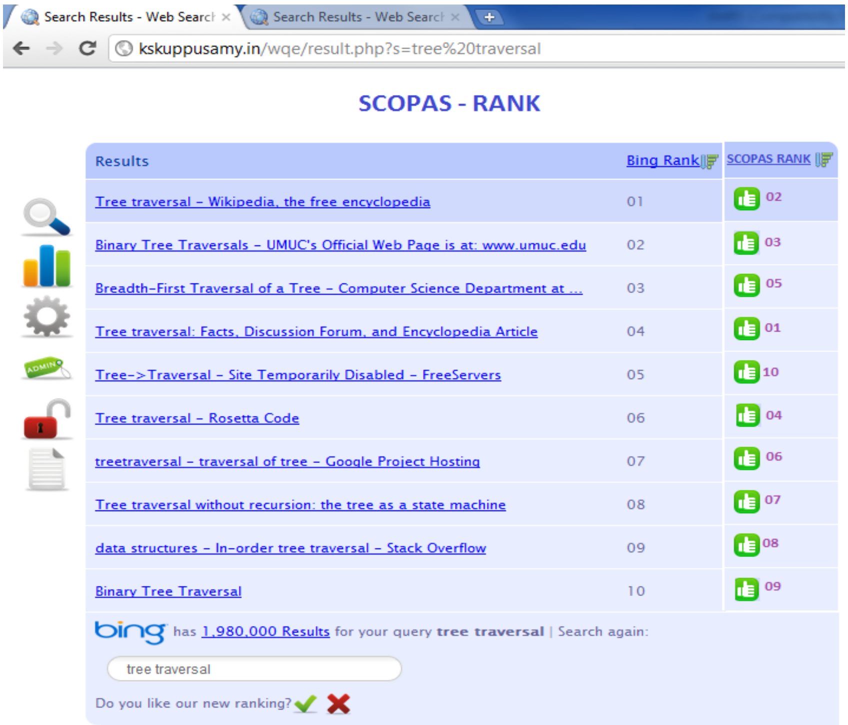
Fig 7.9: SCOPAS Model Realization

As illustrated in Fig 7.9, the SCOPAS-Rank is aimed to re-rank the result list of web search engines with the help of SCOPAS components. The CaSePer (Change detection using Segmentation and Personalization) is focused on detecting the changes in the web pages with the help of segmentation and personalization components of SCOPAS. The "MORPES" (Mobile Rendering of Pages using Evaluation of Segments) is targeted on rendering of web pages in the mobile devices by applying the segmentation and personalization components of SCOPAS. The Summarizer attempts to generate the web page summaries with the help of SCOPAS components. All of these domain specific implementations are explored in this section.

7.2.1 SCOPAS Rank

The objective of the SCOPAS-Rank is to re-rank the results of the web search engines by utilizing the SCOPAS components. The ranking of results is one of the key elements in web information retrieval area. The SCOPAS-Rank evaluates each page against the query and the user interest and ranks them in the descending order of the weight. The implementation included both a regular ranking system and the SCOPAS-Rank for experimentation. In order to establish the model's validity empirically, various experiments were conducted on the prototype implementation.

The SCOPAS-Rank requires each user to register. The proposed SCOPAS model utilizes FOAF (Friend-Of-A-Friend) which is machine readable ontology for maintaining profiles. The user needs to provide his/her Friend-Of-A-Friend (FOAF) ontology file which includes various profile information like the topic in which the user is interested in. On providing the link of the FOAF file the registration process gets completed. The user can submit the query terms in the provided text box.


Search Results - Web Searc
Search Results - Web Searc
kskuppusamy.in/wqe/result.php?s=tree%20traversal

SCOPAS - RANK

Results	Bing Rank	SCOPAS RANK
Tree traversal - Wikipedia, the free encyclopedia	01	02
Binary Tree Traversals - UMUC's Official Web Page is at: www.umuc.edu	02	03
Breadth-First Traversal of a Tree - Computer Science Department at ...	03	05
Tree traversal: Facts, Discussion Forum, and Encyclopedia Article	04	01
Tree->Traversal - Site Temporarily Disabled - FreeServers	05	10
Tree traversal - Rosetta Code	06	04
treetraversal - traversal of tree - Google Project Hosting	07	06
Tree traversal without recursion: the tree as a state machine	08	07
data structures - In-order tree traversal - Stack Overflow	09	08
Binary Tree Traversal	10	09

bing has 1,980,000 Results for your query **tree traversal** | Search again:

tree traversal

Do you like our new ranking? ✓ ✗

Fig 7.10: The SCOPAS RANK Screenshot

The SCOPAS-Rank returns the result page with two sets of ranking. The first one is the result retrieved by a regular web search engine. Any regular web search engine which provides an API can be utilized for this purpose. For the current implementation the Bing API is used. The results can be re-ranked using SCOPAS-Rank by clicking on the icon provided. This process is shown in Fig 7.10. In the SCOPAS rank column an icon with a thumps-up symbol in green color is provided when the segmentation is successful; a red color icon is provided when it is not.

By clicking on the green color icon provided in the SCOPAS-Rank column, the corresponding page's segmentation can be viewed. This is shown in Fig 7.11. The green color lines indicate the segment boundaries. In the segmented page, the query terms present are given a unique style with a highlight. It facilitates easier identification of the segments which consists of query terms. In addition to this in-place segmentation view, the implementation also provides a separate "box view" of the segments, in which all the segments of the page are displayed one after another in a vertical alignment.

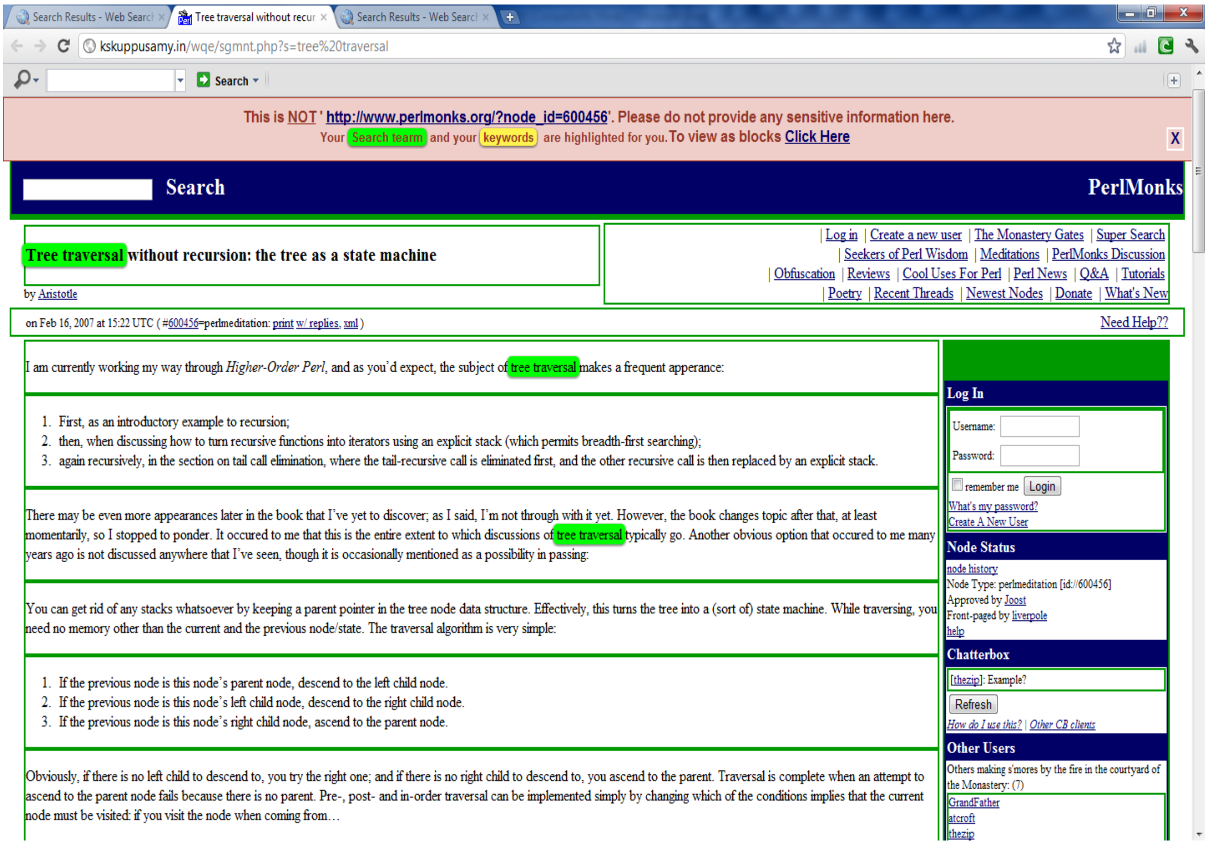
Experiments were conducted on this prototype implementation with a group of fifteen volunteers. The groups of volunteers were selected such that it covers various skill levels. The volunteers consisted of people who are doing their schooling, persons from under graduate course, persons pursuing post graduate courses and scholars doing their Ph.D research.

The experiments were setup such that each user is asked to register their profile in order to incorporate personalization in the ranking process. For this registration process, they are presented with a Web Form with fields asking for their "interest terms", web pages in which they are interested in, weblog, "workplace home page" etc. Submitting this form would automatically create the FOAF representation.

The experiments were conducted in sessions. In each session, they are asked to enter query terms and rate the results at each position in the ranking as shown in Table 7.11. Based on the rating provided by the user, evaluation of the ranking is done with the help of NDCG (Normalized Discounted Cumulative Gain). The query terms entered by the users were selected such that it covers both profile and non-profile terms. The rationale for such an act is that if the evaluation is done with only the query terms which are a sub set of the profile terms, it would lead to a biased output. The average length of the search query in each session is 2.4 terms which is matching with the global average.

