Chapter 4
Materials and Methods

4.1 Introduction

The challenges discussed in the previous chapter are resolved in the present work. This chapter describes the frameworks, algorithms and implementation used to solve the problems discussed in chapter 3. The materials and methods used to overcome these challenges are elaborated in detail in this chapter. It also discusses the tools used for data gathering and data analysis.

To overcome the problem of context independence of the search engines and to reduce the cognitive load on the user, Noise Removal Framework has been proposed in section 4.2. The implementation of Noise Removal Framework is referred as Noise Removal for Semantic Information Processing (NRSIP) framework. Its design and experimental setup are presented in section 4.3.

To assist the user in the search process, section 4.4 illustrates the design of the Cognitive Information Mapping Model (CIMM). Section 4.5 proposes and discusses in detail the structure, working and implementation of the Secured Cognitive Agent based Multi-strategic Intelligent Search System (CAbMsISS). Finally, decisive conclusion is presented in section 4.6.

4.2 Noise Removal Framework (116)

Three different search scenarios S1 (section 3.3.1), S2 (section 3.3.2) and S3 (section 3.3.3) discussed in the previous chapter (section 3.3) elucidate that the irrelevant results were retrieved in these scenarios due to inappropriate mapping of Concept and Context of the keywords (116). This introduces noise in the system, thereby resulting in irrelevant retrieval.

Since, 99% of the results retrieved for scenario S3 (section 3.3.3) were found to be irrelevant (116), hence, it has been suggested in (116) to perform semantic search to find the experts in a specific field and working in a particular Domain.
To achieve semantically improved search results for scenario S3, the researchers in (116) recommended to improve the Concept-Context mapping of the keywords for this scenario. With the aim to search for people with expertise in a specific field and working in a particular Domain, Noise Removal Framework suggested in (116) is elucidated in this section for the Search Problem (SP).

SP: find the human expert(s) (person) with expertise in a specific field and working in Panjab University i.e. Domain = Panjab University.

Justification for selecting SP (Expert Search Task) i.e. finding the human expert(s) (person) with expertise in a specific field.

With vast pool of information and documents available on WWW, it is undoubtedly a leading source of data. It has opened numerous avenues not only for individualistic work, but also for collaborations amongst people with common interest or with expertise in a specific area. The concern for the Expert Search Task is to identify people with expertise in a specific field. The task can be accomplished by browsing and inspecting through the expert’s organization Website (195). This process is called look-up search or fact-finding or known-item search. It is performed when the organization is known.

Many a times, the Website of academic organization or Higher Educational Institutions (HEI) provide a text box and search button for internal search within the Website. The field name (name of specific area for which expert is sought) can be written in the text box to extract the details of the experts in the specified field. This might return relevant results related to the details of the experts and may retrieve irrelevant results related to syllabi for the field, course content for the field, time table etc.

Hence, a search text box in the Website does not completely retrieve relevant results. Another alternative for Expert Search Task is to provide a complete faculty directory in the HEI Website to select the area for which the expert is sought or enter the field name. This feature is presently available in the Website of several Higher Educational Institutions including IIM’s (India) (196) as shown in Figure 4.1(a), Princeton University (197) as shown in Figure 4.1(b).
Figure 4.1(a): Faculty search option in IIM Ahmedabad Website

As shown in Figure 4.1(a), the experts in a particular field can be retrieved, but it cannot retrieve the experts in the specified field who are not Assistant Professors. Moreover, in this present structure, it is also not possible to search for Professors who are Dean or Director. In Figure 4.1(b), the search dialog box is missing and experts need to searched in a linear order.

Figure 4.1(b): Faculty search option in Princeton University Website
Expert Search Task consumes more effort when the HEI from where the expert is to be sought is not known. This involves series of activities including querying and browsing to be performed to obtain relevant information. It is referred as exploratory search and encompasses numerous sub-tasks that require multiple queries and search sessions to investigate before finally the user finds some meaningful and relevant information that extends the existing knowledge about the specific task. It is more time consuming and complex activity as compared to look-up search and includes exploration, uncertainty, creativity, knowledge discovery and learning (198). User’s primarily employ SE’s to perform exploratory search and seek information from WWW. SE’s manage 93% of the internet traffic (199). SE allows the user to search for any type of content from across the globe by entering the information seeking requirements in the form of query keywords.

Justification for selecting Domain = Panjab University for pretesting Noise Removal Framework

Noise Removal Framework has been developed to retrieve information about the experts with expertise in a specific field in HEI around the World. To measure the retrieval effectiveness of this framework; Precision (P), Recall (R) and No. of Relevant Document Missed (RDM) needs to be computed. For computing R and RDM, it is prerequisite to be familiar with the Website of all Higher Educational Institutions around the globe and finally acquire information about all the experts in the specific field to be searched. Taking this constraint into account, researchers considered the ranking given by The Times Higher Education Asia University Rankings 2014 (200).

Figure 4.1(c): The Times Higher Education Asia University Rankings 2014
According to (200), Panjab University scored the highest ranking among HEI in India (Figure 4.1(c)) and had also topped the country in “The World University Rankings for 2013-14”.

Moreover, Google Trends had also been used to compare the Worldwide trend of the search term “Panjab University” with other popular search terms like “Indian Institute of Technology, Kanpur” etc. as shown in Figure 4.1(d). These trends depict the search frequency of a particular search term in comparison to other search terms. Data collected for the time period May 2014 to May 2015 using (201) shown in Figure 4.1(d) illustrates that “Panjab University” was searched more as compared to other HEI in India.

Further, the Websites of HEI listed in Figure 4.1(d) were analyzed using AChecker (202) for compliance with Web Content Accessibility Guidelines (WCAG). It was found that the Website of Panjab University has comparatively less number of potential problems as compared to the Website of other HEI. Figure 4.1(e) and 4.1(f) presents the screenshot of the values retrieved from AChecker.

Hence, among the HEI in India, Panjab University was selected for the pilot study and pretesting of Noise Removal Framework was done using Domain = Panjab University.
Finally, Noise Removal Framework has been proposed by (116) to perform semantic search on SP to find the human expert(s) (person) with expertise in a specific field and working in Panjab University i.e. Domain = Panjab University.
According to (202), stemming and lemmatization algorithms have been implemented in majority of SE’s including Google and Yahoo! search engine to detect morphological disparity among the keywords and to include synonyms for the keywords. Most of the SE’s including Google do not disable the search for synonyms even when search keywords have double quotes. SE’s inability to disambiguate polysemous queries poses serious concerns in linguistic analysis due to their failure in understanding the sense of the query. Therefore, to improve the retrieval effectiveness of the query, it was suggested to employ text classification based filters (203). Keeping this in view, the authors in (116) have employed filters in the Noise Removal Framework to remove noise from the query in the task to search for expert(s) in a particular field.

Based on semantic search, the proposed framework considers the user context and semantic understanding of the information on the Semantic Web. It considers the Concept-Context attributes of the semantic search systems for the structural design of the framework. The framework frees the user from repetitively selecting and entering the appropriate keywords for efficient query formation. Rather, the user is presented with a set of pre-defined options in a customized manner generated using the Concept-Context mapping of the keywords for search scenario SP.

### 4.2.1 Working of Noise Removal Framework (116)

The diagrammatic representation of Noise Removal Framework proposed by researchers in (116) is shown in Figure 4.2(a).

![Figure 4.2(a): Noise Removal Framework](image-url)
The structural composition of Noise Removal Framework consists of the following three components:

- **User**: The user provides the input to the system and constitutes the input sub-system.
- **Mapping section**: It refers to the processing sub-system. The search options or choices are presented to the user by semantically formed mapping section.
- **Search Engine**: The results corresponding to the search options selected by the user are obtained from this output sub-system.

As shown in Figure 4.2(a), working of the framework depends on the user input that comprises of Field (Field name/Subject name) for which the expert is to be searched and the choices selected by the user from the available options. The task of the framework is to convert the query (Q) entered by the user to Semantic Query (SQ) that retrieves more relevant results. Initially SQ is empty.

When the user enters the Field and selects the choices, the Mapping section performs the Concept and Context Mapping. This results in forming SQ progressively as the choices are selected. Finally, SQ formed at the end of choice selection process is send to the third sub-system called Search Engine (SE) for efficient result retrieval. The following two sub-sections discuss in detail the mapping section consisting of both the Concept and Context Mapping. Diagrammatic representations and algorithm are presented in the subsequent sections to clarify the logic used in the processing module called Mapping.

### 4.2.1.1 Concept Mapping

Concept of SP is to find the human expert(s) i.e. Concept(SP) = Person (Human). To restrict the search engine to retrieve only the results referring to Person (Human), Concept Mapping Keyword (CMK) like Biodata, CV, Resume, Expertise, Research Area, Specialization, Experience, Profile etc. is added in SQ.

### 4.2.1.2 Context Mapping

SP consists of four Context (stated in Figure 4.2(b)) that needs to be mapped for enhancing retrieval effectiveness.
Text classification has been employed to classify different types of Context:

- Teaching Staff / Teaching Employee / Academic Staff, Non-Teaching Staff, Research Scholar, Student, Previous Employee, Alumni
- Academic Staff Categories (Assistant Professor, Associate Professor, Professor)
- Different Positions (Chairperson, Coordinator, Dean, Director)
- Different Academic Units (Centre, Chair, Department, Institute, School)

The diagrammatic illustration of Concept and Context Mapping for Noise Removal Framework is presented in Figure 4.3.
Description of the four Context for SP is as follows:

- **Field:** It refers to the Field name or Subject name for which the human expert must have expertise. Mapping section allows the user to enter the text referring to the Field name. The text entered by the user is added in SQ. In case, no Field name is entered, then SQ remains same.

- **Domain:** SP entails finding the human expert(s) only from a specific Domain i.e. Panjab University. Mapping section uses the operator (site:) and URL of Panjab University Website to fix the Domain. The fixed Domain is added in SQ. This refines the query and helps to achieve better results.

- **Works(Person:Type):** The Context:Works searches for the results only referring to the Teaching Staff working within the specified Domain. This helps to restrain the results corresponding to Student, Research Scholar, Non-Teaching Staff or Alumni. The following mapping is used for the Context:Works.

```plaintext
Teaching Staff (presently working)

  =
  !(Student)
  &&
  !(Research Scholar)
  &&
  !(Non-Teaching Staff)
  &&
  !(Previous Employee)
```

This mapping does not allow the search engine to retrieve the result corresponding to any Student of the University. It includes restricting the results pertaining to the Student on-roll as well as those who have passed out.

```plaintext
!(Student) = !(Present Student) || !(Alumni)
```

Similarly, the mapping also restricts from obtaining the result related to any Previous Employee of the University. In context of the framework, Previous
Employee refers to a Retired Employee or who had previously worked in the University, but is presently not on-roll of the University.

!(Previous Employee) = !(Retired Employee) || !(worked as)

If Teaching Staff = True,

then: perform mapping for Specializes(Person : Position).

➢ **Specializes(Person:Position):** All the Teaching Staff of the University is designated with some specific designation. Matlab (204) has been used to illustrate the mapping shown in Figure 4.4. The mapping allows to search for the expert(s) among the Teaching Staff with Position = Assistant Professor or Associate Professor or Professor. The corresponding position is added to SQ.

![Figure 4.4: Context Mapping: Specializes(Person:Position)](image)

**Algorithm:** Noise Removal presented in next section discusses the steps used to convert user query (Q) to semantically improved query (SQ).

**Step 1 - Concept Mapping**

**Step 2 - Context Mapping**

**Step 3 - Perform the search for SQ**
4.2.2 Algorithm: Noise Removal (Field name, User Choices)

Here, Field name is the subject for which expert is to be searched and User Choices refer to the options selected by the user.

1. **Perform Concept Mapping for Person(Human).**
   [use Concept Mapping Keyword (CMK)]

2. **[Concept Mapped ?]**
   If Concept Mapped = Person(Human), then:
   
   a. Add Concept Mapped in SQ.
   
   b. Accept Field name from user and add in SQ.
   
   c. Fix the search Domain = Panjab University and add in SQ.
   
   d. Perform Context Mapping for Works and Specializes.
   
   e. **[Context Mapped ?]**
      
      If Context Mapped for Works(Person) && Position(Person), then:
      
      Add matched Context in SQ.
      [All Noise is removed.]
      
      Else:
      
      Some Noise is removed.
      [End of If structure]
      
      Else: [If Concept is not mapped]
      
      No Noise is removed.
      [End of If structure]

3. **Perform the search using SQ if all Noise is removed.**

   /* End of Algorithm Noise Removal */

The logical flow diagram for Noise Removal Framework is presented in Figure 4.5. The framework provides more meaningful mapped query SQ to the user and thus removes noise from the query.
4.2.3 Experimental Setup

Out of four context specified for Noise Removal Framework (116) in Figure 4.2(b), pretesting of the Framework (116) has been conducted for only two context: field and Domain = Panjab University. **Context Mapping for Works and Specializes has not been considered in the pilot study.** Based on judgmental sampling\(^7\), three Field (Subject) names: “pattern recognition”, “phytoremediation” and “software engineering” were selected to form the queries for the experimental study (116).

For each of these three fields, queries were formulated using six different **Concept Mapping Keywords** (*person, faculty, expert, specialised, expertise and research*) without fixing Domain = Panjab University (query: keyword “subject name” “panjab university”) and with fixing Domain = Panjab University (query: keyword “subject name” site: puchd.ac.in). These queries have been executed using **Google SE**. For the experiment, 36 queries were formed. 6 queries for each Field; thus, for 3 Fields, total 18 queries were formulated. Initially, these 18 queries were executed without fixing Domain = Panjab University i.e. using organization’s name (“Panjab University”) in the query. Later, these queries were executed with fixing Domain = Panjab University i.e. by adding part of the Panjab University Website URL using operator site: puchd.ac.in. These 36 queries fall in the category

\(^7\)It is also known as purposive sampling and is primarily used when the interest of research is on a specific field or small group. In this sampling, selection of the sample is based on researcher’s choice.
of navigational queries. This has been done purposely to enable the authors to initially check accuracy of results corresponding to each context separately.