The reasons for conducting experiments in various sessions are as follows: to avoid the loss of concentration while rating the results, to derive conclusions from a larger set. In a session, each volunteer is asked to enter twenty different queries. For each resultant rank list, the volunteer has to rate the top ten results which makes two hundred ratings per session. Hence in one session there were 3000 ratings for fifteen volunteers. In total, sixteen sessions were conducted leading to 48000 overall ratings.

The volunteers were asked to submit the queries, and rate the individual results and their ordering. The measure of effectiveness used in the experiments was "Normalized Discounted Cumulative Gain (NDCG)". (Järvelin and Kekäläinen, 2002, 2000). The normalized discounted cumulative gain is selected over other measures like Mean Average Precision (MAP) because of the fact that NDCG can accept more values, than simple binary relevant judgments of "relevant" , "irrelevant". The users can mark their satisfaction level in a scale of 4 starting from 0 to 3. The value and their meaning are displayed in Table 7.11. The score of “1” which indicates “unable to decide relevancy” is incorporated because the user may sometimes be not able to come to a conclusion whether this result is relevant or not. In order to incorporate such conditions this scoring level “1” is included.


Search Results - Web Search
Tree traversal without recur
Search Results - Web Search
kskuppusamy.in/wqe/sgmnt.php?s=tree%20traversal
Search
Search
PerlMonks
| Log in | Create a new user | The Monastery Gates | Super Search
| Seekers of Perl Wisdom | Meditations | PerlMonks Discussion
| Obfuscation | Reviews | Cool Uses For Perl | Perl News | Q&A | Tutorials
| Poetry | Recent Threads | Newest Nodes | Donate | What's New
Need Help??
Log In
Login
Refresh
Node Status
Chatterbox
Other Users

This is NOT ' http://www.perlmonks.org/?node_id=600456'. Please do not provide any sensitive information here.
Your Search tearm and your keywords are highlighted for you. To view as blocks Click Here X

Tree traversal without recursion: the tree as a state machine

by Aristotle

on Feb 16, 2007 at 15:22 UTC (#600456=perlmeditation: print w/ replies, xml)

I am currently working my way through *Higher-Order Perl*, and as you’d expect, the subject of tree traversal makes a frequent apperance:

1. First, as an introductory example to recursion;
2. then, when discussing how to turn recursive functions into iterators using an explicit stack (which permits breadth-first searching);
3. again recursively, in the section on tail call elimination, where the tail-recursive call is eliminated first, and the other recursive call is then replaced by an explicit stack.

There may be even more appearances later in the book that I’ve yet to discover; as I said, I’m not through with it yet. However, the book changes topic after that, at least momentarily, so I stopped to ponder. It occured to me that this is the entire extent to which discussions of tree traversal typically go. Another obvious option that occured to me many years ago is not discussed anywhere that I’ve seen, though it is occasionally mentioned as a possibility in passing:

You can get rid of any stacks whatsoever by keeping a parent pointer in the tree node data structure. Effectively, this turns the tree into a (sort of) state machine. While traversing, you need no memory other than the current and the previous node/state. The traversal algorithm is very simple:

1. If the previous node is this node’s parent node, descend to the left child node.
2. If the previous node is this node’s left child node, descend to the right child node.
3. If the previous node is this node’s right child node, ascend to the parent node.

Obviously, if there is no left child to descend to, you try the right one; and if there is no right child to descend to, you ascend to the parent. Traversal is complete when an attempt to ascend to the parent node fails because there is no parent. Pre-, post- and in-order traversal can be implemented simply by changing which of the conditions implies that the current node must be visited: if you visit the node when coming from…

Username:
Password:
□ remember me
What's my password?
Create A New User

node history
Node Type: perlmeditation [id://600456]
Approved by Joost
Front-paged by liverpole
help

[thezip]: Example?
How do I use this? | Other CB clients

Others making s'mores by the fire in the courtyard of the Monastery: (7)
GrandFather
atcroft
thezip

Fig. 7.11: The SCOPAS Segmentation view

Score	**Meaning**
0	Completely irrelevant
1	Unable to decide relevancy
2	Partially relevant
3	Completely relevant

Table 7.11: Result Rating By User

The Discounted Cumulative Gain is calculated using the formula as shown in Eq. (7.10)

$$DCG_P = rel_1 + \sum_{i=2}^{P} \frac{rel_i}{\log_2 i} \tag{7.10}$$

In (7.10) rel_i indicates the relevance at position “i". The Normalized Cumulative Gain is calculated as shown in Eq. (7.11).

$$NDCG = \frac{DCG}{IDCG} \tag{7.11}$$

In-order to make the experiments real-time, each user was asked to focus only on the top ten results returned. This decision was taken to adapt to the real world search behavior of the users, where the users mostly focus on the first page of the result listing. The experiments were conducted in sessions. The measure of effectiveness is tabulated in Table 7.12. The results in the table are records for fifteen sessions. The DCG and DCG SCOPAS indicate the discounted cumulative gain with the regular results and SCOPAS results respectively. The IDCG is the measure of DCG for ideal condition. The user has to rate the ideal result ranking for the query.

Session	**IDCG**	**DCG**	**DCG-SCOPAS**	**NDCG**	**NDCG SCOPAS**
1	11.084	9.359	9.769	0.844	0.881
2	10.956	9.768	9.835	0.892	0.898
3	9.885	8.456	8.678	0.855	0.878
4	10.986	9.456	9.859	0.861	0.897
5	11.256	9.878	9.972	0.878	0.886
6	12.125	9.895	10.135	0.816	0.836
7	9.125	7.568	7.963	0.829	0.873
8	9.965	8.563	9.012	0.859	0.904
9	10.351	8.458	9.135	0.817	0.883
10	9.135	8.435	8.835	0.923	0.967
11	10.215	9.325	9.568	0.913	0.937
12	9.632	7.325	8.125	0.76	0.844
13	11.235	9.145	10.125	0.814	0.901
14	8.635	7.658	7.995	0.887	0.926
15	10.258	8.125	9.225	0.792	0.899

Table 7.12: SCOPAS-Rank Effective Measures

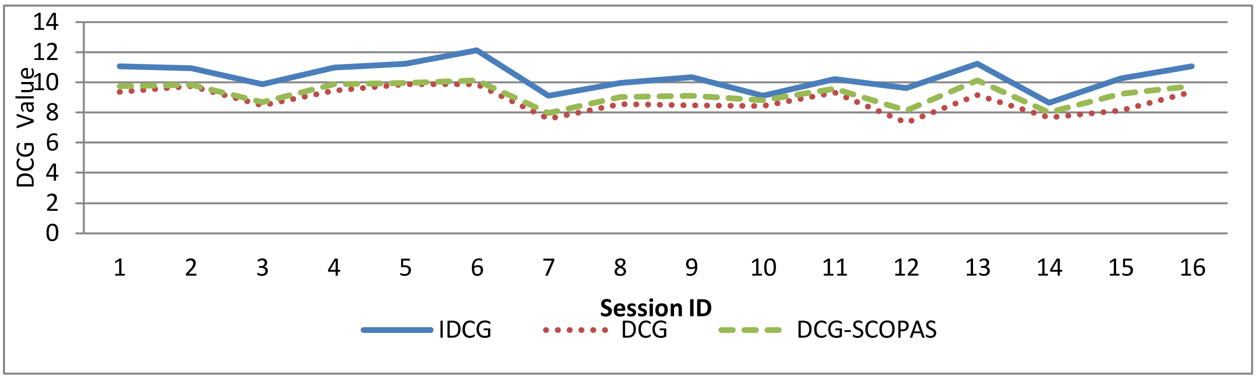


Fig 7.12: SCOPAS Rank DCG vs IDCG

The NDCG is the normalized cumulative gain which is the result of division of DCG by NDCG. The results are charted out in Fig. 7.12 and Fig. 7.13.

It can be observed from the experiments with various sessions that the Normalized Discounted Cumulative Gain for the proposed SCOPAS-Rank model showed an encouraging result of 89.4 % as a measure of effectiveness. The average NDCG across the sessions was observed as 0.894. As NDCG is the result of division of DCG by IDCG, the ideal result ranking system would yield a value of 1 for NDCG. The observed NDCG value 0.894 for the SCOPAS-Rank is the empirical evidence for the validity of the proposed SCOPAS – Rank model. The Normalized Discounted Cumulative Gain chart in Fig. 7.13 illustrates the fact that the NDCG of the proposed SCOPAS-Rank is higher.

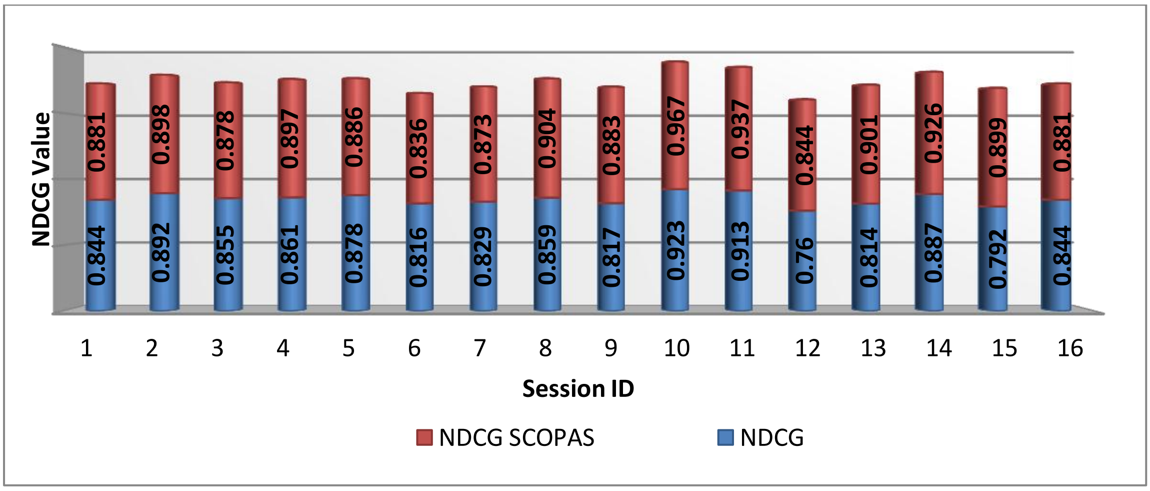


Fig 7.13: SCOPAS –Rank NDCG Comparison

The SCOPAS-Rank utilized the SCOPAS model to re-rank the search engine result listing and the re-ranking results were evaluated with metrics like NDCG, DCG etc. The incorporation of user preferences through profiles made the re-ranked results match the user's expectation closely than the initial ranking which got established through the improved metrics values for the DCG and NDCG. The SCOPAS realization in the change detection domain is explored in the following section.