The proposed framework has been executed on a Windows 7, 64 bit system environment with an Intel Core i5–2430M CPU@ 2.40 GHz with 4 GB RAM and 500 GB hard disk. The snapshots and retrieved query results are presented in chapter 5. The top 10 retrieved results (DCV = 10) were used for the study. The data for the experiment was gathered after analyzing the documents retrieved for the queries. The gathered data is presented in chapter 5 in tabular form using MS-Excel software.

For each of the three Fields and six Context Mapping Keywords, queries were executed and data was collected for the following attributes:

- Total Documents Retrieved
- No. of Retrieved Documents Judged Relevant (RDJR)
- No. of Retrieved Documents Judged Irrelevant (RDJI)
- No. of Relevant Documents Missed (RDM)

The percentage of RDJR has also been calculated. Descriptive Statistics: Mean has been used to calculate the Mean RDJR percentile. For this, statistical function AVERAGE( ) available in MS-Excel software was used. Finally, the Mean RDJR values calculated for all the three fields were used to test the Null Hypothesis $H_0$ and analyze the results.

**Null Hypothesis $H_0$:** There is no significant difference in the Mean Retrieved Documents Judged Relevant (RDJR) without fixing the Domain = Panjab University and with fixing the Domain = Panjab University.

**Inferential Statistics: Independent $t$-test** was used to study whether there is any significant difference in the Mean RDJR without and with fixing the Domain = Panjab University.

After the success of pilot study, Noise Removal Framework was enhanced and implemented for all the four context specified in Figure 4.2(b) as shown in section 4.3 (Figure 4.7). The completely implemented Noise Removal Framework is referred as NRSIP (117).
4.3 Noise Removal for Semantic Information Processing (NRSIP) Framework (117)

The existing keyword-based search engines are unable to semantically process the information. This has been explained in (117) using the search scenario S4 (3.4.1) in which the user had searched for expert(s) in the field of HR in Panjab University. Different queries used for the search could not retrieve relevant results. The semantic ambiguity in the terms used in the query and Web documents has been attributed as one of the main reason for low precision and low recall. This ambiguity generates noise in the system. Some of the noise was removed in search scenario S4 by using variations of the keywords. The present structure of the Web desires user effort for word variation. This imposes high cognitive and learning load on the user as it requires proper selection of keywords for the query formation.

The inability of the search engines to map the user’s intent for Concept (thought) and Context of the query keywords with the result set and the incapability to handle word variation have been identified as the major problems accountable for introducing noise in the query. To resolve these issues, the researchers in (117) suggested to implement Noise Removal Framework (116) discussed in the previous section (4.2).

Noise Removal Framework proposed in (116) to perform semantic search on SP has been implemented and experimentally evaluated in this section. The implemented work is referred as Noise Removal for Semantic Information Processing (NRSIP) framework (117). It semantically improves the query by filtering (removing) noise from the query and thereby overcomes the inability of the keyword-based search engines in semantically processing the information for SP. This section focuses on the design, implementation and experimental setup of NRSIP framework. Moreover, it is comprehended in detail through the flowcharts developed using Matlab (204) and by the screen snapshots. The tools used for data gathering and analysis are also presented in this section.

The working and modular structure of NRSIP framework (117) is described in detail in section 4.3.1 and 4.3.2 using visuals. The experimental setup used for the
development of NRSIP framework is discussed in section 4.3.3. **Web Performance Load Tester (205) has been used for obtaining the results of the experiment.** It is discussed in section 4.3.3.1.1. The gathered data is presented in tabular form and analyzed using MS-Excel software in chapter 5.

### 4.3.1 Working of NRSIP Framework for Specific Domain (117)

NRSIP framework has been explicitly designed and implemented to address the issues regarding semantic and relevant retrieval of results for Search Problem (SP) stated in section 4.2. Figure 4.6 presents the high level view of NRSIP framework. It accepts the user's conceptual search problem (SP) and converts it to semantically improved statement called Semantic Query (SQ). The conversion of SP to SQ removes noise and thereby helps to improve retrieval efficiency of the search engine. **SQ formed after noise removal has been executed using Present Search Engine (PSE) Yahoo! search engine to retrieve the search results.**

![Diagram of NRSIP Framework](image)

**Figure 4.6: High Level View of Proposed Framework: NRSIP**

Categorical search provides the platform for the design of NRSIP framework. Choices are provided by NRSIP framework to allow the user to specify SP. It then converts the choices selected by the user to a semantically meaningful query called SQ by removing noise from the query. This frees the user from the chaos and burden of accurately selecting and specifying the keywords to form the query for SP.
As shown in Figure 4.7, NRSIP framework allows the user to find the expert(s) (Teaching Staff) in the Domain = Panjab University:

- in a particular field (subject area);
- with specific designation;
- holding additional position;
- associated with some specific division (Centre, Chair, Department, Institute, School);
- with or without Doctorate Degree.

![Figure 4.7: Categorical Search used in NRSIP framework for SP (find expert(s) in Panjab University)](image)

4.3.2 Structure of NRSIP Framework (117)

The modular composition of NRSIP framework consists of three modules as shown in Figure 4.8. These modules include:

- Query Formulation (QF)
- Query Execution (QE)
- Result Retrieval (RR)
4.3.2.1 Query Formulation (QF)

The module QF facilitates the user to specify the conceptual search query for
**SP: find expert(s) in Panjab University**. The module QF consists of a text box that
allows the user to enter the text corresponding to the field (subject). It also provides
several drop down list box with multiple options to enable the user to perform the
selection process. As the user enters the text and/or selects the options, the
framework converts SP to SQ by mapping the Concept and Context of SP. The
mapping removes the noise and ensures that the Semantic Information Processing
(SIP) of SQ results in relevant retrieval.

**Variables used**

- **Temporary variable m**: The data type of variable m is string. The process begins
  by initializing the temporary variable m with an empty string i.e. m="".

- **Variable SQ**: It is a string type variable used to store the Semantic Query (SQ)
  formed by the framework.
The following two steps are performed by the module QF to generate SQ:

**Step 1: Concept Mapping**

As discussed in (116), CMK can be used to map the Concept of SP. In this case, “show-biodata” is the CMK used for Concept Mapping. This is done by initializing SQ with the string “show-biodata”.

\[ SQ = \text{“show-biodata”} \]  \hspace{1cm} (1)

As shown in Equation (1), the keyword “show-biodata” maps the Concept for SP and ensures to retrieve results only referring to human experts.

**Step 2: Context Mapping**

The following five sub-modules perform the Context Mapping:

1. **in field:** It allows the user to enter the field/subject name for which the expert(s) is/are to be searched. The entered field is stored in a Text String (TS). NRSIP framework appends the entered string TS in SQ in Equation (1) as follows:

   \[ SQ = SQ \text{ TS} \]  \hspace{1cm} (1.1)

   \[ SQ = \text{“show-biodata” TS} \]

   Here, TS may refer to any field/subject like “Accounting”, “Data Structure”, “Marketing”, “Mathematics” etc.

   **Example:** In case the user enters “Marketing” as the field/subject, then:

   \[ SQ = \text{“show-biodata” “Marketing”} \]  \hspace{1cm} (E1)

   If the user does not enter any field i.e. TS is not entered, then SQ is not appended.

2. **who works as:** SQ in Equation (1.1) not only retrieves the details of Teaching Staff (employees) who have expertise in specified field TS, but it also retrieves details of Non-Teaching employees and other irrelevant results. The irrelevant results are retrieved due to the lack of Context Mapping. Hence, to enable the user to retrieve only the details of expert(s) who are presently working as Teaching Employees in the University, NRSIP framework provides itemization of different possible designation options as presented in Figure 4.9(a).
This facilitates the user to explicitly specify the designation of the expert(s). If the appropriate designation of the Teaching Employee is not known, then the user can choose a semantically equivalent option called “Faculty”. This retrieves the results corresponding to all the Teaching Employees irrespective of their designation.

Based on the option chosen by the user from among those listed in Figure 4.9(a), Mapping of this Context is performed by executing append action on SQ in Equation (1.1) or Equation (1) as illustrated in Figure 4.9(b).
a) **Faculty**: As shown in the logical flow diagram in Figure 4.9(b), this is the default option for sub-module: who works as. On selecting this option, the framework appends the string “* Professor” in SQ.

\[
\begin{align*}
\text{SQ} &= \text{“show-biodata” TS “*Professor”} \quad (2.1) \\
\text{SQ} &= \text{“show-biodata” “* Professor”} \quad (2.2)
\end{align*}
\]

**Outcome of (2.1) and (2.2)**

Equation (2.1) retrieves the details of all the expert(s) (Teaching Employees) in the specified field TS.

Equation (2.2) retrieves the details of all the expert(s) (Teaching Employees) irrespective of the field TS because the user has not specified any field in this case.

b) **Assistant Professor**: Append the string “Assistant Professor” in SQ.

\[
\begin{align*}
\text{SQ} &= \text{“show-biodata” TS “Assistant Professor”} \quad (2.3) \\
\text{SQ} &= \text{“show-biodata” “Assistant Professor”} \quad (2.4)
\end{align*}
\]

**Outcome of (2.3) and (2.4)**

Equation (2.3) retrieves the details of expert(s) (Teaching Employees - only Assistant Professor) in the specified field TS.

Equation (2.4) retrieves the details of expert(s) (Teaching Employees - only Assistant Professor) irrespective of the field TS.

---

**Example**: On selecting the option “Assistant Professor”, Equation (E1) becomes:

\[
\begin{align*}
\text{SQ} &= \text{“show-biodata” “Marketing” “Assistant Professor”} \quad (E2)
\end{align*}
\]

Equation (E2) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing”.

c) **Associate Professor**: Append the string “Associate Professor” in SQ and update the temporary variable m with the string “–Assistant”.

\[
\begin{align*}
\text{SQ} &= \text{“show-biodata” TS “Associate Professor”} \quad (2.5) \\
\text{SQ} &= \text{“show-biodata” “Associate Professor”} \quad (2.6)
\end{align*}
\]
Outcome of (2.5) and (2.6)

Equation (2.5) retrieves the details of expert(s) (Teaching Employees – only Associate Professor) in the specified field TS. Equation (2.6) retrieves the details of expert(s) (Teaching Employees – only Associate Professor) irrespective of the field TS.

d) Professor: Append the string “Professor” in SQ and update temporary variable m with the string “–Assistant –Associate”.

\[ \text{SQ} = \text{“show-biodata” TS “Professor”} \quad (2.7) \]
\[ \text{SQ} = \text{“show-biodata” “Professor”} \quad (2.8) \]

Outcome of (2.7) and (2.8)

Equation (2.7) retrieves the details of expert(s) (Teaching Employees – only Professor) in the specified field TS. Equation (2.8) retrieves the details of expert(s) (Teaching Employees – only Professor) irrespective of the field TS.

3. holds additional position of: Teaching employees in the university may also be given some additional charge. Hence, they may hold certain additional position.

Example: An Assistant Professor of some Department in the University may also be working as Chairperson of the Department.

For this scenario, NRSIP framework provides a set of designation options as shown in Figure 4.10(a) to accomplish more specific results.

Figure 4.10(a): User Input Module for sub-module: holds additional position of
(Additional position of the expert(s) in the University)
a) **None**: Being set as the default option, no append action is performed in SQ on selecting None.

\[ SQ = SQ \]  

(3.1)

On selection of any option other than None, the framework appends the string “Positions held” in SQ. This augments Context Mapping.

\[ SQ = SQ \text{“Positions held”} \]  

(3.2)

b) **Chairperson / Coordinator / Dean / Director**: The flow chart illustrated in Figure 4.10(b) brings to light the different actions performed by the framework in response to a particular option selected.

Append the string corresponding to the selected option in SQ as follows:

\[ SQ = SQ \text{“Chairperson”} \]  

(3.3)

\[ SQ = SQ \text{“Coordinator”} \]  

\[ SQ = SQ \text{“Dean”} \]  

\[ SQ = SQ \text{“Director”} \]  

**Outcome of (3.3)**

Equation (3.3) retrieves the details of expert(s) who also hold some additional position like Chairperson or Coordinator or Dean or Director.

**Example**: Selecting the option “Chairperson”, SQ in Equation “(E2)” becomes:

\[ SQ = \text{“show-biodata” “Marketing” “Assistant Professor” “Positions held” “Chairperson”} \]  

(E3)

Equation (E3) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing” and hold the position of “Chairperson”.

To retrieve the list of all the expert(s) designated as “Chairperson”, the following SQ can be used.

\[ SQ = \text{“show-biodata” “* Professor” “Positions held” “Chairperson”} \]
Figure 4.10(b): Flow Chart: holds additional position of
(Additional position of the expert(s) in the University)

4. associated with: It supports the user to access the details of expert(s) associated with some specific division of the University.

Figure 4.11: User Input Module for sub-module: associated with

No append action is performed in SQ on selecting the default option: Any. For any other selected option, the framework appends the string corresponding to the selected option in SQ as follows:


\[ SQ = SQ \text{“Centre”} \]
\[ SQ = SQ \text{“Chair”} \]
\[ SQ = SQ \text{“Department”} \] (4)
\[ SQ = SQ \text{“Institute”} \]
\[ SQ = SQ \text{“School”} \]

Outcome of (4)

Equation (4) retrieves the details of expert(s) associated with a specific division of the University.

**Example:** On selecting the option “Department”, the corresponding SQ in Equation (E3) becomes:

\[ SQ = \text{“show-biodata” “Marketing” “Assistant Professor” “Positions held” “Chairperson” “Department”} \] (E4)

Equation (E4) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing” and hold the position of “Chairperson” of a “Department”.

5. **who is:** This sub-module retrieves the details of expert(s) who hold or do not hold the Doctorate Degree. For the **default option:** Any, no append action is performed. Append action is performed for the other two options (Figure 4.12) as follows:

\[ \text{Doctorate: } SQ = SQ \text{“Ph.D.”} \]
\[ \text{Not Doctorate: } SQ = SQ \text{“~Ph.D.”} \] (5)

![Figure 4.12: User Input Module for sub-module: who is](image-url)
The above-mentioned Context Mapping steps (1 to 5) can be executed in any sequence. Moreover, it is not mandatory to perform all the steps.