7.2.2 The CaSePer – Change Detection using SCOPAS

The CaSePer (Change detection using Segmentation and Personalization) is a method to detect both structural and content changes in the web pages. The changes in the web pages can either be content changes or structural changes (Flesca and Masciari, 2003).

The structural changes are primarily the changes which occur at the template level. The changes in the placement of elements are also covered under the structural changes. The content changes include modification, insertion or deletion of hypertext elements. The users navigate to web pages of their interest. A user interested in a particular topic might visit the web pages related to that topic at regular intervals of time. Such users might be interested in knowing the recent changes happened in that web page, rather than the entire web page. The change detection in web pages is a complex process. Due to large amount of nodes in HTML tree, the change detection requires innumerous comparisons, leading to a NP hard problem (Chawathe and Garcia-Molina, 1997; Chawathe et al., 1996).

The temporal dynamism exhibited in the web contents has been studied in detail by various researchers. AT&T Internet Difference Engine (AIDE) is a change detection system which employs a token based approach (Chen and Koutsofios, 1998). AIDE works on HTMLDiff (Douglis and Ball, 1996) which is based on the Longest Common Subsequence (LCS) algorithm (Daniel, 1997). AIDE displays the changes between the current version and the immediate last version. The AIDE system compares the versions of complete web page to detect the changes, whereas the proposed CaSePer model reduces the problem space by narrowing down the comparison region without compromising the quality of change detection process.

The change detection approach which considers only the latest two versions of the page was given by WebCQ(Liu et al., 2000). At the same time, the WebCQ incorporates user specific notifications. The proposed CaSePer model incorporates the personalization process and multi-hop signature matrix to consider various versions of the page.

A dataflow based approach to "change detection" was proposed in WebVigiL(Jacob et al., 2004). One of the important parameters in the change detection system is the revisit time interval. The revisit interval can either be given explicitly or to be learned by the system automatically. The WebVigiL system facilitates both these approaches. The proposed CaSePer model incorporates a category based revisit interval policy. An improved change detection system by restricting the number of comparisons was also explored (Artail and Fawaz, 2008). The work states that the change detection can be performed by comparing only the similar tag types and by utilizing hashing. The proposed CaSePer model employs the web page segmentation followed by hashing technique to reduce the number of comparisons significantly.

The "change detection" is attempted with the help of "edit-script" which refers to steps involved in converting one version of the document tree to another. The XyDiff(Cobena et al., 2002) algorithm provides a solution for change detection problem using edit scripts. But the sequence generated by XyDiff need not be optimal always. The X-Diff (Wang et al., 2003) ensures the optimal differences but it can't handle the move operations. Normally the change detection approaches create a change representation file called delta file. The delta file visualization using style sheets was proposed by (La Fontaine, 2001).

The CaSePer utilizes a variation of X-Diff in combination with a node sequence based algorithm to handle both content and structural changes. The CaSePer utilizes the SCOPAS-Segmentor component to segment the initial version of the web page. For the successive versions of the same web page, segmentation is carried out by using the node boundary based approach. In this technique, the boundaries of segments are identified with the help of DOM tree nodes, which formed the boundaries in the previous version. As the boundary nodes themselves can get removed or relocated, a cascaded node sequence from the segments of earlier versions are used to mark the segment boundaries. The sequence of nodes in the earlier versions is utilized to mark the segment boundary. As the sequence can change across versions due to structural modification, the approach is adopted in combination with the boundary node. This method is followed so that the number of segments in all versions of the page remains the same.

A hashing function is applied to calculate the hash value of each segment. In the proposed implementation, the hash function chosen is the MD5 (Rivest, 1992). The reason for selection of MD5 is that it will never produce the same hash value for two inputs, even with the slightest variation in data. The previous snapshots of the web page Ω in the temporal dimension is fetched from the Page Evolution Track (PET). The Page Evolution Track is an indexing structure, with a temporal capability, which holds the snapshots of the previous versions of the web page. The segments are built for the pages in PET and their hash value is computed. The segments with non-identical hash values along the temporal dimension are selected for change detection which reduces the complexity of the problem to a greater extent by narrowing down the regions to be compared.

In order to introduce personalization, into the process of change detection, the profile built by the SCOPAS-Profiler is utilized. This facility of identifying the user specific changes in the web page would make it convenient for the user to easily locate the changes, which might be relevant to one's interest, in much simpler and efficient manner. At the implementation level, these user specific changes shall be rendered with a unique style definition which is different from general changes, in-order to catch the user's attention.

Session ID	MSC (Mean Segment Count)	MPETL (Mean Page Evolution Track Length)	MSM (Mean Segments Modified)	MSMP (Mean Segments Modified With Personalization)	MSUD (Mean Segments UnDetected)	SDP (Successful Detection Percentage)
1	23.12	4.3	3.1	1.2	0.43	98.14
2	18.45	5.2	4.1	2.3	0.78	95.772
3	17.45	5.3	3.2	2.6	0.23	98.682
4	20.32	4.3	4.4	3.1	1.1	94.587
5	18.76	3.5	1.2	0.5	0.45	97.601
6	21.08	5.6	3.2	2.6	0.35	98.34

7	20.45	4.2	4.5	2.5	0.21	98.973
8	20.32	3.5	4.3	3.1	0.33	98.376
9	19.34	4.1	3.5	1.2	0.21	98.914
10	24.12	5.1	3.7	1.1	0.56	97.678
11	17.65	5.2	7.8	3.8	0.87	95.071
12	16.78	5.3	6.5	4.1	0.23	98.629
13	17.65	4.4	4.2	2.1	0.4	97.734
14	14.89	4.5	5.3	3.1	0.33	97.784
15	13.23	4.7	4.1	1.3	0.55	95.843

Table 7.13: Segment Component Mean Values

The MSM with general changes across the session is observed as 4.2. The mean of overall segment count in the experiments is 18.9. By the incorporation of segmentation based hash value comparison, the number of segments to be considered by the tree comparison was reduced by 77.8%. The MSMP across the sessions is observed as 2.3. The segments with user specific modifications estimated for 54.77% of overall changes which got rendered with unique style for the easy identification of the user. This 54.77% emphasizes the importance of incorporating the personalization in the change detection process. The Mean of the Successful Detection Percentage across the session is 97.45 which validate the robustness of the proposed change detection model.

The chart in Fig 7.14, depicts the time taken for various components of the CaSePer model like segmentation, content change detection, structural change detection, profile term highlighting and rendering for web pages of various sizes. The chart depicts the mean values of experiments conducted in 15 sessions. In the X-axis the mean page size is plotted and Y-axis denotes the time in seconds. The segmentation method adopted in the CaSePer model is optimized for speed rather than the semantics.

It can be observed from Fig 7.14, that the mean of segmentation time is 0.2 seconds which confirms the efficiency of the segmentation process associated with the CaSePer. With respect to the change detection, the time to find out changes varies according to the amount of change a web page has gone through.

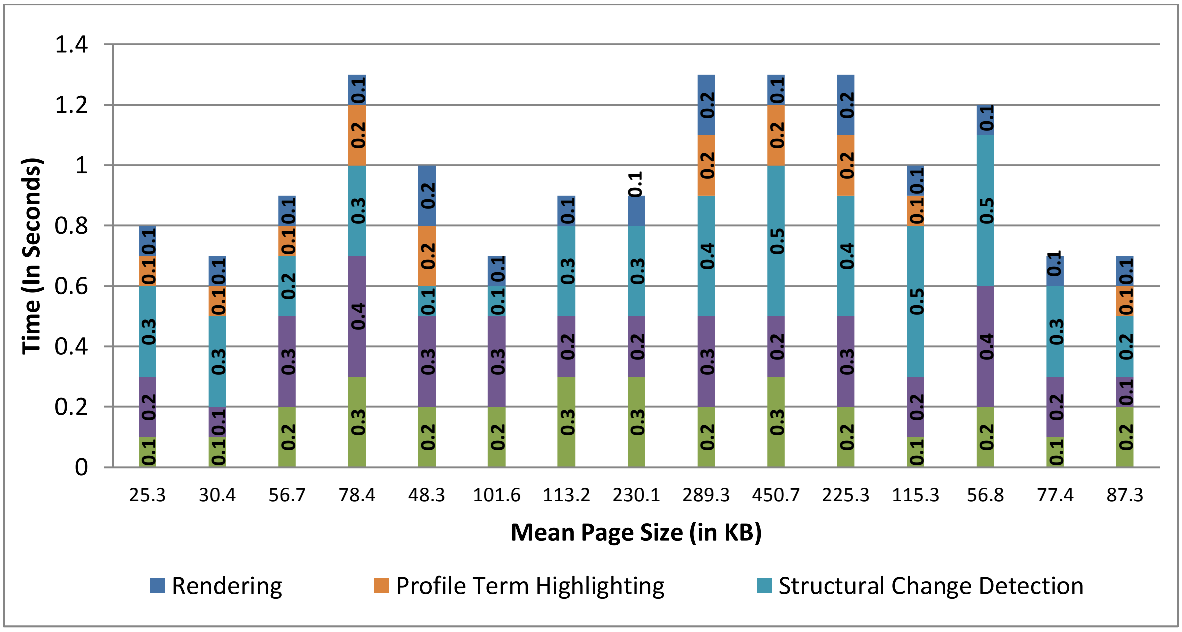


Fig 7.14: CaSePer Time Analysis

In the experiments, the values are measured at ~5% change. The mean of content and structural change detection time is observed as 0.24 and 0.31 seconds. Earlier studies (Wang et al, 2003) have reported a change detection time of ~1 second for documents of ~80 Kilo Byte size at 5% change. In the CaSePer model the mean of change detection by combining both structural and content changes is observed as 0.55 seconds which serves as an empirical evidence for the efficiency of change detection process in the CaSePer.