The user terminates the choice selection process by clicking on the Button (YAHOO) of PSE as shown in Figure 4.13(a).

As a corollary, it accomplishes these two actions:

1. **Complete SQ**: NRSIP framework performs the following two actions:
   - Append the string stored in temporary variable m in SQ.
     \[
     SQ = SQ \ m \quad (6.1)
     \]
   - Append the string `site:puchd.ac.in` in SQ.
     \[
     SQ = SQ \ site:puchd.ac.in \quad (6.2)
     \]

2. **Display SQ**: The completely formed SQ is displayed in the text box shown in Figure 4.13(a).

**4.3.2.2 Query Execution (QE)**

After displaying SQ in the text box, NRSIP framework establishes connection with PSE and encodes SQ using UTF-8. PSE performs the parsing of the encoded SQ and initiates the search for the parsed query string.

**4.3.2.3 Result Retrieval (RR)**

Consider a search problem SP1: find Professor(s) in the field of Accounting who is/are also Chairperson of some School in Panjab University. Figure 4.13(a) highlights the various options selected by the user during query formulation as:

- in field: Accounting
- who works as: Professor
- holds additional position of: Chairperson
- associated with: School
- who is: Any

The corresponding SQ formed by NRSIP framework is displayed in the text box.

\[
SQ = \text{“show-biodata” “Accounting” “Professor” “Positions held” “Chairperson” “School” site:puchd.ac.in}
\]
Figure 4.13(a): Search for Professor(s) in the field of Accounting who is/are also Chairperson of some School in Panjab University

The results corresponding to SQ formed by NRSIP framework for search problem SP1 were retrieved using PSE (Yahoo!) as presented in Figure 4.13(b).

Figure 4.13(b): Result Set - Details of Professor(s) in the field of Accounting who are also Chairperson of some School in Panjab University
4.3.3 Experimental Setup

NRSIP framework has been designed and implemented in JAVA programming language using the following Abstract Window Toolkit (AWT) controls:

- **Frame**: It is used to define the structure for NRSIP framework.

- **Label**: The names to the seven different headings used in NRSIP framework are provided with the help of Label. As shown in Figure 4.13(a), Different labels used in the framework include:
  
  NOISE REMOVAL FROM QUERY
  Search for
  in field
  who works as
  holds additional position of
  associated with
  who is

- **TextField**: It supports entering the data. The field/subject name entered by the user is stored in this text box.

- **Choice**: It provides multiple options to the user for the following categories: who works as, holds additional position of, associated with and who is.

- **Button**: It performs the action to activate PSE and to Exit from NRSIP framework.

A click on the Button (YAHOO) referring to PSE completes the conversion of SP to SQ (Figure 4.13(a)). Finally, it sends SQ to PSE for retrieving the relevant result set.

*NRSIP framework has been executed on a Windows 7, 64 bit system environment with an Intel Core i5–2430M CPU@ 2.40 GHz with 4 GB RAM and 500 GB hard disk. The query retrieval effectiveness of NRSIP framework has been experimentally tested. The test data for NRSIP framework includes the Web documents referring to SP: find expert(s) in Panjab University. The comparative analysis of the results retrieved from PSE and NRSIP framework has also been performed.*
4.3.3.1 Tools Used

This section discusses in detail the tools used for data gathering and data analysis.

4.3.3.1.1 Web Performance Load Tester

Web Performance Load Tester (205) shown in Figure 4.14(a) was used to record the Query Retrieval Time (QRT) taken by both the approaches (PSE and NRSIP framework) in retrieving the query results for DCV = 10 (top 10 retrieved results).

![Figure 4.14(a): Working of Web Performance Load Tester](image)

Generally, users view the query results displayed only on the top in the first pages (206) and access only the first few results (207). Over 25% of users chose a website because it’s been returned as first result by the SE (207). Thus, DCV = 10 has been considered.

8Web Performance Load Tester (Version: 6.5.14205)
4.3.3.1.2 Sampling

Based on judgmental sampling, the following five search problems (SP1, SP2, SP3, SP4 and SP5) pertinent to the search scenario: find expert(s) in Panjab University were identified by the researchers in (117):

- **SP1**: Search for experts in the field of marketing who are designated as Assistant Professor and are associated with some School in Panjab University.
- **SP2**: Search for experts who are designated as Assistant Professor and also hold additional position of Chairperson in some Department in Panjab University.
- **SP3**: Search for experts in the field of pattern recognition in Panjab University.
- **SP4**: Search for experts who are designated as Associate Professor and also hold additional position of Chairperson in some Department in Panjab University.
- **SP5**: Search for experts in the field of human resource management who are designated as Professor in Panjab University.

4.3.3.1.3 Query Execution

Many studies have been conducted to assess the appropriateness of the Web results with respect to user queries. These studies have focused on IR effectiveness of SE’s from the user’s perspective (208,209). In an attempt to compare three SE’s including Google, Yahoo! and Bing, the researchers in (210) found that Google has higher precision and relative recall in comparison to Yahoo! and Bing. Moreover, Yahoo! scored ahead in precision as compared to Bing. Google and Yahoo! followed by Bing have taken a lead of other SE’s with respect to their IR effectiveness (202). Furthermore, these three SE’s also dominate the search engine market (211).

Webometric investigations have been performed by (212) to compare Google, Yahoo! and Live Search. According to the results based on single word searches, Yahoo! returned slightly more matching URL’s as compared to Google and Live Search. Besides this, Yahoo! produced result set URL’s from extensive range of domains and sites. Thus, Google has been recommended for hit count estimates and Yahoo! for all other Webometric purposes. Moreover, Google generated divergent results for symmetric property of AND operator (213).
An “e-business” report released by (214) points on failure of SE’s to satisfy the users requirements in terms of retrieving relevant results. Based on the response obtained from nearly 5000 US consumers, the report depicts that users’ satisfaction for SE’s have dropped by 1.3 points from 2016 to 2017. The two SE’s MSN and Bing have experienced the biggest decline by 4% and 3%. Customer satisfaction for Google has also dropped from 84% to 82%, thus contributing 2% decline. While, Yahoo! managed to improvise its functionalities, thus leading with a rise of 1% in its customer satisfaction. Hence, Yahoo! was selected for the experiment.

Corresponding to the five search problems stated in section 4.3.3.1.2, \( N = 5 \) queries were formulated and executed using both the approaches (PSE and NRSIP framework). Yahoo! was used to retrieve the query results during the experiment. The search results obtained and query execution time i.e. QRT (in min:sec.millisecond) for each query was recorded. The experimental data obtained is presented in tabular form using MS-Excel software. Each executed query was further replayed 9 more times for both the approaches. Different replay time for the query executed using Replay command are presented in Figure 4.14(b).
The **Replay command** available in Web Performance Load Tester (Figure 4.14(c)) was used for replaying the queries.

Figure 4.14(c): Menu highlighting the Replay command

Figure 4.15(a)-(d) exhibit the working and results obtained from Web Performance Load Tester for the search problem SP1: Search for experts in the field of marketing who are designated as Assistant Professor and are associated with some School in Panjab University.

The query executed and results obtained by PSE (Yahoo!) for SP1 are shown in Figure 4.15(a). The Testcase Properties dialog box in Figure 4.15(b) lists the different replay time for the above executed query.

Figure 4.15(c) illustrate the options selected by the user for **query formulation of SP1 using NRSIP framework**. The query formed was executed and the **results obtained by the NRSIP framework** are shown in Figure 4.15(d). The Testcase Properties dialog box in Figure 4.15(e) lists the **different replay time for the above executed query using NRSIP framework**.
Figure 4.15(a): Query Retrieval Results obtained by PSE

Figure 4.15(b): Different Replay Time for query executed in Figure 4.15(a)
Figure 4.15(c): Query Formulation (NRSIP Framework) - Search for experts in the field of marketing who are designated as Assistant Professor and are associated with some School in Panjab University

Figure 4.15(d): Query Retrieval Results obtained by NRSIP Framework
Finally, the mean of 10 QRT observations obtained from the queries executed using both the approaches was calculated. The summarized data is presented in tabular form in chapter 5. Similarly, corresponding to all other search problems mentioned in section 4.3.3.1.2, queries were executed using PSE and NRSIP framework.

4.3.3.1.4 Statistical Tool Used for Data Analysis

The 10 QRT values obtained after executing each query 10 times was used to calculate the mean QRT per query using MS-Excel Software. This process was repeated for all the 5 queries executed using both the approaches (PSE and NRSIP framework). Finally, the mean QRT calculated for each of the 5 queries was analyzed to test and compare the time effectiveness of PSE and NRSIP framework.

The result sets retrieved by both the approaches (PSE and NRSIP framework) for each query were analyzed to find the number of: Retrieved Documents Judged Relevant (RDJR), Retrieved Documents Judged Irrelevant (RDJI), Relevant Documents Retrieved (RDR) and Relevant Documents Missed (RDM).

For summarization of QRT and other retrieved results, Descriptive Statistics: Mean was used. Finally, Inferential Statistics: Independent t-test was used to measure the significance of difference between QRT and other retrieval effectiveness measures like Precision, Recall etc. for PSE and NRSIP framework. The results and comparative figures are illustrated in chapter 5.
4.3.3.1.5 Parameters Used for Experiment

The summarization and comparative analysis of the query retrieval results obtained from both the approaches (PSE and NRSIP framework) is based on the five parameters discussed below.

The graphical illustration of the comparative view of these parameters for PSE and NRSIP framework is depicted using charts prepared in MS-Excel (discussed in chapter 5).

- **Query Retrieval Time (QRT):** It refers to the time taken by the search engine for display of first result set starting from the time of issue of query to the system. Independent t-test was used to study whether there is any significant difference in the Mean QRT for PSE and NRSIP framework.

- **Precision (P):** It is defined as the ability to retrieve top-ranked documents that are mostly relevant. Precision is defined as the Proportion of Retrieved Documents Judged Relevant (RDJR).
  
  It is calculated using the formula:
  
  \[
  \frac{\text{No. of RDJR}}{\text{Total No. of Documents Retrieved}}
  \]
  
  Independent t-test was used to study whether there is any significant difference in the Precision for PSE and NRSIP framework.

- **Retrieved Documents Judged Irrelevant (RDJI):** Independent t-test was used to study whether there is any significant difference in the RDJI for PSE and NRSIP framework.

- **Recall (R):** It is the ability to find all of the relevant items in the corpus. Recall is defined as the Proportion of Relevant Documents Retrieved (RDR).
  
  It is calculated using the formula:
  
  \[
  \frac{\text{No. of RDJR}}{\text{Total No. of Relevant Documents}}
  \]
  
  Independent t-test was used to study whether there is any significant difference in the Recall for PSE and NRSIP framework.

- **Relevant Documents Missed (RDM):** Independent t-test was used to study whether there is any significant difference in the RDM for PSE and NRSIP framework.
4.3.3.1.6 Hypotheses

Table 4.1 enumerates the five hypotheses formulated using the parameters discussed in section 4.3.3.1.5. Null Hypothesis was formulated for each parameter. The results derived from the statistical test are exemplified in chapter 5.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hypothesis</th>
<th>Test Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$H_0$ (1): There is no significant difference in the Query Retrieval Time (QRT) for PSE and NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>$H_0$ (1): QRT (PSE) = QRT (NRSIP)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$H_0$ (2): There is no significant difference in the Proportion of Retrieved Documents Judged Relevant (RDJR) for PSE and NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>$H_0$ (2): Proportion of RDJR (PSE) = Proportion of RDJR (NRSIP) or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>There is no significant difference in the Precision (P) for PSE and NRSIP framework.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$H_0$ (2): Precision (PSE) = Precision (NRSIP)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$H_0$ (3): There is no significant difference in the Proportion of Retrieved Documents Judged Irrelevant for PSE and NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>$H_0$ (3): Proportion of RDJI (PSE) = Proportion of RDJI (NRSIP)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$H_0$ (4): There is no significant difference in the Proportion of Relevant Documents Retrieved (RDR) for PSE and NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>$H_0$ (4): Proportion of RDR (PSE) = Proportion of RDR (NRSIP) or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>There is no significant difference in the Recall (R) for PSE and NRSIP framework.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$H_0$ (4): Recall (PSE) = Recall (NRSIP)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$H_0$ (5): There is no significant difference in the Proportion of Relevant Documents Missed (RDM) for PSE and NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>$H_0$ (5): Proportion of RDM (PSE) = Proportion of RDM (NRSIP)</td>
<td></td>
</tr>
</tbody>
</table>
4.3.3.1.7 NRSIP Framework for Extended Dataset

Section 4.3.3.1.2 to 4.3.3.1.6 illustrates the execution of experimental test queries on PSE and NRSIP framework for five search problems: SP1, SP2, SP3, SP4 and SP5.

To provide better representation of the population of the study, the authors have extended the experimental dataset from N=5 to N=30.

This section discusses the NRSIP framework with extended dataset i.e. more data variations and larger sample set (N=30).

Based on judgmental sampling, the following 30 search problems:

SPE1, SPE2, ……, SPE30

pertinent to the search scenario: find expert(s) in Panjab University were identified by the authors:

- **SPE1**: Search for experts in the field of Marketing who are designated as Assistant Professor and are associated with some School in Panjab University.
- **SPE2**: Search for experts in the field of Police Administration who are designated as Assistant Professor and are associated with some Centre in Panjab University.
- **SPE3**: Search for experts in the field of Constitutional Law who are designated as Assistant Professor and are associated with some Institute in Panjab University.
- **SPE4**: Search for experts in the field of Data Structure who are designated as Assistant Professor and are associated with some School in Panjab University.
- **SPE5**: Search for experts in the field of Organic Chemistry who are designated as Assistant Professor and are associated with some Department in Panjab University.
- **SPE6**: Search for experts in the field of Pattern Recognition who are designated as Assistant Professor and are associated with some Department in Panjab University.
- **SPE7**: Search for experts in the field of E-Commerce who are designated as Assistant Professor and are associated with some School in Panjab University.
SPE8: Search for experts in the field of Human Rights who are designated as Assistant Professor and are associated with some Department in Panjab University.

SPE9: Search for experts in the field of Molecular Biology who are designated as Assistant Professor and are associated with some Department in Panjab University.

SPE10: Search for experts in the field of Microeconomics who are designated as Assistant Professor and are associated with some Department in Panjab University.

SPE11: Search for experts who are designated as Assistant Professor and also hold additional position of Chairperson in some Department in Panjab University.

SPE12: Search for experts who are designated as Associate Professor and also hold additional position of Chairperson in some Department in Panjab University.

SPE13: Search for experts who are designated as Professor and also hold additional position of Chairperson in some Department in Panjab University.

SPE14: Search for experts who are designated as Assistant Professor and also hold additional position of Coordinator in some Centre in Panjab University.

SPE15: Search for experts who are designated as Associate Professor and also hold additional position of Coordinator in some Centre in Panjab University.

SPE16: Search for experts who are designated as Professor and also hold additional position of Coordinator in some Centre in Panjab University.

SPE17: Search for experts who are designated as Professor and also hold additional position of Dean in Panjab University.

SPE18: Search for experts who are designated as Associate Professor and also hold additional position of Director in Panjab University.

SPE19: Search for experts who are designated as Professor and also hold additional position of Director in Panjab University.

SPE20: Search for experts who are designated as Professor and also hold additional position of Director in some Institute in Panjab University.
- **SPE21**: Search for experts in the field of **Pattern Recognition** in Panjab University.
- **SPE22**: Search for experts in the field of **Environmental Geography** in Panjab University.
- **SPE23**: Search for experts in the field of **Theatre** in Panjab University.
- **SPE24**: Search for experts in the field of **Algebra** in Panjab University.
- **SPE25**: Search for experts in the field of **Teacher Education** in Panjab University.
- **SPE26**: Search for experts in the field of **Statistics** in Panjab University.
- **SPE27**: Search for experts in the field of **Economics** in Panjab University.
- **SPE28**: Search for experts in the field of **Cultural Studies** in Panjab University.
- **SPE29**: Search for experts in the field of **Organizational Behaviour** in Panjab University.
- **SPE30**: Search for experts in the field of **Nanotechnology** in Panjab University.

Experimental results were retrieved using PSE = Yahoo! and NRSIP framework. The queries formulated for each of the 30 search problems: SPE1, SPE2, ……, SPE30) were executed 10 times using both the approaches (PSE and NRSIP framework). The retrieved result sets for these 30 queries have been analyzed using human relevancy judging.

Summarization and comparative analysis of the query retrieval results obtained from both the approaches (PSE and NRSIP framework) for extended data set (N=30) have been based on the following parameters:

- Query Retrieval Time (QRT)
- Precision (P)
- Retrieved Documents Judged Irrelevant (RDJI)
- Recall (R)

These parameters have been discussed in section 4.3.3.1.5. The graphical illustration of the comparative view of these parameters for PSE and NRSIP framework is depicted using charts in chapter 5 (section 5.4). The results obtained for these parameters are presented in tabular form. Comparative analysis of the effectiveness of PSE and NRSIP framework is presented in section 5.4.1, 5.4.2 and 5.4.3.
Mean and Independent t-test has been used for analysis. The following hypotheses have been used for performance evaluation of PSE and NRSIP framework for extended dataset. Table 4.2 enumerates the hypotheses formulated for comparative analysis of PSE and NRSIP framework for extended dataset. Null Hypothesis was formulated for each parameter. The results derived from the statistical test are exemplified in chapter 5.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hypothesis</th>
<th>Test Used</th>
</tr>
</thead>
</table>
| 1     | $H_0$ (E1): There is no significant difference in the Query Retrieval Time (QRT) for PSE and NRSIP framework for extended dataset.  
\[ H_0 (E1): \text{QRT (PSE)} = \text{QRT (NRSIP)} \] | Independent t-test |
| 2     | $H_0$ (E2): There is no significant difference in the Proportion of Retrieved Documents Judged Relevant (RDJR) for PSE and NRSIP framework for extended dataset.  
\[ H_0 (E2): \text{Proportion of RDJR (PSE)} = \text{Proportion of RDJR (NRSIP)} \] \text{or}  
\[ H_0 (E2): \text{Precision (PSE)} = \text{Precision (NRSIP)} \] | Independent t-test |
| 3     | $H_0$ (E3): There is no significant difference in the Proportion of Retrieved Documents Judged Irrelevant for PSE and NRSIP framework for extended dataset.  
\[ H_0 (E3): \text{Proportion of RDJI (PSE)} = \text{Proportion of RDJI (NRSIP)} \] | Independent t-test |
| 4     | $H_0$ (E4): There is no significant difference in the Proportion of Relevant Documents Retrieved (RDR) for PSE and NRSIP framework for extended dataset.  
\[ H_0 (E4): \text{Proportion of RDR (PSE)} = \text{Proportion of RDR (NRSIP)} \] \text{or}  
\[ H_0 (E4): \text{Recall (PSE)} = \text{Recall (NRSIP)} \] | Independent t-test |
4.3.3.1.8 Generalized NRSIP Framework (NRSIP-G)

Previous sections illustrate the working, structure and experimental setup of NRSIP framework for search scenario: find expert(s) in Domain = Panjab University. For pilot study, pretesting of NRSIP framework was performed for N=5 search problems: SP1,SP2,……,SP5 as discussed in section 4.3.3.1.2.

On completion of pilot study and successful pretesting of NRSIP framework, the authors utilized larger and varied datasets with N=30 search problems: SPE1,SPE2,……,SPE30 for experimentation of NRSIP framework for the above mentioned search scenario. This has been discussed in section 4.3.3.1.7.

Though NRSIP framework had been designed and implemented for general queries i.e. finding expert(s) in a specified field without fixing any Domain. But, due to the difficulty in calculating Recall for an open set, it was initially tested for a specific Domain i.e. Panjab University.

On successful evaluation of the NRSIP framework for Domain = Panjab University, the authors then evaluated it for general queries i.e. without fixing any Domain. NRSIP framework not only works for a specific domain, but it also facilitates the user to search for expert(s) in Higher Educational Institutions (HEI) around the World.

The present section discusses the working, structure and experimental setup for Generalized NRSIP framework (NRSIP-G) implemented to search for experts in HEI around the World i.e. without fixing Domain. This section describes N=30 generalized search problems referred as: SPG1, SPG2,……,SPG30.

Query 1G(a), 2G(a), …… ,30G(a) are the 30 general queries formulated by the user for the 30 search problems SPG1, SPG2, …… ,SPG30 to be executed using PSE (Yahoo!). Query 1G(b), 2G(b), …… ,30G(b) are the 30 general queries formulated and executed using Generalized NRSIP framework for the 30 search problems. Generalized NRSIP framework has been evaluated for the parameters: Query Retrieval Time (QRT) and Precision (P). The experimental data for these parameters and comparative analysis is presented in chapter 5.
4.3.3.1.8.1. Working of NRSIP-G Framework

As shown in Figure 4.15(f), Generalized NRSIP framework allows the user to find the expert(s) in HEI around the World.

- in a particular field (subject area);
- with specific designation;
- holding additional position;
- associated with some specific division (Centre, Chair, Department, Institute, School);
- with or without Doctorate Degree.

![Image: Categorical Search used in Generalized NRSIP framework for SPG (find expert(s) in HEI around the World)]

Figure 4.15(f): Categorical Search used in Generalized NRSIP framework for SPG (find expert(s) in HEI around the World)

4.3.3.1.8.2 Structure of NRSIP-G Framework

The modular composition of Generalized NRSIP framework also consists of three modules as discussed in section 4.3.2. These modules include:

- Query Formulation (QF)
- Query Execution (QE)
- Result Retrieval (RR)
4.3.3.1.8.2.1 Query Formulation (QF)

The module QF facilitates the user to specify the conceptual search query for **SPG: find expert(s) in HEI around the World**. The module QF consists of a text box that allows the user to enter the text corresponding to the field (subject). It also provides several drop down list box with multiple options to enable the user to perform the selection process. As the user enters the text and/or selects the options, the framework converts SPG to SQ by mapping the Concept and Context of SPG. The mapping removes the noise and ensures that the Semantic Information Processing (SIP) of SQ results in relevant retrieval.

**Variables used**

- Temporary variable m: The data type of variable m is string. The process begins by initializing the temporary variable m with an empty string i.e. m="".
- Variable SQ: It is a string type variable used to store the Semantic Query (SQ) formed by the framework.
The following two steps are performed by the module QF to generate SQ:

**Step 1: Concept Mapping**
As discussed in (116), CMK can be used to map the Concept of SP. In this case, “faculty-profile” is the CMK used for Concept Mapping. This is done by initializing SQ with the string “faculty-profile”.

\[ \text{SQ} = \text{“faculty-profile”} \]  \hspace{1cm} (1G)

As shown in Equation (1G), the keyword “faculty-profile” maps the Concept for SP and ensures to retrieve results only referring to human experts.

**Step 2: Context Mapping**
The following sub-modules perform the Context Mapping:

1. **in field:** It allows the user to enter the field/subject name for which the expert(s) is/are to be searched. The entered field is stored in a Text String (TS). NRSIP framework appends the entered string TS in SQ in Equation (1G) as follows:

\[ \text{SQ} = \text{SQ} + \text{TS} \]  \hspace{1cm} (1G.1)

\[ \text{SQ} = \text{“faculty-profile”} + \text{TS} \]

Here, TS may refer to any field/subject like “Accounting”, “Data Structure”, “Marketing”, “Mathematics” etc.

**Example:** In case the user enters “Marketing” as the field/subject, then:

\[ \text{SQ} = \text{“faculty-profile”} + \text{“Marketing”} \]  \hspace{1cm} (E1G)

If the user does not enter any field i.e. TS is not entered, then SQ is not appended.

2. **who works as:** It provides itemization of different possible designation options as presented in Figure 4.15(h). This enables the user to retrieve the details of expert(s) with a specific designation like Assistant Professor, Associate Professor, Professor.

![Figure 4.15(h): User Input Module for sub-module: who works as](image)

(Designation of the expert(s) in the HEI around the World)
In case designation is not known, then the user can choose a semantically equivalent option called “Faculty”. This retrieves the results corresponding to all the Teaching Employees irrespective of their designation.

Based on the option chosen by the user from among those listed in Figure 4.15(h), Mapping of this Context is performed by executing append action on SQ in Equation (1G.1) or Equation (1G) as illustrated in Figure 4.15(i).

![Flow Chart: who works as (Designation of the expert(s) in the HEI around the World)](image)

**Figure 4.15(i): Flow Chart: who works as (Designation of the expert(s) in the HEI around the World)**

**a) Faculty:** As shown in the logical flow diagram in Figure 4.15(i), this is the default option for sub-module: who works as. On selecting this option, the framework appends the string “* Professor” in SQ.

\[
\text{SQ} = \text{“faculty-profile” + TS + “*Professor”} \quad \text{(2G.1)}
\]

\[
\text{SQ} = \text{“faculty-profile” “* Professor”} \quad \text{(2G.2)}
\]

**Outcome of (2G.1) and (2G.2)**

Equation (2G.1) retrieves the details of all the expert(s) (Teaching Employees) in the specified field TS.

Equation (2G.2) retrieves the details of all the expert(s) (Teaching Employees) irrespective of the field TS because the user has not specified any field in this case.
b) **Assistant Professor:** Append the string “Assistant Professor” in SQ.

\[
\text{SQ} = \text{“faculty-profile”} + \text{TS} + \text{“Assistant Professor”} \quad (2G.3)
\]

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Assistant Professor”} \quad (2G.4)
\]

**Outcome of (2G.3) and (2G.4)**

Equation (2G.3) retrieves the details of expert(s) (Teaching Employees - only Assistant Professor) in the specified field TS.

Equation (2G.4) retrieves the details of expert(s) (Teaching Employees - only Assistant Professor) irrespective of the field TS.

**Example:** On selecting the option “Assistant Professor”, Equation (E1G) becomes:

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Marketing”} + \text{“Assistant Professor”} \quad (E2G)
\]

Equation (E2G) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing”.

c) **Associate Professor:** Append the string “Associate Professor” in SQ and update the temporary variable m with the string “–Assistant”.

\[
\text{SQ} = \text{“faculty-profile”} + \text{TS} + \text{“Associate Professor”} \quad (2G.5)
\]

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Associate Professor”} \quad (2G.6)
\]

**Outcome of (2G.5) and (2G.6)**

Equation (2G.5) retrieves the details of expert(s) (Teaching Employees - only Associate Professor) in the specified field TS.

Equation (2G.6) retrieves the details of expert(s) (Teaching Employees - only Associate Professor) irrespective of the field TS.

d) **Professor:** Append the string “Professor” in SQ and update temporary variable m with the string “–Assistant –Associate”.

\[
\text{SQ} = \text{“faculty-profile”} + \text{TS} + \text{“Professor”} \quad (2G.7)
\]

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Professor”} \quad (2G.8)
\]

**Outcome of (2G.7) and (2G.8)**

Equation (2G.7) retrieves the details of expert(s) (Teaching Employees - only Professor) in the specified field TS. Equation (2G.8) retrieves the details of expert(s) (Teaching Employees – only Professor) irrespective of the field TS.
3. **holds additional position of**: Teaching employees in the university may also be given some additional charge and hence they may hold certain additional position.

**Example:** An Assistant Professor of some Department in the University may also be working as Chairperson of the Department.

For this scenario, Generalized NRSIP framework provides a set of several designation options as shown in Figure 4.15(j) to accomplish more specific results.

![Figure 4.15(j): User Input Module for sub-module: holds additional position of](image)

*(Additional position of the expert(s) in the HEI around the World)*

**a) None:** Being set as the default option, no append action is performed in SQ on selecting None.

\[ SQ = SQ \]  \hspace{1cm} (3G.1)

**b) Chairperson / Coordinator / Dean / Director:** The flow chart illustrated in Figure 4.15(k) brings to light the different actions performed by the framework in response to a particular option selected.