7.2.3 Mobile Rendering

The SCOPAS model components are realized using the "mobile rendering" domain which is termed as "MORPES" (Mobile Rendering of Pages using Evaluation of Segments).[6] The rendering of web pages in smaller screen portable devices is an interesting research topic which has been explored by various studies.

The approach followed in the MORPES, utilizes the SCOPAS-Segmentor for segmentation of web pages. It incorporates the personalized segment evaluation component "SCOPAS-Evaluator" for choosing the segments to display. The MORPES, renders the specific segments of a web page in the small screen display by

[6] The core idea proposed by MOREPS is recognized as "Disruptive Innovation" by L'Atelier http://www.atelier.net/trends/articles/optimiser-pages-web-mobile-ne-oublier-personnalisation

segmenting the page. The segments are chosen in the decreasing order of their score. The top “n” segments are displayed initially in the first shot.

The experiments were conducted in fifteen different sessions and the results are tabulated in Table 7.14. In Table 7.14 MSC stands for mean segment count which indicates the mean of the number of segments in that session. MSFS stands for Mean of Segments in First Shot, and MPSC stands for mean of page shot count. In Fig 7.15, the chart depicts the comparison of various segment parameters while rendering the web page contents on mobile devices. It was observed from the experiments across the sessions that the mean of segments at the first shot was 4.06 whereas the mean of segments was 19.13. The segments which carry the highest score with respect to the user’s informational requirements only are displayed in the first shot there by giving a clear view of the data that user might be interested.

Session ID	**MSC**	**MSFS**	**MPSC**
1	25.21	5.3	5.1
2	16.15	4.2	7.1
3	18.35	2.3	4.2
4	22.37	1.3	5.4
5	20.11	4.5	3.2
6	23.38	5.2	6.2
7	21.45	3.3	6.5
8	22.32	5.5	7.3
9	18.64	6.1	5.2
10	22.32	5.3	1.7
11	16.15	1.2	1.8
12	17.58	4.3	3.5
13	17.65	3.5	1.2
14	13.21	4.3	3.3
15	12.10	4.6	3.1

Table 7.14: MORPES – Segment Rendering

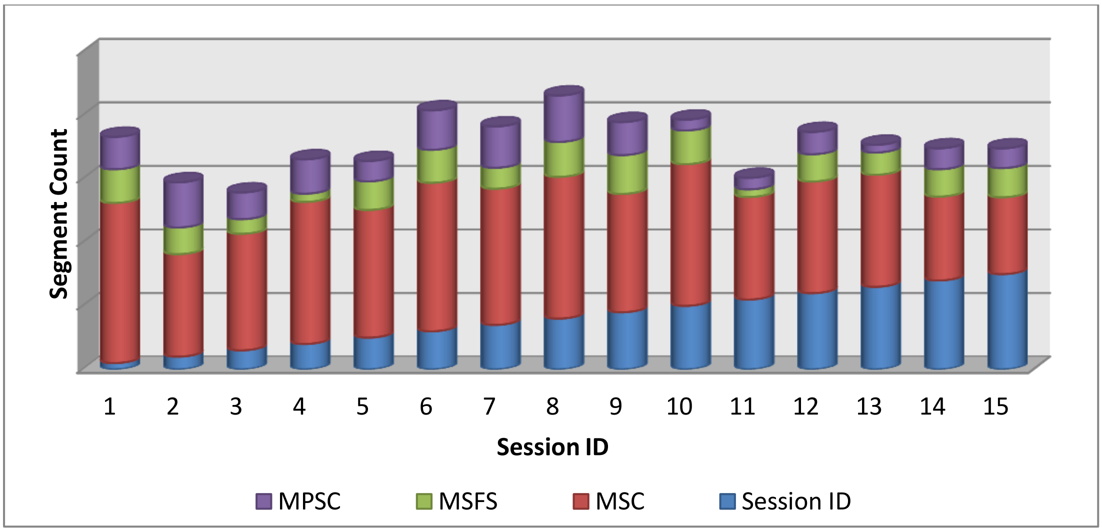


Fig 7.15: MORPES – Comparison of Segment Parameters

From the experiments conducted it is also observed that users were able to find their needy contents in the first few shots of the page itself rather than going deep through the segment buffer. This core idea solves the problem of providing the information relevant to the user in the best possible way with the available screen size.

7.2.4 Web Page Summarization

The process of web page summarization differs from the traditional text summarization due to the inherent features in the structure of web pages comparing with normal documents. The SCOPAS model components are utilized for web page summarization. The summarizer is based on the segmentation approach. The proposed model performs an “inclusive summarization” by representing entities from different portions of the web page resulting in the miniature of the original page, termed as “Micro-page”. With the incorporation of personalization in the summarization process, the micro-page can be tailored based on the user preferences.

The web page is initially segmented using the SCOPAS-Segmentor. The summarizer has to take these segments as input. The summarization process is carried out on these segments individually. The summarization task on each of these segments is carried by incorporating four critical factors: Segment Weight, Luhn’s Significance Factor, Profile Keywords and Compression ratio, as illustrated in Fig 7.16. The segment weight is computed by the SCOPAS-Evaluator component.

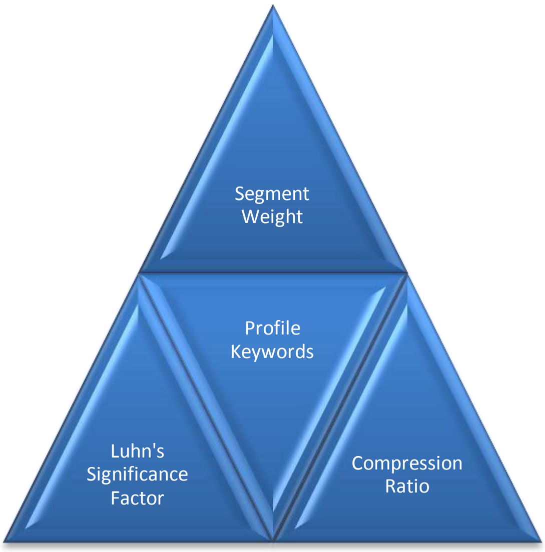


Fig 7.16: Summarization Factors

The Luhn's formula is utilized to calculate the importance of sentences present in a document based on the distance measure of important words in that document. The proposed model utilizes the Luhn's significance factor to select important sentences from the segment. The profile keywords are fetched from the profile-bag which is built by the SCOPAS-Profiler. The compression ratio specifies the summarization level. The core idea of the summarizer is to gather the components from all sections of the web page. The summarization is evaluated with the help of various metrics which measure the micropage in terms of link count, image count etc. The experiments on the summarizer are conducted on various sessions and the mean of metrics for each session is tabulated in Table 7.15.

Session	SSP	SMP	ISP	IMP	LSP	LMP
1	25.3	20.13	3.1	2.02	7.8	5.85
2	23.2	18.43	4.1	2.67	12.4	9.30
3	19.7	18.2	2.1	1.37	13.5	10.13
4	16.4	14.2	5.3	3.45	12.5	9.38
5	15.2	13.4	6.2	4.03	13.2	9.90
6	10.3	10.1	5.5	3.58	6.5	4.88
7	11.1	9.8	4.7	3.06	4.5	3.38

8	10.1	9.5	7.5	4.88	3.8	2.85
9	17.1	16.3	6.5	4.23	4.7	3.53
10	16.1	15.2	6.8	4.42	6.7	5.03
11	16.5	14.7	6.9	4.49	4.8	3.60
12	23.1	20.1	6.8	4.42	4.8	3.60
13	19.3	18.7	6.9	4.49	4.3	3.23
14	18.2	16.3	4.5	2.93	5.4	4.05
15	16.5	14.5	4.2	2.73	3.8	2.85

Table 7.15: Summarization Metrics Values

In Table 7.15, the column SSP indicates Segments in Source Page, SMP indicates Segments in Micro-Page, ISP indicates Images in Source Page, IMP indicates Images in Micro-Page, LSP indicates Links in Source Page and LMP indicates Links in Micro-Page.

The values listed in the Table 7.15 are for 70% summarization. It can be observed that the micro-page holds contents from 88.9% of segments from the source page. The ratio between the images in the source page and the micro-page is observed as 0.65; the ratio between the links in the source page and the micro-page is observed as 0.75. Hence it confirms that the summarizer grabs the contents from various segments and the intra-segment level components are also summarized at the proper ratio.

Apart from the above mentioned four realizations of the model, the SCOPAS is utilized for other applications like segmentation based content filtering (Kuppusamy K.S and Aghila G, 2012a), dynamic page construction from search results (Kuppusamy K.S and Aghila G, 2011) and semantic snippet construction (Kuppusamy K.S and Aghila G, 2012b)with the help of segment evaluation.

This Chapter explored various experiments conducted on the SCOPAS components and the realization of the SCOPAS model. Both the components and the individual realizations of the SCOPAS model are evaluated with the help of various metrics. These experiments serve as the experimental validation for the approach proposed by the SCOPAS model. The Conclusions derived from this research work and the future directions are explored in the following Chapter.

Chapter VIII: Conclusions and Future Directions

This thesis explored a multimodal approach to web content scoring with the help of segmentation, classification and personalization. The conclusions derived from this thesis and the future directions for this research work are discussed in this Chapter.

8.1. Conclusions

This work is aimed to address the challenges in scoring the contents of web pages with respect to the user's information requirement context. This is achieved with the help of modeling the web pages by segmenting them into smaller blocks and evaluating the individual blocks in a multimodal manner. The process is enhanced further by incorporating the segment class weight which facilitates a variable magnitude approach for evaluating different segments of the page based on various structural semantics. One more layer of enhancement is achieved by personalizing the scoring procedure by incrementally amalgamating the user profiles.

This work approached the problem of content scoring by web page modeling, which is explored through four research questions presented in Chapter I. This thesis answered all the four research questions by proposing the SCOPAS model. The individual components of the SCOPAS model were aimed towards solving the issues posed by the research questions.

The "Research Question-1" is concerned with the granularity of scoring. The content scoring techniques generally operates at the level of complete web pages. Each web page is considered to be an atomic unit. This thesis proposed the fine-graining of content scoring with the help of web page segmentation. The segmentation technique proposed in this research work belonged to the hybrid category which incorporated page tree and densitometry. The hybrid segmentation method proposed by this research work harnessed both the in-memory representation of pages and the slope variation in density of contents in the web pages to mark the segment boundary. The segmentation technique proposed in this research work is evaluated with the help two different metrics namely, Adjusted Rand Index (ARI) and the Normalized Mutual Information (NMI) which confirmed the efficiency of segmentation with the values of 0.67 and 0.77 respectively.

The “Research Question-2” is concerned with the incorporation of differential approach for various intra-page components. This question is answered by this research work by proposing a segment classification method called ClaPS (Classification of Page Segments). The ClaPS proposed five different types of segments namely “Simple Text”, “Image”, “Navigation”, “Head” and “Audio/Video”. Each segment of the web page is classified into any one of the above specified types by extracting various segment features. This thesis encompassed five intra-segment features namely “Text Ratio”, “Link Ratio”, “Image Count”, “Head Ratio” and “Object Count”. The web page segment classification approach utilized the J48 decision tree classifier from Weka machine learning software workbench. The classifier is evaluated with the help of various metrics like “True Positive Rate”, “Precision”, “Kappa Statistics” and “ROC” etc.

The experimentation methodology adopted to evaluate the classifier is “Stratified Cross Validation with 10-Folds”. The decision tree classifier is trained with the features specified above. The values of the metrics confirmed the accuracy and efficiency of the classification approach proposed in this thesis. The Kappa Statistics value of 0.899 confirmed the encouraging classification agreement. The ROC – Area Under the Curve is another important measure for classification whose value is observed as 0.956 which is an empirical evidence to the efficiency of the classification process. The performance of the proposed classification is compared with other techniques and the experimental results confirmed the efficiency of ClaPS.

The “Research Question-3” is concerned with the evaluation of segments through a multimodal approach. This thesis proposed a multimodal segment evaluation model termed “MUSEUM” which utilized 6-dimensions in computing the score of segments. The content scoring component is evaluated with the help of realization of the model termed “SCOPAS-Rank”. The evaluation of SCOPAS –Rank is carried out with the help of various metrics like DCG (Discounted Cumulative Gain) and NDCG (Normalized Discounted Cumulative Gain). The values of DCG and NDCG are compared using search engine result re-ranking problem. The Normalized Discounted Cumulative Gain value of 0.894 confirms the efficiency of the proposed SCOPAS-Evaluator component. The DCG values are compared with the Ideal Discounted Cumulative Gain to gauge the rank-list computed by SCOPAS-Evaluator.

The “Research Question-4” is concerned with incorporation of personalization in the content scoring procedure. This thesis has proposed an incremental profile amalgamation approach which builds the profiles using FOAF and user action monitoring. The SCOPAS profiler is analyzed using metrics like MTIT (Mean of Terms from Item Interest) and MTWB (Mean of Terms from Weblog). The profile terms extracted using the SCOPAS-profiler is rated by the users using a three-scale rating. The results of the experiments confirmed that 64.51% of profile terms are completely relevant, 31.42% are partially relevant and 4.07% are completely irrelevant. The sum of completely relevant and partially relevant terms is 95.92% which validates the accuracy of the model in extracting the user profile terms.

The proposed SCOPAS model is realized with domain specific applications to gauge the usability of the proposed model to solve application specific problems. This thesis presented four domain specific realizations of the proposed SCOPAS model namely, “SCOPAS-Rank”, “CaSePer”, “MORPES” and “Summarizer”. The SCOPAS-Rank is a realization of the model in the search engine result list re-ranking. The CaSePer (Change detection using Segmentation and Personalization) is targeted on the change detection problem in various versions of the same web page. The MOREPS (Mobile Rendering of Pages using Evaluation of Segments) is a realization of the SCOPAS model to render the web pages on small screen mobile terminals using segmentation and personalization. The “Summarizer” is utilized to summarize the web pages using web page segmentation.

All the above specified realizations of the SCOPAS model are evaluated using specific metrics and the robustness of these realizations are confirmed with encouraging results with experiments conducted on these realizations. The conclusions derived from the experiments conducted on domain specific realizations and individual components of the SCOPAS model are listed below:

- The SCOPAS model enriches the web content evaluation process by incorporating a multidimensional, variable magnitude approach.
- The machine learning based segment classification enables a differential approach to various structural elements.

- The incremental profile amalgamation using FOAF, user action monitoring and segment evaluation strengthens the content scoring process.

The future directions for this research work are explored in the following section.

8.2. Future Directions

The multimodal web page modeling for content scoring based on segmentation and incremental profile amalgamation approach proposed by this thesis can be enhanced further with the following future directions:

- Incorporating specialized parsers for handling a wide array of content types. In addition to the normal web pages, different types of contents available in the web horizon like Portable Document Format (PDF), flash content etc., shall be incorporated into the evaluation component of the proposed model.

- Incorporating NLP techniques in the evaluation process. The content semantics can be further enhanced by the incorporation of Natural Language Processing techniques in the evaluation process.

- Introducing image understanding techniques in evaluating the images instead of relying only on meta-data. The image analysis using digital image processing techniques would provide further insight into the understanding of image-semantics to enhance the Image Weight Coefficient of the SCOPAS-Evaluator.

- Incorporating more intra-segment features in the classification approach, in addition to the currently proposed five features. This would facilitate in increasing the spectrum of segments to be classified.

- The SCOPAS-Profiler component shall be further improved by a collaborative keyword gathering with the help of linked FOAF files. In addition to this, additional fields from FOAF specification can be utilized in building the initial profile of the users which would facilitate in enhancing the capturing of preferences of the users. In addition to this, the profile term unlearning shall also be introduced. Another dimension in the profile enhancement is the incorporation of Short-Term and Long-Term interest in addition to the normal preferences in-order to encompass the context specificity.

REFERENCES

Adamic, L.A., Buyukkokten, O., Adar, E., 2003. A social network caught in the web. First Monday 8 (6).

Adar, E., Karger, D., Stein, L.A., 1999. Haystack: Per-user information environments, in: Proceedings of the Eighth International Conference on Information and Knowledge Management. ACM, pp. 413–422.

Ahmadi, H., Kong, J., 2008. Efficient web browsing on small screens, in: Proceedings of the Working Conference on Advanced Visual Interfaces. ACM, pp. 23–30.

Akpinar, E., Yesilada, Y., 2012. Vision Based Page Segmentation: Extended and Improved Algorithm.

Anagnostopoulos, I., Anagnostopoulos, C., Vergados, D.D., 2010. Estimating evolution of freshness in Internet cache directories under the capture–recapture methodology. Computer Networks 54, 741–765.

Antonacopoulos, A., Karatzas, D., Ortiz Lopez, J., 2001. Accessing textual information embedded in internet images, in: SPIE, Internet Imaging II. Presented at the SPIE, San Jose, USA, pp. 198–205.

Apte, C., Damerau, F., Weiss, S., 1998. Text mining with decision rules and decision trees. IBM Watson Research Center.

Artail, H., Fawaz, K., 2008. A fast HTML web page change detection approach based on hashing and reducing the number of similarity computations. Data & Knowledge Engineering 66, 326–337.

Asirvatham, A.P., Ravi, K.K., 2001. Web page classification based on document structure, in: IEEE National Convention.

Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern information retrieval. ACM press New York.

Baluja, S., 2006. Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework, in: Proceedings of the 15th International Conference on World Wide Web. ACM, pp. 33–42.

Bar-Ilan, J., 2002. Methods for measuring search engine performance over time. Journal of the American Society for Information Science and Technology 53, 308–319.

Barrett, R., Maglio, P.P., Kellem, D.C., 1997. How to personalize the Web, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 75–82.

Bar-Yossef, Z., Rajagopalan, S., 2002. Template detection via data mining and its applications, in: Proceedings of the 11th International Conference on World Wide Web. pp. 580–591.

Bennett, P.N., Dumais, S.T., Horvitz, E., 2005. The combination of text classifiers using reliability indicators. Information Retrieval 8, 67–100.

Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X., 2012. Modeling the impact of short- and long-term behavior on search personalization, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’12. ACM, New York, NY, USA, pp. 185–194.

Borodin, Y., Mahmud, J., Ramakrishnan, I.V., Stent, A., 2007. The HearSay non-visual web browser, in: Proceedings of the 2007 International Cross-disciplinary Conference on Web Accessibility (W4A). ACM, pp. 128–129.

Bouramoul, A., Kholladi, M.-K., Doan, B.-L., 2011. PRESY: A Context Based Query Reformulation Tool for Information Retrieval on the Web. arXiv:1106.2289.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159.

Breiman, L., Friedman, J.H., Olshen, R.A., 1984. Classification and Regression Trees. Wadsworth, Belmont, California.

Brin, S., Page, L., 1998a. The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems 30, 107–117.

Brin, S., Page, L., 1998b. The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems 30, 107–117.

Burget, R., Rudolfová, I., 2009. Web page element classification based on visual features, in: Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference On. IEEE, pp. 67–72.

Bush, V., 1945. The Atlantic Monthly. As We May Think 176, 101–108.

Buyukkokten, O., Garcia-Molina, H., Paepcke, A., 2001. Accordion summarization for end-game browsing on PDAs and cellular phones, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 213–220.

Cai, D., He, X., Wen, J.R., Ma, W.Y., 2004. Block-level link analysis, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 440–447.

Cai, D., Yu, S., Wen, J.R., Ma, W.Y., 2003. VIPS: a vision based page segmentation algorithm. Microsoft Technical Report, MSR-TR-2003-79.

Cambazoglu, B.B., Karaca, E., Kucukyilmaz, T., Turk, A., Aykanat, C., 2007. Architecture of a grid-enabled Web search engine. Information processing & management 43, 609–623.

Cao, J., Mao, B., Luo, J., 2010. A segmentation method for web page analysis using shrinking and dividing. International Journal of Parallel, Emergent and Distributed Systems 25, 93–104.

Cardoso-Cachopo, A., Oliveira, A., 2003. An empirical comparison of text categorization methods, in: String Processing and Information Retrieval. Springer, pp. 183–196.

Castells, P., Fernandez, M., Vallet, D., 2007. An adaptation of the vector-space model for ontology-based information retrieval. Knowledge and Data Engineering, IEEE Transactions on 19, 261–272.

Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F., 2007. Know your neighbors: Web spam detection using the web topology, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 423–430.

Chakrabarti, D., Kumar, R., Punera, K., 2008. A graph-theoretic approach to webpage segmentation, in: Proceeding of the 17th International Conference on World Wide Web. ACM, pp. 377–386.

Chauhan, N., Sharma, A.K., 2007. Analyzing anchor-links to extract semantic inferences of a web page, in: Information Technology,(ICIT 2007). 10th International Conference On. IEEE, pp. 277–282.

Chawathe, S.S., Garcia-Molina, H., 1997. Meaningful change detection in structured data, in: ACM SIGMOD. pp. 26–37.

Chawathe, S.S., Rajaraman, A., Garcia-Molina, H., Widom, J., 1996. Change detection in hierarchically structured information, in: ACM SIGMOD. pp. 493–504.

Chen, L., Sycara, K., 1998. WebMate: a personal agent for browsing and searching, in: Proceedings of the Second International Conference on Autonomous Agents. ACM, pp. 132–139.

Chen, Y., Xie, X., Ma, W.Y., Zhang, H.J., 2005. Adapting web pages for small-screen devices. Internet Computing, IEEE 9, 50–56.

Chen, Y.F., Koutsofios, E., 1998. The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web. World Wide Web 1, 27–44.

Chien, W.S., 2000. Learning query behavior in the haystack system.

Chun, Y., Yazhou, L., Qiong, Q., 2012. An Approach for News Web-Pages Content Extraction Using Densitometric Features, in: Hu, W. (Ed.), Advances in Electric and Electronics, Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, pp. 135–139.

Claypool, M., Le, P., Wased, M., Brown, D., 2001. Implicit interest indicators, in: Proceedings of the 6th International Conference on Intelligent User Interfaces. ACM, pp. 33–40.

Cobena, G., Abiteboul, S., Marian, A., 2002. Detecting changes in XML documents, in: Data Engineering, 2002. Proceedings. 18th International Conference On. IEEE, pp. 41–52.

Cooper, W.S., 1988. Getting beyond boole. Information Processing & Management 24, 243–248.

Craswell, N., Hawking, D., Robertson, S., 2001. Effective site finding using link anchor information, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 250–257.

Craven, T.C., 2003. HTML tags as extraction cues for web page description construction. Informing Science 6, 1–12.

Dai, N., Davison, B.D., 2010. Freshness matters: in flowers, food, and web authority, in: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 114–121.

Daniel, S.H., 1997. Algorithms for the Longest common subsequences problem. Journal of Association for Computing Machinery 24, 664–675.

Dawson, A., 2004. Creating metadata that work for digital libraries and Google. Library Review 53, 347–350.

Dietterich, T.G., Bakiri, G., 1994. Error-correcting output codes: A general method for improving multiclass inductive learning programs, in: Santa Fe Institute Studies In The Sciences Of Complexity. Addison-Wesley Publishing Co, pp. 395–395.

Dietterich, T.G., Bakiri, G., 1995. Solving multiclass learning problems via error-correcting output codes. arXiv preprint cs/9501101.

Ding, L., Zhou, L., Finin, T., Joshi, A., 2005. How the semantic web is being used: An analysis of foaf documents, in: System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference On. IEEE, p. 113c–113c.

Douglis, F., Ball, T., 1996. Tracking and viewing changes on the web, in: Proc. of the 1996 USENIX Technical Conference.

Duan, K.B., Keerthi, S., 2005. Which is the best multiclass SVM method? An empirical study. Multiple Classifier Systems 732–760.

Eiron, N., McCurley, K.S., 2003. Analysis of anchor text for web search, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 459–460.

Fauzi, F., Hong, J.L., Belkhatir, M., 2009. Webpage segmentation for extracting images and their surrounding contextual information, in: Proceedings of the Seventeen ACM International Conference on Multimedia.

Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E., 2011. Semantically enhanced Information Retrieval: An ontology-based approach. Web Semantics: Science, Services and Agents on the World Wide Web 9, 434–452.

Flesca, S., Masciari, E., 2003. Efficient and effective web change detection. Data & Knowledge Engineering 46, 203–224.

Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T., 2005. Evaluating implicit measures to improve web search. ACM Transactions on Information Systems (TOIS) 23, 147–168.

Fredrikson, M., Livshits, B., 2011. RePriv: Re-envisioning in-browser privacy, in: IEEE Symposium on Security and Privacy.

Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A., 2007. User profiles for personalized information access. The adaptive web 54–89.

Goecks, J., Shavlik, J., 2000. Learning users' interests by unobtrusively observing their normal behavior, in: Proceedings of the 5th International Conference on Intelligent User Interfaces. ACM, pp. 129–132.

Goh, K.-S., Chang, E., Cheng, K.-T., 2001. SVM binary classifier ensembles for image classification, in: Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM '01. ACM, New York, NY, USA, pp. 395–402.

Golbeck, J., Parsia, B., Hendler, J., 2003. Trust networks on the semantic web. Cooperative Information Agents VII 238–249.

Gomory, R.E., Hu, T.C., 1961. Multi-terminal network flows. Journal of the Society for Industrial & Applied Mathematics 9, 551–570.

Grimnes, G., Edwards, P., Preece, A., 2004. Learning meta-descriptions of the foaf network. The Semantic Web–ISWC 2004 152–165.

Guha, S., Reznichenko, A., Tang, K., Haddadi, H., Francis, P., 2009. Serving ads from localhost for performance, privacy, and profit, in: Proceedings of the 8th Workshop on Hot Topics in Networks (HotNets' 09), New York, NY.

Gupta, P., 2012. Context Based Relevance Evaluation of Web Documents, in: Parashar, M., Kaushik, D., Rana, O.F., Samtaney, R., Yang, Y., Zomaya, A. (Eds.), Contemporary Computing, Communications in Computer and Information Science. Springer Berlin Heidelberg, pp. 201–212.

Gyongyi, Z., Garcia-Molina, H., 2005. Web spam taxonomy, in: First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11, 10–18.

Hattori, G., Hoashi, K., Matsumoto, K., Sugaya, F., 2007. Robust web page segmentation for mobile terminal using content-distances and page layout information, in: Proceedings of the 16th International Conference on World Wide Web. ACM, pp. 361–370.

Haveliwala, T.H., 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. Knowledge and Data Engineering, IEEE Transactions on 15, 784–796.

Hernández, I., Rivero, C.R., Ruiz, D., Arjona, J.L., 2012. An Experiment to Test URL Features for Web Page Classification, in: Rodríguez, J.M.C., Pérez, J.B., Golinska, P., Giroux, S., Corchuelo, R. (Eds.), Trends in Practical Applications of Agents and Multiagent Systems, Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, pp. 109–116.

Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. Neural Networks, IEEE Transactions on 13, 415–425.

Hu, J., 2008. Personalized web search by using learned user profiles in re-ranking.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of classification 2, 193–218.

Hunt, E.B., Marin, J., Stone, P.J., 1966. Experiments in induction. Academic Press, New York.

Iglesias, J.A., Angelov, P., Ledezma, A., Sanchis, A., 2012. Creating Evolving User Behavior Profiles Automatically. IEEE Transactions on Knowledge and Data Engineering 24, 854 –867.

Ito, T., Sano, H., Ozono, T., Shintani, T., 2008. A hierarchical web page segmentation algorithm using machine learning, in: Proceedings of the 11th IASTED International Conference. p. 044.