Append the string corresponding to the selected option in SQ as follows:

\[ SQ = SQ + \text{“Chairperson”} \]

\[ SQ = SQ + \text{“Coordinator”} \]

\[ SQ = SQ + \text{“Dean”} \]

\[ SQ = SQ + \text{“Director”} \]  \hspace{1cm} (3G.2)

**Outcome of (3G.2)**

Equation (3G.2) retrieves the details of expert(s) who also hold some additional position like Chairperson or Coordinator or Dean or Director.
Example: Selecting the option “Chairperson”, SQ in Equation “(E2G)” becomes:

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Marketing”} + \text{“Assistant Professor”} + \text{“Chairperson”}
\]

(E3G)

Equation (E3G) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing” and hold the position of “Chairperson”.

To retrieve the list of all the expert(s) designated as “Chairperson”, the following SQ can be used.

\[
\text{SQ} = \text{“faculty-profile”} + \text{“Professor”} + \text{“Chairperson”}
\]

Figure 4.15(k): Flow Chart: holds additional position of

(Additional position of the expert(s) in the HEI around the World)

4. associated with: It supports the user to access the details of expert(s) associated with some specific division of the University.

Figure 4.15(l): User Input Module for sub-module: associated with

(Search expert(s) in HEI around the World associated with a specific division)
No append action is performed in SQ on selecting the **default option: Any**.

On selection of any option other than None, the framework appends the string `intitle:` in SQ. This augments Context Mapping.

\[
\text{SQ} = \text{SQ} + \text{intitle:} \quad (4G)
\]

For any other selected option, the framework appends the string corresponding to the selected option in SQ as follows:

\[
\begin{align*}
\text{SQ} &= \text{SQ} + \text{intitle: Centre} \\
\text{SQ} &= \text{SQ} + \text{intitle: Chair} \\
\text{SQ} &= \text{SQ} + \text{intitle: Department} \\
\text{SQ} &= \text{SQ} + \text{intitle: Institute} \\
\text{SQ} &= \text{SQ} + \text{intitle: School}
\end{align*}
\]

**Outcome of (4G)**

Equation (4G) retrieves the details of expert(s) associated with a specific division of the HEI around the World.

**Example:** On selecting the option “Department”, the corresponding SQ in Equation (E3G) becomes:

\[
\text{SQ} = \text{“faculty-profile” + “Marketing” + “Assistant Professor” + “Chairperson” + intitle: Department} \quad (E4G)
\]

Equation (E4G) retrieves the details of Assistant Professor(s) who is/are expert(s) in the field of “Marketing” and hold the position of “Chairperson” of a “Department”.

5. **who is:** This sub-module retrieves the details of expert(s) who hold or do not hold the Doctorate Degree. For **default option: Any**, no append action is performed. Append action is performed for the other two options (Figure 4.15(m)) as follows:

\[
\begin{align*}
\text{Doctorate}: \quad \text{SQ} &= \text{SQ} + \text{“Dr.”} \\
\text{Not Doctorate}: \quad \text{SQ} &= \text{SQ} - \text{Dr.} \quad (5G)
\end{align*}
\]

**Figure 4.15(m): User Input Module for sub-module: who is**

(Search expert(s) in the HEI around the World who are Doctorate or not)
The above-mentioned Context Mapping steps (1 to 5) can be executed in any sequence. Moreover, it is not mandatory to perform all the steps.

The user terminates the choice selection process by clicking on the Button (YAHOO) of PSE as shown in Figure 4.15(g).

As a corollary, it accomplishes these two actions:

3. **Complete SQ**: Generalized NRSIP framework performs the following action:
   - Append the string stored in temporary variable m in SQ.
   
   \[
   SQ = SQ \ m \quad (6G)
   \]

4. **Display SQ**: Completely formed SQ is displayed in the text box shown in Figure 4.15(g).

### 4.3.3.1.8.2.2 Query Execution (QE)

After displaying SQ in the text box, Generalized NRSIP framework establishes connection with PSE and encodes SQ using UTF-8. PSE performs the parsing of the encoded SQ and initiates the search for the parsed query string.

### 4.3.3.1.8.2.3 Result Retrieval (RR)

Consider a search problem SPG1: Search for experts in the field of **Marketing** who are designated as **Assistant Professor** and are associated with some **School**. Figure 4.15(n) highlights the various options selected by user during query formulation as:
- in field: Marketing
- who works as: Assistant Professor
- holds additional position of: Chairperson
- associated with: School
- who is: Any

The corresponding SQ formed by Generalized NRSIP framework is displayed in the text box.

\[
SQ = \text{“faculty-profile”} + \text{“Marketing”} + \text{“Assistant Professor”} + \text{“Chairperson”} \ \intitle{School}
\]
Figure 4.15(n): Search for Assistant Professor(s) in the field of Marketing who is/are also Chairperson of some School in the HEI around the World

The results corresponding to SQ formed by NRSIP framework for search problem SP1 were retrieved using PSE (Yahoo!) as presented in Figure 4.15(o).

Figure 4.15(o): Result Set - Details of Assistant Professor(s) in the field of Marketing who are also Chairperson of some School in HEI around the World
This framework has been executed on a Windows 7, 64 bit system environment with an Intel Core i5–2430M CPU@ 2.40 GHz with 4 GB RAM and 500 GB hard disk. The query retrieval effectiveness of Generalized NRSIP framework has been experimentally tested.

The test data for Generalized NRSIP framework includes the Web documents referring to SPG: find expert(s) in HEI around the World.

Web Performance Load Tester (205) was used to record the Query Retrieval time (QRT). The comparative analysis of the results retrieved from PSE and Generalized NRSIP framework has also been performed.

4.3.3.1.8.2.4 Sampling

- **SPG1**: Search for experts in the field of **Marketing** who are designated as **Assistant Professor** and are associated with some **School**.
- **SPG2**: Search for experts in the field of **Police Administration** who are designated as **Assistant Professor** and are associated with some **Centre**.
- **SPG3**: Search for experts in the field of **Constitutional Law** who are designated as **Assistant Professor** and are associated with some **Institute**.
- **SPG4**: Search for experts in the field of **Data Structure** who are designated as **Assistant Professor** and are associated with some **School**.
- **SPG5**: Search for experts in the field of **Organic Chemistry** who are designated as **Assistant Professor** and are associated with some **Department**.
- **SPG6**: Search for experts in the field of **Pattern Recognition** who are designated as **Assistant Professor** and are associated with some **Department**.
- **SPG7**: Search for experts in the field of **E-Commerce** who are designated as **Assistant Professor** and are associated with some **School**.
- **SPG8**: Search for experts in the field of **Human Rights** who are designated as **Assistant Professor** and are associated with some **Department**.
- **SPG9**: Search for experts in the field of **Molecular Biology** who are designated as **Assistant Professor** and are associated with some **Department**.
- **SPG10**: Search for experts in the field of **Microeconomics** who are designated as **Assistant Professor** and are associated with some **Department**.
SPG11: Search for experts who are designated as Assistant Professor and also hold additional position of Chairperson in some Department.

SPG12: Search for experts who are designated as Associate Professor and also hold additional position of Chairperson in some Department.

SPG13: Search for experts who are designated as Professor and also hold additional position of Chairperson in some Department.

SPG14: Search for experts who are designated as Assistant Professor and also hold additional position of Coordinator in some Centre.

SPG15: Search for experts who are designated as Associate Professor and also hold additional position of Coordinator in some Centre.

SPG16: Search for experts who are designated as Professor and also hold additional position of Coordinator in some Centre.

SPG17: Search for experts who are designated as Professor and also hold additional position of Dean.

SPG18: Search for experts who are designated as Associate Professor and also hold additional position of Director.

SPG19: Search for experts who are designated as Professor and also hold additional position of Director.

SPG20: Search for experts who are designated as Professor and also hold additional position of Director in some Institute.

SPG21: Search for experts in the field of Pattern Recognition.

SPG22: Search for experts in the field of Environmental Geography.

SPG23: Search for experts in the field of Theatre.

SPG24: Search for experts in the field of Algebra.

SPG25: Search for experts in the field of Teacher Education.

SPG26: Search for experts in the field of Statistics.

SPG27: Search for experts in the field of Economics.

SPG28: Search for experts in the field of Cultural Studies.

SPG29: Search for experts in the field of Organizational Behaviour.

SPE30: Search for experts in the field of Nanotechnology.
Experimental results have been retrieved using PSE = Yahoo! and Generalized NRSIP framework. The queries formulated for each of the 30 search problems: SPG1, SPG2, ……, SPG30 were executed 5 times using both the approaches (PSE and Generalized NRSIP framework). The retrieved result sets for these queries have been analyzed using human relevancy judging.

Summarization and comparative analysis of the query retrieval results have been based on the following parameters: Query Retrieval Time (QRT), Precision (P). The graphical illustration of the comparative view of these parameters is depicted using charts in chapter 5. Mean and Independent t-test has been used for analysis. Performance evaluation of PSE and Generalized NRSIP framework has been based on the following hypotheses.

4.3.3.1.8.2.5 Hypotheses

Table 4.3 enumerates the hypotheses formulated for comparative analysis of PSE and NRSIP-G framework. Null Hypothesis was formulated for each parameter. Results derived from the statistical test are exemplified in chapter 5 in section 5.5.

<p>| Table 4.3: Hypotheses Formulated for Data Analysis of PSE and NRSIP-G Framework |
|---------------------------------|---------------------------------|-----------------|</p>
<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hypothesis</th>
<th>Test Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H₀ (G1): There is no significant difference in the Query Retrieval Time (QRT) for PSE and Generalized NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>H₀ (E1): QRT (PSE) = QRT (NRSIP-G)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>H₀ (G2): There is no significant difference in the Proportion of Retrieved Documents Judged Relevant (RDJR) for PSE and Generalized NRSIP framework.</td>
<td>Independent t-test</td>
</tr>
<tr>
<td></td>
<td>H₀ (E2): Proportion of RDJR (PSE) = Proportion of RDJR (NRSIP-G) or There is no significant difference in the Precision (P) for PSE and Generalized NRSIP framework.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H₀ (E2): Precision (PSE) = Precision (NRSIP-G)</td>
<td></td>
</tr>
</tbody>
</table>
4.4 Cognitive Information Mapping Model (CIMM) for Web-based Search (118)

To assist the user in the search process, Section 4.4 vindicates the design of a Cognitive Information Mapping Model (CIMM) (118) using pictorial representations and suitable algorithms that describe the logic. Focusing on the challenges mentioned in section 3.5, CIMM has been proposed by (118) to provide assistance to the user in the Web-based search process. The guidance provided by CIMM is based on the cognitive style elicitation of the user. During the search process, the user performs different types of activities that include: entering query for retrieval and selecting resource(s) from the retrieved result set. As a by-product of search process, user experiences different types of emotions. It has been found that emotions impact the search process and can help in evaluating the search experience to finally generate consequent coping behaviors (12). Moreover, negative feelings like fear, nervousness and anxiety generated during user-computer interaction has been found to greatly affect the performance of the user in computer related tasks and other affective processes (110).

Thus, taking in to account the dependence of Web-based performance on the cognitive style of the user and the emotions experienced in the search process; the proposed CIMM has been based on Cognitive Mapping. The subsequent sections elaborate the working and modular structure of CIMM.

4.4.1 Working of CIMM (118)

During the search process, the user performs various activities on the resources retrieved and experience different types of emotions. As shown in Figure 4.16, CIMM gathers the user activities and emotions experienced. The user information gathered is stored and cognitively mapped to provide suggestions during the search process. In context of the work in (118), Cognitive Mapping refers to the mapping performed using the concept of Belief-Desire-Intention (BDI) model (88). This mapping is performed in a way analogous to how humans perform a task starting with the Belief(s), then performing the activities (Intention(s)), learning from these activities and finally reaching the goal (Desire(s)). The mapped information provided
by CIMM yields search suggestions/recommendations for the user during the search process. Thus, it offers personalized search based on Cognitive Information Mapping (CIM).

**Figure 4.16: Cognitive Information Mapping Model (CIMM)**

### 4.4.2 Modular Structure of CIMM (118)

The high-level view of CIMM illustrated in Figure 4.17 consists of five modules namely: Search Activity (SA), Capture, Store, Decision Making and Search Suggestions. CIMM adopts a user-focused information processing approach.
4.4.2.1 Search Activity (SA)

The purpose of this module is to support the user in initiating a search task. The user-friendly interface of CIMM facilitates the user to either enter the query (Q) in the text box or select Q from the existing list of previously searched queries stored in the Database shown in Figure 4.17. **The working of this module is explained using the Algorithm: Search_Activity( ).**

The following variables describe the logic of Algorithm: Search_Activity()

Q-Query, QL-Query List, RL-Resource List, SE-Search Engine, Q.ACTV-Activity Type performed on Q, Q.CON-Continue the search using Database or initiate new search using SE.

**Working of the Algorithm: Search_Activity()**

**Step 1: Initialize variables**

Initially the variable Q.ACTV is set to 0 i.e. no activity is performed by the user. Similarly, Q.CON = 0 i.e. RL will be displayed using SE.

**Step 2: Check the Activity Type performed by the user**

The logical flow diagram of Step 2 (Activity Type) is illustrated in Figure 4.18. If the user performs any activity by entering Q or selecting Q from QL, the value of the variable Q.ACTV is reset as follows:

a. If Q is entered, then: Reset Q.ACTV = 1. [new query is entered by the user]

b. If Q is selected from QL, then: Reset Q.ACTV = 2.