Jacob, J., Sanka, A., Pandrangi, N., Chakravarthy, S., 2004. Web-Vigil: an approach to just-in-time information propagation in large network-centric environments. Web dynamics. Springer, Berlin 301–318.

Järvelin, K., Kekäläinen, J., 2000. IR evaluation methods for retrieving highly relevant documents, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 41–48.

Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS) 20, 422–446.

Jones, K.S., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 28, 11–21.

Kamba, T., Sakagami, H., Koseki, Y., 1997. ANATAGONOMY: a personalized newspaper on the World Wide Web. International Journal of Human-Computer Studies 46, 789–803.

Kang, J., Yang, J., Choi, J., 2010. Repetition-based Web page segmentation by detecting tag patterns for small-screen devices. Consumer Electronics, IEEE Transactions on 56, 980–986.

Kelly, D., Teevan, J., 2003. Implicit feedback for inferring user preference: a bibliography, in: ACM SIGIR Forum. ACM, pp. 18–28.

Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) 46, 604–632.

Kohavi, R., Provost, F., 1998. Glossary of terms. Machine Learning 30, 271–274.

Kohlschütter, C., 2009. A densitometric analysis of web template content, in: Proceedings of the 18th International Conference on World Wide Web. ACM, pp. 1165–1166.

Kohlschütter, C., Nejdl, W., 2008. A densitometric approach to web page segmentation, in: Proceeding of the 17th ACM Conference on Information and Knowledge Management. ACM, pp. 1173–1182.

Kuppusamy K.S, Aghila G, 2011. Segmentation Based Approach to Dynamic Page Construction from Search Engine Results. IJCSE 3.

Kuppusamy K.S, Aghila G, 2012a. A personalized web page content filtering model based on segmentation. (IJIST) 2.

Kuppusamy K.S, Aghila G, 2012b. Semantic snippet construction for search engine results based on segment evaluation. International Journal of Information Technology and Knowledge Management 4, 581–583.

Kwon, O.W., Jung, S.H., Lee, J.H., Lee, G., 1999. Evaluation of category features and text structural information on a text categorization using memory based reasoning, in: Proceedings of the 18th International Conference on Computer Processing of Oriental Languages (ICCPOLÕ99). pp. 153–158.

La Fontaine, R., 2001. A delta format for XML: Identifying changes in XML files and representing the changes in XML, in: XML Europe.

Lam, W., Ho, C.Y., 1998. Using a generalized instance set for automatic text categorization, in: Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 21 St Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 81–89.

Larkey, L.S., Croft, W.B., 1996. Combining classifiers in text categorization, in: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 289–297.

Leung, K.W.T., Lee, D.L., 2010. Deriving concept-based user profiles from search engine logs. Knowledge and Data Engineering, IEEE Transactions on 22, 969–982.

Lewis, D., 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. Machine Learning: ECML-98 4–15.

Li, J.Q., Zhao, Y., Garcia-Molina, H., 2012. A path-based approach for web page retrieval. World Wide Web 15, 257–283.

Li, L., Shang, Y., Zhang, W., 2002. Improvement of HITS-based algorithms on web documents, in: Proceedings of the 11th International Conference on World Wide Web. ACM, pp. 527–535.

Lin, S.H., Ho, J.M., 2002. Discovering informative content blocks from Web documents, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 588–593.

Liu, F., Yu, C., Meng, W., 2002. Personalized web search by mapping user queries to categories, in: Proceedings of the Eleventh International Conference on Information and Knowledge Management. ACM, pp. 558–565.

Liu, L., Pu, C., Tang, W., 2000. WebCQ-detecting and delivering information changes on the web, in: Proceedings of the Ninth International Conference on Information and Knowledge Management. ACM, pp. 512–519.

Liu, X., Lin, H., Tian, Y., 2011. Segmenting Webpage with Gomory-Hu Tree Based Clustering. Journal of Software 6, 2421–2425.

Luhn, H.P., 1957. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development 1, 309–317.

Maan, A.K., James, A.P., 2012. Ranking importance based information on the world wide web, in: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, ICACCI '12. ACM, New York, NY, USA, pp. 889–897.

Mahmud, J., Borodin, Y., Das, D., Ramakrishnan, I.V., 2007a. Combating information overload in non-visual web access using context, in: Proceedings of the 12th International Conference on Intelligent User Interfaces. ACM, pp. 341–344.

Mahmud, J.U., Borodin, Y., Ramakrishnan, I.V., 2007b. Csurf: a context-driven non-visual web-browser, in: Proceedings of the 16th International Conference on World Wide Web. ACM, pp. 31–40.

Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval. Cambridge University Press Cambridge.

Matthijs, N., Radlinski, F., 2011. Personalizing web search using long term browsing history, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM, pp. 25–34.

Mayoraz, E., Alpaydin, E., 1999. Support vector machines for multi-class classification. Engineering Applications of Bio-Inspired Artificial Neural Networks 833–842.

McCallum, A., Nigam, K., 1998. A comparison of event models for naive bayes text classification, in: AAAI-98 Workshop on Learning for Text Categorization. Citeseer, pp. 41–48.

Mianowska, B., Nguyen, N.T., 2012. Tuning user profiles based on analyzing dynamic preference in document retrieval systems. Multimed Tools Appl 1–26.

Milic-Frayling, N., Sommerer, R., 2002. Smartview: Flexible viewing of web page contents, in: Intl. World Wide Web Conf.(WWW).

Mitchell, T.M., 1997. Machine Learning. McGraw-Hill.

Moukas, A., 1997. Amalthaea information discovery and filtering using a multiagent evolving ecosystem. Applied Artificial Intelligence 11, 437–457.

Mukherjee, S., Yang, G., Ramakrishnan, I., 2003. Automatic annotation of content-rich html documents: Structural and semantic analysis. The Semantic Web-ISWC 2003 533–549.

Nguyen, C.K., Likforman-Sulem, L., Moissinac, J.-C., Faure, C., Lardon, J., 2012. Web Document Analysis Based on Visual Segmentation and Page Rendering, in: 2012 10th IAPR International Workshop on Document Analysis Systems (DAS). Presented at the 2012 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 354 –358.

Noruzi, A., 2007. A study of HTML Title tag creation behavior of academic Web sites. The Journal of academic librarianship 33, 501–506.

Ogilvie, P., Callan, J., 2003. Combining structural information and the use of priors in mixed named-page and homepage finding, in: Text Retrieval Conference. p. 177.

Ong, W.K., Hong, J.L., Fauzi, F., Tan, E.X., 2012. Ontological based webpage classification, in: 2012 International Conference on Information Retrieval Knowledge Management (CAMP). Presented at the 2012 International Conference on Information Retrieval Knowledge Management (CAMP), pp. 224 –228.

Pant, G., 2003. Deriving link-context from HTML tag tree, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. ACM, pp. 49–55.

Pazzani, M.J., Muramatsu, J., Billsus, D., 1996. Syskill & Webert: Identifying interesting web sites, in: Proceedings of the National Conference on Artificial Intelligence. pp. 54–61.

Pnueli, A., Bergman, R., Schein, S., Barkol, O., 2009. Web Page Layout Via Visual Segmentation. HP Laboratories.

Ponte, J.M., Croft, W.B., 1998. A language modeling approach to information retrieval, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 275–281.

Porter, M.F., 1980. An algorithm for suffix stripping.

Pretschner, A., Gauch, S., 1999. Ontology based personalized search, in: Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference On. IEEE, pp. 391–398.

Qi, X., Davison, B.D., 2009. Web page classification: Features and algorithms. ACM Computing Surveys (CSUR) 41, 12.

Quinlan, J.R., 1993. C4. 5: programs for machine learning. Morgan kaufmann.

Quiroga, L.M., Mostafa, J., 1999. Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems, in: Proceedings of the Fourth ACM Conference on Digital Libraries. pp. 238–239.

Raggett, D., 1998. Clean up your Web pages with HP's HTML tidy. Computer networks and ISDN systems 30, 730–732.

Rivest, R., 1992. The MD5 message-digest algorithm.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., 1995. Okapi at TREC-3. NIST SPECIAL PUBLICATION SP 109–109.

Saad, M.B., Gançarski, S., 2010. Using visual pages analysis for optimizing web archiving, in: Proceedings of the 2010 EDBT/ICDT Workshops. ACM, p. 43.

Salton, G., 1968. Automatic Information Organization and Retrieval. New York: McGraw-Hill.

Salton, G., 1971. The SMART retrieval system—experiments in automatic document processing.

Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. Communications of the ACM 18, 613–620.

Sato, N., Uehara, M., Sakai, Y., 2003. Temporal ranking for fresh information retrieval, in: Proceedings of the Sixth International Workshop on Information Retrieval with Asian languages-Volume 11. pp. 116–123.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR) 34, 1–47.

Shavlik, J., Calcari, S., Eliassi-Rad, T., Solock, J., 1998. An instructable, adaptive interface for discovering and monitoring information on the World-Wide Web, in: Proceedings of the 4th International Conference on Intelligent User Interfaces. ACM, pp. 157–160.

Shavlik, J., Eliassi-Rad, T., 1998. Intelligent agents for web-based tasks: An advice-taking approach, in: AAAI/ICML Workshop on Learning for Text Categorization. pp. 63–70.

Singhal, A., Salton, G., Mitra, M., Buckley, C., 1996. Document length normalization. Information Processing & Management 32, 619–633.

Song, R., Liu, H., Wen, J.R., Ma, W.Y., 2004. Learning block importance models for web pages, in: Proceedings of the 13th International Conference on World Wide Web. ACM, pp. 203–211.

Sontag, D., Collins-Thompson, K., Bennett, P.N., White, R.W., Dumais, S., Billerbeck, B., 2012. Probabilistic models for personalizing web search, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM ’12. ACM, New York, NY, USA, pp. 433–442.