[existing query selected from QL]

![Logical Flow Diagram for Step 2 (Activity Type) of Algorithm: Search_Activity()](image)

**Figure 4.18: Logical Flow Diagram for Step 2 (Activity Type) of Algorithm: Search_Activity( )**
Step 3: Check the availability of Q in the Database

Based on the Activity Type performed in Step 2 of Algorithm: Search_Activity( ), Step 3 is performed accordingly.

a. If in Step 2, new Q is entered, then: Q.ACTV = 1. [Activity Type = New]
   i. Use Function Found( ) to check whether Q entered by the user already exists in the Database or not.
   ii. If Function Found( ) returns True (i.e. Q is found in the Database and Q exists in QL), then: provide options to the user to either select Q from QL or start a new search for Q using SE.
   iii. Use Function Conti(Q) to check if the user selects Q from QL or not i.e. the user wants to continue the search using existing RL or not.
   iv. If Function Conti(Q) returns True, (i.e. user selects Q from QL), then: Reset Q.CON = 1(Q is selected from QL). Display existing resources present in RL.
   v. If Function Conti(Q) returns False, then: Q.CON remains same and is not reset. In this case, start the search for Q using SE. RL is displayed using SE.

b. If in Step 2, user selects Q from QL, then: Q.ACTV = 2. [Activity Type = Existing]
   i. Goto Step a.iii.

| If (Q.CON = 1), then: CIMM displays RL from Database i.e. list of resources previously accessed corresponding to Q. |
| If (Q.CON = 0), then: CIMM displays RL from SE. |

While displaying RL, CMMM highlights the activities previously performed on the resources i.e. whether the resource was only read by the user or any other activity (save etc.) was also performed. CMMM also specifies about the usefulness of the resource depending on the emotions attached to the resource by the user during the previous search process. This facilitates the user in prioritizing the order for accessing the resources according to their importance specified in RL by the user.

During the search process, the user might encounter a resource previously found not useful or frustrating. To avoid repeated access to such a resource, CMMM captures the emotions of the user while initiating the query and accessing the resource. This serves as useful criteria for the user while selecting the resource from RL as it saves time.
**Algorithm: Search_Activity()**

It describes the different search activities performed and subsequent actions.

1. **Q.ACTV = 0, Q.CON = 0.** [Initialize variables for Q]

2. [Check the Activity Type performed]
   [Activity can be New, Existing or no activity]

   If (Q is entered in the text box), then:
   
   Q.ACTV = 1. [Activity is New and Reset Q.ACTV]

   Else
   
   { If (Q is selected from QL), then:
     
     Q.ACTV = 2. [Activity is Existing, Reset Q.ACTV]
     [End of If structure]
   
   } [End of If structure] [If (Q.ACTV == 0), then: no activity is performed by user]

3. If (Q.ACTV == 1), then:[Activity is New]

   [Check Q in the Database using Function Found( )]

   { If (Found(Q) == True), then: [Q exists in the Database]

     { a.Highlight Q in QL.[user can select Q from QL or start new search for Q]
       b.If (Conti(Q) == True), then:
          [Function Conti(Q) checks if user selects Q from QL]
          Q.CON = 1.[Reset Q.CON][RL will be displayed using Database]
          Else [No change in Q.CON and Q.CON = 0] [RL displayed using SE]
          [End of If structure]

     } [End of If structure]

   Else [Q does not exist in the Database][No change in Q.CON and Q.CON= 0]

   [RL will be displayed using SE]
   [End of If structure]

   }
Else
{
    If (Q.ACTV == 2), then: [Activity is Existing and Q is selected from QL]
    {
        If (Conti(Q) == True), then:
        [Function Conti(Q) checks if user selects Q from QL]
        Q.CON = 1. [Reset Q.CON]
        [RL displayed using Database]
        Else
        [No change in Q.CON, it remains 0][RL displayed using SE]
        [End of If structure]
    }
    [End of If structure]
}
[End of If structure]

/* End of Algorithm Search_Activity*/

Based on the above-discussed two search activities (New and Existing), CIMM captures the user activities and emotions experienced using the module Capture.

4.4.2.2 Capture

When some activity is accomplished by the user in the module Search Activity, then CIMM activates the module Capture. The value of the variables Q.ACTV and Q.CON are passed from the module Search Activity to the module Capture.

*Algorithm: Capture_Activity(Q) captures the information associated with the search process initiated by the user.*

4.4.2.3 Store

This module stores the data gathered by the module Capture in the Database. The gathered data includes queries initiated, resources accessed, activities performed on the resources, logstamp and emotions experienced by the user during the search process. This serves as the base for the module Decision Making.
Algorithm: Capture_Activity(Q)

1. Call Search_Activity(). [check if the user enters Q or selects Q from QL]
   [Check Activity Type = New or Existing ?]

2. If (Action is performed on Q) i.e. If (Q.ACTV = 1 or Q.ACTV = 2) , then:
   
   a. Store details of Q in the Database using the module Store.
   b. [Show the list of resources RL for Q depending on Q.CON]
      If Q.CON = 1 , then:
      Display RL from the Database.
      Else [If Q.CON = 0]
      Display RL from SE. [Q is searched using SE]
      [End of If structure]

   c. Repeat until the user closes RL window for Q
      
      For every Resource (RS) selected from RL for Q , store details of
      RS in the Database using the module Store.

      For any activity performed on RS, store details of RS in the
      Database using the module Store.

      If the user closes any RS, then:
      
      Call module Store to store details of RS in the Database.
      Close RS.
      
      [End of If structure]

      [End of Repeat until loop]

      [End of If structure]

/* End of Algorithm Capture_Activity */
4.4.2.4 Decision Making

For making decisions, CIMM uses Cognitive Mapping based on Belief-Desire-Intention (BDI) model as illustrated in Figure 4.19. It analyses the data from the perspective of queries issued, resources accessed, user emotions experienced and timestamp. It uses rule-based mechanisms and data mining techniques to recognize regularities in the user’s path to reach the resource. This forms the basis to develop the user model.

![Diagram of Cognitive Mapping based on BDI Model](image)

**Figure 4.19: Cognitive Mapping based on BDI Model**

4.4.2.5 Search Suggestions

The data analyzed by the module Decision Making assists the user and provides valuable suggestions regarding the queries initiated, resources accessed and emotions experienced during the search process. It indicates whether the query already exists in the Database or not. If the query is found, then it lists all the resources for the query from RL and highlights the resources liked and disliked by the user.

**Example to demonstrate the working of CIMM**

Search scenario (S) is used to explain the working of the proposed CIMM.

Scenario S = Search for the information regarding Carbon Footprint.

1. User entered Q: Carbon Footprint in the text box provided as shown in Figure 4.20.
2. CIMM checks for the existence of Q in the Database in QL. As Q is found in the existing list of queries in QL at S.No. 2, so, CIMM highlights Q.

3. As the user selects Q: Carbon Footprint (at S.No. 2) from QL, CIMM displays RL (list of resources previously accessed by the user for Q).

4. User selects R1 (first resource accessed) from RL. The resource is opened for reading and details of R1 are stored in Database. As asserted by the user, the emotions experienced while reading R1 were Happy (H).

5. From within R1, user now selects R2 (second resource accessed) and the details of R2 are stored in the Database. As asserted by the user, the emotions experienced while reading R2 were Happy (H).

6. From within R2 user selects R3 (third resource accessed). CIMM now suggests the user not to access the resource R3 because during the previous search process, the user had tagged R3 with emotion Sad (S).

Hence, it can be concluded that R3 is not a useful resource as it resulted in Sad (S) emotions for the user. Similar emotions of Sadness were experienced for the resource R5 (fifth resource accessed). Thus, CIMM suggests the user not to access the undesired resources R3 and R5. Moreover, as R4 (fourth resource accessed) was previously saved by the user and it concluded with Happy (H) emotions so, it can be a suggested link for the user whenever Q is initiated.

<table>
<thead>
<tr>
<th>S. No</th>
<th>Query</th>
<th>Logstamp</th>
<th>Emotion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BDI Model</td>
<td>11:30 am</td>
<td>H</td>
</tr>
<tr>
<td>2</td>
<td>Carbon Footprint</td>
<td>10:30 am</td>
<td>H</td>
</tr>
<tr>
<td>3</td>
<td>Climate Change</td>
<td>11:15 am</td>
<td>S</td>
</tr>
<tr>
<td>4</td>
<td>E-Learning</td>
<td>10:15 am</td>
<td>H</td>
</tr>
<tr>
<td>5</td>
<td>Paper and Pulp</td>
<td>11:05 am</td>
<td>S</td>
</tr>
</tbody>
</table>

**Figure 4.20:** QL and RL accessed by user for Query (Q): Carbon Footprint
4.5 Secured Cognitive Agent based Multi-strategic Intelligent Search System (CAbMsISS) (119)

The gap between the user’s requirements and results retrieved by the search engines can be attributed due to the inability of the search engines to understand the user’s heterogeneous needs, preferences and emotions experienced during the search. This results in overflow of redundant and useless information. Another problem encountered in the search engines is the absence of functionality to inform the user about a previously accessed resource tagged with negative emotions or stated as useless by the user. The problems encountered due to lack of user adaptivity in the search engines as discussed in section 3.6 necessitates the need for a Multi-strategic Intelligent Search System (MsISS) designed to search on the basis of user context and emotions. Though various search engines provide security during the search process, still the security issues needs improvement.

For resolving the above-mentioned issues, this section proposes and discusses a framework to accomplish Multi-strategic Intelligent Search (MsIS) based on user cognitive style, context, timestamp and emotions experienced during the search. The proposed framework is referred as Secured Cognitive Agent based Multi-strategic Intelligent Search System (CAbMsISS) and is presented in (119). Its implementation including the technology and structure of the Database used along with the experimental setup are also exemplified in this section.

The implemented structure of the Secured CAbMsISS is referred as Cognitive Search Engine based on Emotions (COGSEMO). Snapshots of COGSEMO have been used to highlight the working. The statistical tool used for the analysis and comparison is also reported in this section. Finally, decisive conclusion is presented in section 4.6

4.5.1 Design of Secured CAbMsISS (119)

The present work suggests to perform the search based on the user cognitive style, context, timestamp and emotions experienced during the search process. It is referred as Multi-strategic Intelligent Search (MsIS) (119) as shown in Figure 4.21.
User modeling forms the backbone for the present work. Registration form used at the entry level in the system helps to gather the static data of the user. The dynamic data (184) is collected through behavioral analysis, self-reports and image captured through Web Camera (Webcam). Behavioral data includes navigation behavior of the user comprising of queries entered, resources accessed, activities performed on the resources and timestamp. Self-reports of the user are based on emoticons. Webcam is used to capture the image of the user. Image recognition (215) helps to extract the emotions experienced by the user during the search process.

To generate the user model based on cognitive style of the user, context, timestamp and emotions experienced during the search process, the work discussed in (119) uses both the approaches: user guided modeling and dynamic user modeling. The browsing activities of the user helps to recognize the regularities in the search terms used, path followed in accessing the resources and emotions experienced. Clustering the queries and resources based on emotions, timestamp etc. guides the reasoning, judging and decision making process.

The design of the proposed Multi-strategic Intelligent Search (MsIS) System (119) is inspired by BDI model (88) of cognitive agent.
The system uses five-layer architecture of Cognitive Agent (CA) (92) as discussed below:

- **Perceptual Layer** perceives the information from the environment. It includes perceiving the user at the time of entry into the system through Webcam, perceiving the queries entered, resources accessed and emotions experienced during the search process. This layer includes a Judging Block that performs “feature extraction”, “identification” and “estimation”.

- **Reasoning Layer** is the thinking layer of the system as it makes decisions using information received from Perceptual Layer and existing knowledge. It then passes these decisions to Judging Layer.

- **Judging Layer** not only receives the information from Reasoning Layer but also sends back the processed information to Reasoning Layer for enhancing the decisions.

- **Response Layer** receives the information from Judging Layer, applies the rules applicable based on the situation and further instructs Perceptual Layer to issue the response.

- **Learning Layer** is responsible for monitoring the actions performed and deriving new knowledge. It is responsible for updating the existing knowledge based on which Reasoning Layer functions.

Furthermore, CIMM proposed by (118) provides the baseline for the design of a Secured CAbMsISS as shown in Figure 4.21. The framework in Figure 4.21 allows the user to search the Web based on the user CIMM (cognitive style and emotions), context and timestamp.

**4.5.2 Structure of Secured CAbMsISS (119)**

The framework for Secured CAbMsISS consists of three different sub-systems as shown in Figure 4.22. These include:

- **User Entry (UE)** - labeled as 1 in Figure 4.22.
- **User Authenticate (UA)** - labeled as 2 in Figure 4.22.
- **Secured Search (SS)** - labeled as 3 in Figure 4.22. Two-layered security has been integrated in the framework to provide a secured search environment to the user.
The proposed framework first checks the type of the user on entry in the system. In case of new user, the details are stored in the Database and authorization is given to the user for search. For existing user, the details entered are matched with the data stored in the Database. On authentication, the user is permitted access to search the Web. The three sub-systems shown in Figure 4.22 are discussed in the subsequent sections.

4.5.3 User Entry (UE)

The sub-system User Entry (UE) (labeled as 1) in Figure 4.22 initiates the security check for the user through the module Entry. It allows the user to either register in the system as new user or enter the login details for existing user. The logic for the module Entry is discussed in the flowchart shown in Figure 4.23. Before providing the access to search the Web, the module Entry at the outset checks the type of the user (new/existing).

Register (1.1) and Login (1.2) shown in Figure 4.23 are the two sub-modules of the module Entry that are initiated depending on the type of the user.
4.5.3.1 Register (New User)

The module Entry initiates sub-module Register (1.1) (Figure 4.23) to **generate a registration form to be filled by the user**. Filling the registration form is a prerequisite to initiate the Web-based search process using Secured CAbMsISS framework (119). The form consists of several fields that assist the framework in providing adaptive search results to suit the user requirements.

**The registration form includes the following fields:** Name, E-Mail, Date of Birth, Sex, Qualification, Organization the user is associated with, Work Field, Interest and Address. The most important field in the form is the E-Mail of the user as it is used to augment security in the system. The module generates and sends One Time Password (OTP) to the user E-Mail for user authentication at the time of new user registration. If OTP entered by the user matches with OTP generated by the system, then the user is authenticated. This activates the Webcam to capture the user image. Finally, the user details including user image are stored in the Database as shown in Figure 4.23. **Algorithm:** Register describes the logic for the sub-module Register.
Algorithm: Register [New user registration]

1. Display Registration Form consisting of field: Name, E-Mail, Date of Birth, Sex, Qualification, Organization of the user, Work Field, Interest and Address.
2. Generate OTP and send it to User E-Mail.
3. Allow User to enter OTP.
4. If (OTP entered matched), then : /* User is Authenticated */
   a. Activate Webcam to capture User image.
      /* Image stored is used for User Recognition */
   b. Authenticate = True.
   c. Store User details and image in the Database.

Else: /* OTP not matched, User not Authenticated */
   a. Authenticate = False.

[End of If structure]

[End of Algorithm Register]

4.5.3.2 Login (Existing User)

The module Entry initiates sub-module Login (1.2) (Figure 4.23) to allow the user to enter the E-Mail. The E-Mail entered by the user is checked in the Database. If match is found for the E-Mail i.e. the user already exists, then Webcam captures the user image. The image captured is matched with the user image stored in the Database. If the user image matches, then the user is authenticated. In case, user image does not match, then the module generates and sends OTP to the user E-Mail for authentication. On entering the correct OTP, the user is authorized to use the search system.

The existing user is denied access if:

- E-Mail entered by the user does not match with the E-Mail stored in the Database.
  
or
- OTP does not match.
Algorithm: Login describes user authentication logic used in sub-module Login.

Algorithm: Login [Existing user]

1. Enter User E-Mail.
2. /* Match E-Mail with Database */
   
   If (E-Mail matched), then:
   
   a. Activate Webcam to capture User image for User Identification.
      /* Face Recognition */
   b. If (User Image matched / Identified) then: /* User is Authenticated */
      
     Authenticate = True.
   
   Else: /* Image not matched, User is not Authenticated */
      
      i. Authenticate = False.
      ii. Display “User not Identified”.
      iii. Generate OTP and send it to User E-Mail.
      iv. Allow User to enter OTP.
      v. If (OTP entered matched), then: /* User is Authenticated */
         
         Activate Webcam to capture User image.
         /* Image stored is used for User Recognition */
         Authenticate = True.
         Store User image in the Database.
   
   Else: /* OTP not matched, User not Authenticated */
      
      Authenticate = False.
      Goto step iii.
   [End of If structure]

   [End of If structure]

Else: /* E-Mail not matched */

   Authenticate = False.
   Display: “E-Mail not Matched”.
   Goto step 1.
[End of If structure]

[End of Algorithm Login]
4.5.4 User Authenticate (UA)

If the user successfully crosses the security layer of the sub-system UE, then this sub-system authenticates the user and provides the access to search. Otherwise, the system denies access to the user and transfers the control to the sub-system User Entry (UE). The authentication may fail in the following two cases:

- **New user**: OTP does not match. It provides the facility to regenerate the OTP.
- **Existing user**: Either the E-Mail or the user image does not match.

For authenticated user, system transfers control to sub-system Secured Search (SS).

4.5.5 Secured Search (SS)

It provides search interface to the authenticated user to perform Web-based search. The two modules namely MsIS and Security as shown in Figure 4.22 control the functionality of this sub-system.

4.5.5.1 MsIS

It performs Multi-strategic Intelligent Search (MsIS) based on cognitive style of the user, emotions experienced during the search, context of the query and timestamp. CIMM discussed in (118) supplements the present work by enabling the MsIS module to capture the search activities performed by the user, emotions experienced and timestamp.

4.5.5.1.1 CIMM (118)

This section describes the detailed working of the five modules (Search Activity, Capture, Store, Decision Making and Search Suggestions) discussed in the high-level view of CIMM (Figure 4.17).

A. Search Activity (SA)

This module provides a platform to the user for initiating a search task. It allows the user to either enter a new query or select from the list of previously searched queries called Query List (QL) stored in the Database. As shown in Figure 4.24, New and
Existing are the two different types of search activities performed by the user. New refers to the process of entering / typing query (Q) in the text box. If no Q is entered, then the activity type is Existing. In this case, the user can select Q from QL. Corresponding to Q selected, a list of resources called Resource List (RL) previously accessed by the user is displayed. RL is stored in the Database and it highlights whether a Resource (RS) is only read by the user or any other activity (save etc.) is also performed on RS. It also specifies about the usefulness of RS depending on the emotions attached to RS. This facilitates the user to prioritize the order in which different RS’s are to be accessed according to their importance.

In real life many times in a search process some RS is accessed by the user and by advancing more deep in the search tree for RS, it is found to be useless or frustrating. To avoid repeated access to such useless RS, CIMM (118) captures the emotions of the user with reference to Q and RS. This serves as useful criteria in selecting RS from RL. It also saves user time. Based on these two search activities, CIMM processes the user request as shown in Figure 4.24. Here, A and B are the connectors.

![Figure 4.24: Working of module Search Activity (SA) of CIMM](image-url)
Detailed description of the two search activities (New and Existing) is as follows:

1. Activity Type = New (Query (Q) is entered by the user in the text box)
   - **Match Query:** Q entered by the user may or may not exist in QL. This checks whether Q already exists in QL or not.
   - **Match Found:** Q exists (found) in the Database. The user can select Q from QL to continue search from previously accessed RS or can initiate another search for Q using the Search Engine (SE). In both the situations when either the user enters Q or selects Q from QL, the module Capture is initiated as shown in Figure 4.24. It records the query keywords, logstamp and user emotions. Similarly, when some RS is selected from RL, then the module Capture is initiated to record the details of RS accessed, logstamp, activity performed on RS and user emotions.
   - **Select RS from RL:** The user is presented with a predefined list of resources RL previously accessed corresponding to Q. The module Capture is initiated to record the details.
   - **Initiate New Search:** If the user does not wish to access the resource from RL, then the search can be initiated for Q by SE. In this case, the RL displayed depends on SE.
   - **Match Not Found:** In case Q entered by the user does not exist in the Database, then the module Capture is initiated and SE displays the result set i.e. RL is displayed by SE.
   - **New Task:** While searching for Q, if the user initiates another query Q₀, then the above mentioned process is repeated for the new task Q₀.

2. Activity Type = Existing (Query (Q) is not entered by the user)
   - **Select Q from QL:** The user is presented with a predefined list of resources RL previously accessed corresponding to Q and the module Capture is initiated.
   - **Not selected from QL:** In case Q is not selected form QL, then CIMM displays a list of similar queries starting with the root keyword.
   - **Similar Queries:** Here, the module Generate_QL creates a sub-list of similar keyword queries from QL and allows the user to perform search from the new QL obtained.
➢ **Not Similar Queries:** If the user does not wish to select the similar keyword queries, then the system allows the user to either enter new query or initiate new search for the existing query. In this case, RL is displayed using SE.

Figure 4.24 illustrates the steps performed by the user and system to capture the cognitive style, context, emotions and timestamp. Since, according to (12), the emotional outcomes of an online search process can be influenced by different structures of perceptions, initial affective states and emotions felt during the search process, thus CIMM is instructed to capture the emotions at the beginning, during and after the search process.

**B. Capture**

This module is initiated when either a query is entered or an existing query is selected from QL. It is also activated when some RS is selected by the user. The algorithm Capture(Q) captures the information associated with the search process.

The modules and actions used to describe the logic in the algorithm are explained as follows:

1. **ACT{Q} (Action performed on Q)**
   
   If action performed == Click, then: call module Record_keywordlog(Q).

2. **Module Record_keywordlog(Q)**
   
   It stores the details of Q including query keywords, logstamp and user emotions in the Database. It also displays RL for Q.
   
   In case Q is selected from QL, then: predefined RL is displayed otherwise SE generates a new RL for Q.

3. **ACR{RL} (Action performed on RL)**
   
   If action performed == Click, then a Resource RS is selected from RL and the module Record_searchlog(RS) is initiated.

4. **Module Record_searchlog(RS)**
   
   It captures and stores the details of the resource RS accessed, logstamp, activity performed on resource and user emotions.
5. **ACT{W.RS[Current]}** (Action performed on window W of current resource RS)
   
   If action performed == Close, then the user has closed the current resource RS. Decrease the existing number of opened resources by 1. The Module Record_searchlog(RS[Current]) now stores the current status of the closed resource RS[Current] and the window for RS[Current] is closed.

6. **Module Check_window(i, current)**
   
   Checks the current resource window that is presently activated by the user.

7. **ACT{W.RL} (Action performed on the window of RL)**
   
   If action performed == Close, then the user does not wish to select an existing resource.
   
   For a query Q, If (ACT{RL} != Click and ACT{W.RL} == Close ), then no action is performed by the user for Q. No resource is selected for Q from RL. In this case, it allows the user to again start the search process for some task (same query using SE or new query).

**Algorithm: Module Check_window(i, current)**

It checks the current window activated by the user.

Repeat for (k = 1 ,2 ,……, i)

[i is the total no. of resource windows opened by the user at any time]

{ 
   If  ACT{W.RS[k]} == Click  
   {  
      current = k.  
   }  
   [End of If structure]  
   break.  
}

/* End of Algorithm Module Check_window */

The logic used to describe working of the module Capture is illustrated using **Algorithm: Capture(Q)**. Here, resource R is synonymous to RS in the algorithm.
Algorithm: Capture(Q)

It captures the information pertaining to the search process initiated by the user. There are two types of activities: New (user enters Q) and Existing (user selects Q from QL).

1. U: Repeat while ACT(Q) = Click [Q is entered or selected from QL]
   
   a) Call Record_keywordlog(Q) [store details of Q in Database]
   
   b) Display RL [RL is predefined for existing Q][RL depends on SE for new Q]
   
   c) i = 0, j = 0 [temporary variables, i is global and j is local]
   
   d) V: Repeat while ACT(RL) = Click [resource R is selected from RL]
      
      i. j = 1 [check if any R is selected for Q or not]
      
      ii. Copy the resource selected in R[++]
          [i is first incremented and then used]
      
      iii. R[i].activity = read [if R is selected then user wants to read R]
      
      iv. current = i [current is global variable that specifies the current resource
          used by the user among the multiple R opened by user]
      
      v. Call Record_searchlog(R[current])
         [store details of R[current] in Database]
      
      vi. Call Check_window(i, current) [check current window used by user]
      
      vii. Repeat while ACT(W.R[current]) != Close
           
           If (R[current].activity != read), then: [user wants to save / print R]
           
           Call Record_searchlog(R[current]) [store details of new activity]
R.activity[current] = read
}

If (ACT{RL} == Click), then: Goto V

Call Check_window(i, current)
} /* end of step vii loop */

viii. R.activity[current] = close, i = i-1

[if any resource R is closed by user then, number of resources currently opened are decreased by 1]

ix. Call Record_searchlog(R[current])

[store details of the current R which is closed by the user]

x. Close W.R[current]

xi. Call Check_window(i, current)

} /* end of step d loop */
e) If (j == 0) and (ACT{W.RL} == Close, then:

[no R is selected for Q and user presses the close button for RL]

{ 
Call Record_keywordlog(Q) [store details of Q in Database]

}
} /* end of step 1 loop */

2. If new task is initiated, then: [either Q is issued or existing Q is selected]

Goto U

Else

Stop

[End of If structure]

/* End of Algorithm Capture */
C. Store

It stores the data gathered by the module Capture in the Database. The Conceptual ER Model of the Database as shown in Figure 4.25 describes the entities and relationships that exist between these entities. The Conceptual ER Model of the Database consists of five entities (webuser, webuserphoto, keywordlog, searchlog and emoticon) related to one another using the relationship “has”.

Figure 4.26 illustrates the detailed view (Logical ER Model) of the Database. It describes the entities in detail. For every entity, it describes the attributes, Primary Key and Foreign Key. The symbols used in ER Model are discussed in Table 4.4.

<table>
<thead>
<tr>
<th>Table 4.4: Symbols used in ER Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Entity</td>
</tr>
<tr>
<td>Relation</td>
</tr>
<tr>
<td>1 : M (1 : Many)</td>
</tr>
<tr>
<td>Attribute</td>
</tr>
</tbody>
</table>
Figure 4.25: Conceptual ER Model Diagram of the Database

Figure 4.26: Logical ER Model Diagram of the Database
Figures 4.27(a) to 4.27(e) illustrate the physical structure of the Database.

**Figure 4.27(a): Structure of Table webuser**

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>bigint(20)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>2</td>
<td>email</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>3</td>
<td>firstname</td>
<td>varchar(30)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>4</td>
<td>lastname</td>
<td>varchar(30)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>5</td>
<td>dob</td>
<td>date</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>6</td>
<td>sex</td>
<td>varchar(1)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>7</td>
<td>qualification</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>8</td>
<td>organization</td>
<td>varchar(100)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>9</td>
<td>workfield</td>
<td>varchar(100)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>10</td>
<td>interestsarea</td>
<td>varchar(100)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>11</td>
<td>address</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>12</td>
<td>city</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>13</td>
<td>country</td>
<td>varchar(50)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>14</td>
<td>status</td>
<td>tinyint(4)</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
</tbody>
</table>

**Figure 4.27(b): Structure of Table webuserphotos**

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>bigint(20)</td>
<td></td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>2</td>
<td>userid</td>
<td>bigint(20)</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
<tr>
<td>3</td>
<td>image</td>
<td>varchar(255)</td>
<td>latin1_swedish_ci</td>
<td>No</td>
<td>None</td>
<td></td>
<td></td>
<td>Change</td>
</tr>
</tbody>
</table>
Figure 4.27(c): Structure of Table keywordlog

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>2</td>
<td>keywords</td>
<td>varchar(1000)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>3</td>
<td>userid</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>4</td>
<td>logstamp</td>
<td>datetime</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>5</td>
<td>emoticon</td>
<td>smallint(6)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
</tbody>
</table>

Figure 4.27(d): Structure of Table searchlog

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Collation</th>
<th>Attributes</th>
<th>Null</th>
<th>Default</th>
<th>Extra</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>id</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td>AUTO_INCREMENT</td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>2</td>
<td>link</td>
<td>varchar(2000)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>3</td>
<td>linktext</td>
<td>varchar(256)</td>
<td>latin1_swedish_ci</td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>4</td>
<td>activity</td>
<td>tinyint(4)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>5</td>
<td>userid</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>6</td>
<td>keyid</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>7</td>
<td>logstamp</td>
<td>datetime</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
<tr>
<td>8</td>
<td>emoticon</td>
<td>smallint(6)</td>
<td></td>
<td></td>
<td>No</td>
<td>None</td>
<td></td>
<td>Change ▶️ Drop ✖️ Primary</td>
</tr>
</tbody>
</table>
D. Decision Making

The Cognitive Mapping for the user search process is based on BDI model as discussed in CIMM (118). On the basis of the data stored in the Database and Cognitive Mapping, this module analyses the data from the perspective of query, resource, user emotions and timestamp.