Spool, J.M., 1999. Web site usability: a designer’s guide. Morgan Kaufmann.

Strehl, A., Ghosh, J., 2003. Cluster ensembles- A knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research 3, 583–617.

Sun, A., Lim, E.P., Ng, W.K., 2002. Web classification using support vector machine, in: Proceedings of the 4th International Workshop on Web Information and Data Management. ACM, pp. 96–99.

Takagi, H., Asakawa, C., Fukuda, K., Maeda, J., 2002. Site-wide annotation: reconstructing existing pages to be accessible, in: Proceedings of the Fifth International ACM Conference on Assistive Technologies. ACM, pp. 81–88.

Teevan, J., Dumais, S.T., Horvitz, E., 2005. Personalizing search via automated analysis of interests and activities, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 449–456.

Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., Barocas, S., 2010. Adnostic: Privacy preserving targeted advertising, in: 17th Network and Distributed System Security Symposium.

Trajkova, J., Gauch, S., 2003. Improving ontology-based user profiles.

Tsoumakas, G., Katakis, I., 2007. Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) 3, 1–13.

Tsukada, M., Washio, T., Motoda, H., 2001. Automatic web-page classification by using machine learning methods. Web Intelligence: Research and Development 303–313.

Vallet, D., Cantador, I., Jose, J., 2010. Personalizing web search with folksonomy-based user and document profiles. Advances in Information Retrieval 420–431.

Vapnik, V.N., 1999. An overview of statistical learning theory. Neural Networks, IEEE Transactions on 10, 988–999.

Vineel, G., 2009. Web page DOM node characterization and its application to page segmentation, in: Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference On. IEEE, pp. 1–6.

Wahba, G., 1999. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. Advances in Kernel Methods-Support Vector Learning 6, 69–87.

Wahba, G., 2002. Soft and hard classification by reproducing kernel Hilbert space methods. Proceedings of the National Academy of Sciences 99, 16524–16530.

Wang, X.J., Ma, W.Y., Xue, G.R., Li, X., 2004. Multi-model similarity propagation and its application for web image retrieval, in: Proceedings of the 12th Annual ACM International Conference on Multimedia. ACM, pp. 944–951.

Wang, Y., DeWitt, D.J., Cai, J.Y., 2003. X-Diff: An effective change detection algorithm for XML documents, in: Data Engineering, 2003. Proceedings. 19th International Conference On. IEEE, pp. 519–530.

Wu, O., Chen, Y., Li, B., Hu, W., 2011. Evaluating the visual quality of web pages using a computational aesthetic approach, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. ACM, pp. 337–346.

Xia, T., Chai, Y., Wang, T., 2012. Improving SVM on web content classification by document formulation, in: 2012 7th International Conference on Computer Science Education

(ICCSE). Presented at the 2012 7th International Conference on Computer Science Education (ICCSE), pp. 110 –113.

Xiang, P., Yang, X., Shi, Y., 2007. Web page segmentation based on gestalt theory, in: Multimedia and Expo, 2007 IEEE International Conference On. IEEE, pp. 2253–2256.

Xie, X., Miao, G., Song, R., Wen, J.R., Ma, W.Y., 2005. Efficient browsing of web search results on mobile devices based on block importance model, in: Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference On. IEEE, pp. 17–26.

Xu, Y., Wang, K., Zhang, B., Chen, Z., 2007. Privacy-enhancing personalized web search, in: Proceedings of the 16th International Conference on World Wide Web. ACM, pp. 591–600.

Xue, Y., Hu, Y., Xin, G., Song, R., Shi, S., Cao, Y., Lin, C.Y., Li, H., 2007. Web page title extraction and its application. Information processing & management 43, 1332–1347.

Yang, X., Shi, Y., 2009. Enhanced gestalt theory guided Web page segmentation for mobile browsing, in: Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences On. IET, pp. 46–49.

Yang, Y., Slattery, S., Ghani, R., 2002. A study of approaches to hypertext categorization. Journal of Intelligent Information Systems 18, 219–241.

Yesilada, Y., 2011. Web Page Segmentation: A Review.

Yin, X., Lee, W.S., 2004. Using link analysis to improve layout on mobile devices, in: Proceedings of the 13th International Conference on World Wide Web. ACM New York, NY, pp. 338–344.

Yin, X., Lee, W.S., 2005. Understanding the function of web elements for mobile content delivery using random walk models, in: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web. ACM, pp. 1150–1151.

Zhang, C., Xue, G.R., Yu, Y., Zha, H., 2009. Web-scale classification with naive bayes, in: Proceedings of the 18th International Conference on World Wide Web. ACM, pp. 1083–1084.

Zhou, D., Lawless, S., Wade, V., 2012. Web Search Personalization Using Social Data, in: Zaphiris, P., Buchanan, G., Rasmussen, E., Loizides, F. (Eds.), Theory and Practice of Digital Libraries, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 298–310.

Zhu, X., Gauch, S., 2000. Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web, in: Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 23 Rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Citeseer, pp. 288–295.

LIST OF PUBLICATIONS

International Conference Publications

1. Kuppusamy, K.S., & Aghila, G. (2009), "FEAST : A Multistep, Feedback Centric, Freshness Oriented Search Engine Page(s)": 997-1001, IEEE International Advance Computing Conference, Patiala, 2009.ISBN : 978-981-08-2465-5

2. Kuppusamy, K.S., Aghila, G., (2013). An Ontology Based Model for User Profile Building Using Web Page Segment Evaluation, in: Meghanathan, N., Nagamalai, D., Chaki, N. (Eds.), Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, pp. 421–430.

Peer Reviewed & Indexed International Journals

3. Kuppusamy, K.S., & Aghila, G. (2011) “Museum: Multidimensional Web page Segment Evaluation Model” Journal of Computing, Vol 3, Issue 3.pp.24-27, 2011, ISSN 2151-9617.
4. Kuppusamy, K.S., & Aghila, G. (2011) “Segmentation Based Approach to Dynamic Page Construction from Search Engine Results”, International Journal on Computer Science and Engineering (IJCSE), Vol. 3, No. 3, pp. 1097-1105, 2011, ISSN : 0975-3397.
5. Kuppusamy, K.S., & Aghila, G. (2011) “A Model for Web Page Usage Mining Based on Segmentation”, International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1144-1148, ISSN: 0975-9646.
6. Kuppusamy, K.S., & Aghila, G. (2011) “Semantic Snippet Construction For Search Engine Results Based On Segment Evaluation”, international journal of information technology and knowledge management , volume 4, no. 2, 2011, pp. 581-583, ISSN: 0973-4414.
7. Kuppusamy, K.S., & Aghila, G. (2011) “Live-Marker: A Personalized Web Page Content Marking Tool”, International Journal of Information Technology and Knowledge Management, Volume 4, No. 2, pp. 485-488, 2011, ISSN: 0973-4414.
8. Kuppusamy, K.S., & Aghila, G. (2011) “We.I.Pe: Web Identification of People using e-mail ID”, International Journal on Computer Science and Engineering (IJCSE), Vol. 3, No. 6, 2011, pp. 2310-2316, ISSN : 0975-3397.

9. Kuppusamy, K.S., & Aghila, G. (2012) "A Personalized Web Page Content Filtering Model Based On Segmentation", International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.1, ISSN: 2249-1139.
10. Kuppusamy, K.S., & Aghila, G. (2012) "MORPES: A Model For Personalized Rendering Of Web Content On Mobile Devices", International Journal in Foundations of Computer Science & Technology (IJFCST),42-51, Vol. 2, No.2, March 2012, ISSN : 1839-7662
11. Kuppusamy, K.S., & Aghila, G. (2012) "A Model for Personalized Keyword Extraction from Web Pages using Segmentation" International Journal of Computer Applications 42(4):21-26, ISSN: 0975 - 8887.
12. Kuppusamy, K.S., & Aghila, G. (2012) "Segmentation Based Personalized Web Page Summarization Model" , Journal of Advances in Information Technology JAIT, ISSN 1798-2340, 2012, Vol 3, No 3, 155-161.
13. Kuppusamy, K.S., & Aghila, G. (2012) "Multidimensional Web Page Evaluation Model Using Segmentation And Annotations", International Journal on Cybernetics & Informatics (IJCI) Vol.1, No.4, ISSN: 2277-548x.
14. Kuppusamy, K.S., & Aghila, G. (2012), "Multimodal Approach to Incremental Profile Building", International Journal of Web and Semantic Technology (IJWEST), Vol: 3, No: 4. ISSN: 0975-9026.

Research Papers Communicated:

15. Kuppusamy, K.S., & Aghila, G. "Semantic Computation of Page Score", World Scientific – International Journal of Information Technology and Decision Making"

16. Kuppusamy, K.S., & Aghila, G. "CasePer: An Efficient Model for Personalized Web Page Change Detection Based on Segmentation", Elsevier JKSU Computer and Information Sciences.

VITAE

Mr. K.S. Kuppusamy, the author of this thesis is an Assistant Professor in the Department of Computer Science, School of Engineering and Technology, Pondicherry University. He was born on 22nd Jul 1980 at Madurai, India.

He has received his Bachelor of Science in Computer Science in the year 2000 and Master of Computer Science and Information Technology in the year 2005, from Madurai Kamaraj University. His area of interest includes Web Search Engineering and Intelligent Information Management.

He is UGC-NET (University Grants Commission – National Educational Testing) qualified with JRF (Junior Research Fellowship) distinction in the year 2005. He is a university rank holder in both Under- Graduation and Post-Graduation.

He has got a total of 8 years of Industry and PG-Teaching experience. He has published 16 peer-reviewed international journal papers and presented papers in IEEE and Springer sponsored conferences. He is recipient of the “Best Teacher” award, two times in a row during 2010 and 2011.