E. Search Suggestions

The data analyzed by the module Decision Making, helps to provide the following suggestions to the user regarding the Query (Q) and Resource (RS).

1. Query (Q)

- Whether Q has been previously searched or not?
- Number of times Q has been previously searched.
- Queries starting with the root keyword of Q.
- Timestamp when Q was repeatedly issued. This helps to find whether the user has fixed some specific time to search Q.
- Emotions of the user while issuing Q.
- Identify queries associated with positive and negative emotions experienced during the search process.
- Whether any resource was accessed corresponding to Q or not?
2. Resource (RS)

- Different activities performed on RS.
- Time spent on RS and emotions experienced for RS.
- Identify useful and useless resources. Identify resources tagged with positive and negative emotions.

4.5.5.1.2 Context

The use of different fields like Organization, Work Field, Interest Area in the user registration process helps the system to provide the user’s context for the query.

4.5.5.1.3 Timestamp

The decision of the system to execute the query at a specific timestamp is executed using the five layered cognitive agent architecture (92).

4.5.5.2 Security

It is implemented into the system using the five layers (92) defined in the structure of the cognitive agent.

- Perceptual Layer of the proposed system perceives the user image through the Webcam.
- Reasoning Layer compares the user image received from the perceptual layer with the existing knowledge i.e. user image stored in the Database.
- Judging Layer receives the information from the reasoning layer i.e. whether the user image matches or not.
- If the image matches, then the Response Layer generates the response regarding the successful match and instructs the Perceptual Layer to send the response regarding the authentication to the user.
- Learning Layer updates the existing knowledge by storing the matched user image in the Database.
4.5.6 Experimental Setup

The proposed framework Secured CAbMsISS has been designed and implemented using Web technologies (HTML 5 + JADE, PHP, Java Script, Jquery) and My SQL. It has been executed on a Windows 7, 64 bit system environment with an Intel Core i5–2430M CPU@ 2.40 GHz with 4 GB RAM and 500 GB hard disk. The implementation of the framework Secured CAbMsISS shown in Figure 4.28 is referred as Cognitive Search Engine based on Emotions (COGSEMO) (119). The structural design and working of COGSEMO (119) uses the following technologies:

- Page Layout - HTML 5 + JADE (Template Engine)
- Client Interactivity - Java Script, AJAX and Jquery
- Database - Structured Query Language (My SQL)
- Server Site Processing and Database Handling - Hypertext Preprocessor (PHP)

4.5.7 Tools Used

This section discusses in detail the tool COGSEMO (119) used for gathering the user search data. The data analysis tool is also discussed in this section.

4.5.7.1 COGSEMO (Implementation of Secured CAbMsISS Framework) (119)

COGSEMO (119) provides a secured and adaptive search environment to the user for the Web-based search. It builds the user model based on the cognitive style, context, timestamp and emotions experienced by the user during the search process. Figure 4.28(a) presents the entry screen for COGSEMO. The secured system provides access to the user for Web-based search only after authenticating the user. The security checks performed at the entry point in the system are discussed below.

4.5.7.1.1 User Entry (COGSEMO provides security checks for the user)

A. New User Registration

Step 1: Button: New User Register Here

New user can register in the system by clicking on the Button: New User Register Here as shown in Figure 4.28(a). Subsequently, the system displays New User Registration Form (Figure 4.28(b)) consisting of 12 different fields.
Figure 4.28(a): COGSEMO

(Implementation of Secured CAbMsISS Framework)

Figure 4.28(b): New User Registration Form
Step 2: Registration Form

The fields: Organization, Work Field and Interest Area help COGSEMO to assist in providing adaptive search results to suit the user requirements. This facilitates in mapping the user context for the search. It is mandatory to fill all the fields in the registration form.

The field: E-Mail provides security to the system. After completely filling the registration form, press the Button: Register Me shown in Figure 4.28(b). This activates the system to check whether the E-mail entered by the user already exists in the Database or not. In case the E-mail entered by the new user already exists in the Database, then the system instructs the user to enter another E-mail. This field helps to uniquely identify the user and acts as Primary Key for the entity: webuser as shown in Logical ER Model of the Database (Figure 4.26).

Step 3: One Time Password (OTP)

If the E-mail entered by the user during the registration process is unique, then COGSEMO generates an OTP (Figure 4.28(c)).

![Figure 4.28(c): User Authentication through One Time Password (OTP)](image)

The OTP generated by the system is sent to the E-mail specified by the user during the registration process (Figure 4.28(d)).
Enter the OTP sent through the E-mail in the **Text Box: One Time Password** and click on the **Button: Proceed** (Figure 4.28(e)). This feature initiates the security check to allow only the authenticated user to access the system.
If the OTP entered by the user matches with the OTP generated by the system, then COGSEMO activates the Webcam (if available with the user computer system). As illustrated in Figure 4.28(f), COGSEMO instructs the user to look straight into the camera to enable capturing the user image.

![Figure 4.28(f): Capturing user image through Webcam](image)

After capturing the user image, the data entered in the registration form and the user image captured are stored in the Database. The image recognition feature (215) adds another level of security. On completing the registration process, the authenticated user is provided access to the COGSEMO search system to perform the Web-based search. Image recognition and OTP adds security and thus, prevents unauthorized access to the user’s search log. After successful login, the user can either search for a new query by typing the keywords in the text box or search from the previously searched queries available in the search log shown in Figure 4.28(g).
B. Existing User Login

**Step 1: Enter Login details**

The existing user is granted access to the system by entering the Login details. This provides the first security check for the existing user. As the user enters the E-Mail in the text box and clicks on the **Button: Sign in** shown in Figure 4.28(a), COGSEMO verifies the E-Mail entered by the user.

**Step 2: Capture user image**

If the E-Mail entered matches, then Webcam captures the user image and performs second security check by matching the user image. On successful match of the user image, access to use COGSEMO is granted. In case, the E-Mail does not match, then the system does not authenticate the user and denies access to the user.

**Step 3: OTP for user authentication**

If Webcam is not attached to the system or if the user image does not match, then COGSEMO generates an OTP (Figure 4.28(h)). Enter OTP for user authentication.
4.5.7.1.2 Searching with COGSEMO

The search problem is defined by the set of keywords (query), user characteristics (defined through the fields in registration form), cognitive style (defined by search style i.e. if in majority, search is from specific domains etc.) and user emotions.

A. User Context

COGSEMO facilitates the user search process by presenting a list box with certain options (fields entered during user registration). These options include: Specified Web Domain, My Organization, My Work Field and My Interest Area as shown in Figure 4.28(g). The values for the fields: My Organization, My Work Field and My Interest Area are entered by the user during the registration process. These fields help to specify the context of the user for the query. This improves the relevancy of the retrieved results and also helps the user to perform the search based on the user context. It also permits the user to perform search from a specific domain. The system maintains the list of domains searched by the user for future reference. Figure 4.28(i) illustrates the option provided for Domain specific search.
Example: Specified Web Domain - Suppose the user enters Domain = ieee.org. The results obtained by COGSEMO for the query = Education searched using Domain = ieee.org are presented in Figure 4.28(j).

Figure 4.28(j): Results for keyword = Education searched with Domain ieee.org

Example: My Organization – Suppose the field: Organization entered in the registration form = Red Cross Haryana and query = secondary carbon footprint. The user’s context is to obtain results corresponding to any work done by Red Cross Haryana in reference to secondary carbon footprint. On selecting the option: My Organization for the specified query, COGSEMO converts the query (Q) to Semantic Query (SQ) and sends SQ to the PSE to retrieve relevant results. Google search engine referred as PSE has been used in (119) to retrieve the search results. Figure 4.28(k) highlights the results obtained after executing the query with user context: My Organization. This eliminates the need to repeatedly specify the user context in the query using the keywords. Once specified, the context can be used repeatedly. COGSEMO also provides the facility to update the fields entered in the registration form by clicking on the E-Mail shown in Figure 4.28(l).
Figure 4.28(k): Results obtained for user context: My Organization

B. Emotions

COGSEMO captures the user image through Webcam as shown in Figure 4.28(l) and analyzes it to extract the emotions (215).

It either automatically extracts the emotions using Webcam (215) or allows the user to explicitly specify the emotions by selecting an option from the list of emoticons shown in Figure 4.28(m). After selecting the emoticon, the corresponding user emotion is tagged (associated) with the query entered or resource accessed. This provides assistance to the user in restricting the access to a resource or query previously tagged with negative emotions or found useless.
Figure 4.28(m): Emoticons presented by COGSEMO

COGSEMO supports the user by listing the queries entered, resources accessed and emotions experienced through a task pane called SEARCH LOG (Figure 4.28(l)). **Option: Search Links** enumerates the resources accessed, time of search and emotions (Figure 4.28(l)). **Option: Keywords** lists the queries searched. It also categorically arranges keywords and resources according to type of emotion tagged with them. **Figure 4.28(n) lists the queries (keywords) for user emotion = Smile.**

**Figure 4.28(n): Queries tagged with user emotion = Smile**

Figure 4.28(o) lists the queries associated with the **user emotion = Frustrated**. This prevents the user from accessing a resource previously tagged with negative emotions. Previous search can be executed repeatedly using the **Button: New Search** shown in Figure 4.28(n) and 4.28(o). COGSEMO displays the queries initiated and corresponding resources accessed as illustrated in Figure 4.28(p). On executing the **Button: New Search, a message is displayed** regarding the emotion experienced majority of time for the query and resource. Emotion Statistics are presented in Figure 4.28(q). The symbols + and − depicts the % age of number of time positive and negative emotions were experienced.
Figure 4.28(o): Queries tagged with user emotion = Frustration

Figure 4.28(p): Query executed using Button: New Search (Query no. 42)

Figure 4.28(q): Emotion Statistics for queries executed
C. Timestamp

The system also lists the queries and resources accessed corresponding to different timestamps (Morning, Afternoon, Evening and Night) interpreted by COGSEMO (Figure 4.28(r)). It also shows the % age of number of time a query has been accessed in a timestamp. Query no. 17 (hair color) shown in Figure 4.28(r) has been accessed 2.35% in Morning and 97.65% at Night. All the queries executed for a timestamp are shown in Figure 4.28(s). It also illustrates the queries accessed for timestamp = Night and % age of number of time it has been accessed is illustrated.

4.5.7.2 Sampling

The retrieval effectiveness of the Secured CAbMsISS system (implemented as COGSEMO) (119) has been tested during the study. Based on Judgmental Sampling, ten queries (N = 10) were selected for the experimental study.

4.5.7.3 Query Execution

All the 10 queries were executed using PSE and CAbMsISS. The QRT (in min:sec.millisecond) was recorded. Each query was re-executed for both approaches and mean value of QRT observations obtained per query for each approach was calculated and analyzed. Google search engine was used to retrieve the results. The search results obtained and QRT was recorded. The experimental data obtained is presented in chapter 5. The results were analyzed to find Proportion of Retrieved Documents Judged Relevant (RDJR) i.e. Precision and Proportion of Retrieved Documents Judged Irrelevant (RDJI). Top 10 documents retrieved were considered.
4.5.7.4 Statistical Tool Used for Data Analysis

Descriptive Statistics: Mean was used to calculate the mean values for the parameters. Inferential Statistics: Independent t-test was used to measure the significance of difference between the various retrieval effectiveness measures (QRT, Precision, RDJI) for PSE and CAbMsISS. The comparative analysis of the results retrieved from PSE and CAbMsISS is presented in section 5.4 of chapter 5.

4.5.7.5 Parameters Used for Experiment

The summarization and comparative analysis of the query retrieval results obtained from PSE and CAbMsISS has been based on: QRT, Precision (P) and RDJI. The graphical illustration of the comparative view of these parameters for PSE and CAbMsISS is depicted using charts prepared in MS-Excel (discussed in chapter 5).

4.5.7.6 Hypotheses

Table 4.5 enumerates the three hypotheses formulated using the parameters described in section 4.5.7.5. Null Hypothesis was formulated for each parameter. The results derived from the statistical test are exemplified in chapter 5 (section 5.6).

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Hypothesis</th>
<th>Test Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$H_0(1)$: There is no significant difference in the Mean QRT for PSE and CAbMsISS. $H_0(1)$: QRT (PSE) = QRT (CAbMsISS)</td>
<td>Independent t-test</td>
</tr>
<tr>
<td>2</td>
<td>$H_0(2)$: There is no significant difference in the Proportion of RDJR for PSE and CAbMsISS. $H_0(2)$: Proportion of RDJR (PSE) = Proportion of RDJR (CAbMsISS) or $H_0(2)$: There is no significant difference in the Precision for PSE and CAbMsISS. $H_0(2)$: Precision (PSE) = Precision (CAbMsISS)</td>
<td>Independent t-test</td>
</tr>
<tr>
<td>3</td>
<td>$H_0(3)$: There is no significant difference in the Proportion of RDJI for PSE and CAbMsISS. $H_0(3)$: Proportion of RDJI (PSE) = Proportion of RDJI (CAbMsISS)</td>
<td>Independent t-test</td>
</tr>
</tbody>
</table>

4.6 Conclusion

Noise Removal Framework, Noise Removal for Semantic Information Processing (NRSIP) framework, Cognitive Information Mapping Model (CIMM) and framework for Secured Cognitive Agent based Multi-strategic Intelligent Search System (CAbMsISS) have been implemented in this chapter to address the challenges highlighted in chapter 3.