KLE Technological University | Creating Value, Leveraging Knowledge

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A Mini Project report on

Search Engine Anatomy with Space Relevance

Submitted

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

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Name	**USN**
Jiya S Palrecha	01FE21BCS094
Abhijna R Kalbhag	01FE21BCS107
Karthik R Khatavkar	01FE21BCS196
Avaneesh S Lad	01FE21BCS200

Under the guidance of

Prof. Prakash Hegade

School of Computer Science and Engineering

KLE Technological University, Hubballi

2023-2024

REF: 2024/P-01

Knit Space
Software Research and Services Private Limited,
Hubballi - 31.
Registration Number: 160767
www.knitarena.com

Project Completion Letter

This letter is to certify that the project titled "Search Engine Anatomy with Space Relevance" from Knit Space was successfully completed by the student team mentioned below from KLE Technological University as a part of V semester three credit mini project. The project work was carried out under the guidance of Mr. Prakash Hegade. The team performance was graded excellent and has positively contributed towards the industry segment growth.

SI. No.	SRN	Team Member Name
1.	01FE21BCS094	Jiya Santosh Palrecha
2.	01FE21BCS107	Abhijna Ravindra Kalbhag
3.	01FE21BCS196	Karthik R Khatavkar
4.	01FE21BCS200	Avaneesh S Lad

Regards,

Vishwanath T
Tech Lead, Knit Space.
31 Jan 2024

KLE Technological University | Creating Value, Leveraging Knowledge

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the project entitled "Contextual Search Anatomy" is a bona fide work carried out by the student team: Mr.Karthik R Khatavkar - 01FE21BCS196, Ms.Jiya S Palrecha - 01FE21BCS094, Mr.Avaneesh S Lad - 01FE21BCS200, Ms.Abhijna R Kalbhag - 01FE21BCS107, in partial fulfillment of the completion of the 5th semester B. E. course during the year 2023 – 2024. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Guide Name
(Prakash Hegade)

SoCSE,Head
Dr. Vijaylakshmi M.

External Viva-Voce

Name of the examiners Signature with date

1 _____ _____

2 _____ _____

ABSTRACT

Traditional search engines have relied on specific keywords as the main source for information retrieval. However, these systems often struggle to understand what users truly intend, leading to challenges in delivering precise information. In response, our contextual search system marks a significant departure from this norm.Our research introduces a transformative contextual search system, aiming to redefine how we search for information. It starts with a single keyword user query.When a user initiates a search, our system retrieves documents from the Wikipedia database that are specific to the entered keyword. The keywords extracted from these documents undergo further refinement, including the removal of unnecessary words. The resulting set of the 15 most repeated keywords becomes the basis for building a complete graph using 'networkx.' This graph generates structured key-value data for all nodes. Subsequent pruning of edges, based on similarity values, refines this into the knowledge graph.Our approach goes beyond graph construction. We apply BFS and DFS traversals for cluster analysis, revealing hierarchical structures and identifying central nodes crucial for information flow. Centrality analysis, involving Betweenness and Closeness Centrality, helps us understand key nodes in the network.The exploration doesn't stop with algorithms; it extends to interactive exploration, allowing users to visually engage with the knowledge graph. This facilitates generating suggestions and discovering related concepts, enhancing the overall user experience.The final stage involves mapping the knowledge graph to predefined contextual spaces – Pratyaksha, Anumana, and Upamana – along with qualities like Rajas, Tamas, and Sattva. These contextual spaces encompass direct knowledge, reasoning or inference, and analogies, respectively, offering users a diverse and dynamic information retrieval experience.In essence, our contextual search anatomy breaks free from traditional keyword-based approaches, ushering in a new era of dynamic, context-aware information retrieval that seamlessly aligns with user intent and preferences.

Keywords: *contextual search, keyword extraction, knowledge graph, query processing, semantic relationship mapping, wikipedia*

ACKNOWLEDGEMENT

We have invested significant efforts in this project. However, its successful completion would not have been possible without the generous support and assistance of numerous individuals and the university. We extend our sincere thanks to all of them. We are profoundly indebted to Prof. Prakash Hegade for his invaluable guidance, constant supervision, and the essential information he provided throughout the project. His support was instrumental in bringing this project to fruition.Our heartfelt gratitude goes to the Mini Project coordinators, Prof. Prashant Narayankar and Prof. Guruprasad Konnurmath, for their kind assistance, cooperation, and encouragement. Their support played a crucial role in the successful completion of our project.Special thanks are also to our parents and fellow mates, whose contributions and insights greatly aided in refining our model at every stage.Additionally, we express our gratitude to Dr. Vijaylakshmi M, the head of the school, for her support and encouragement throughout this endeavor.

Jiya S Palrecha - 01FE21BCS094
Abhijna R Kalbhag - 01FE21BCS107
Avaneesh S Lad - 01FE21BCS200
Karthik R Khatavkar - 01FE21BCS196

CONTENTS

ABSTRACT i

ACKNOWLEDGEMENT i

CONTENTS iii

LIST OF FIGURES iv

1 INTRODUCTION 1

1.1 Preamble 1

1.2 Motivation 1

1.3 Objectives of the project 2

1.4 Literature Survey 2

1.5 Problem Definition 5

2 SOFTWARE REQUIREMENT SPECIFICATION 6

2.1 Overview of SRS 6

2.2 Requirement Specifications 6

2.2.1 Functional Requirements 7

2.2.2 Use case diagrams 7

2.3 Software and Hardware requirement specifications 11

3 PROPOSED SYSTEM 13

3.1 Description of Proposed System. 13

3.2 Description of Target Users 15

3.3 Advantages of Proposed System 15

3.4 Scope of proposed system 15

4 SYSTEM DESIGN 16

4.1 Architecture of the system 16

4.2 Activity Diagram 17

5 IMPLEMENTATION 18

5.1 Proposed Methodology 18

5.1.1 Generating Keywords and Complete Graph: 18

5.1.2 Generating Keyword-Specific Multi-Domain Data: 19

5.1.3 Building the Knowledge Graph (Similarity Measures): 20
5.1.4 Applying Graph Traversals and Graph Algorithms: 21
5.1.5 Contextual Space Mapping: 22

6 TESTING **28**
6.1 Usability Testing 28
6.2 Performance Testing 28
6.3 Reliability Testing 29

7 CONCLUSIONS AND FUTURE SCOPE **30**
7.1 Conclusion 30
7.2 Future Scope 30

REFERENCES **31**

Appendix A **32**
A.1 Context 32
A.2 Gantt Chart 32
A.3 Description of Tools and Technology used 33
A.3.1 Draw.io 33
A.3.2 VS Code 33
A.3.3 Git/Github 33

LIST OF FIGURES

2.1 Use Case for user enters a query 7
2.2 Use case for construction of graph 8
2.3 Use case for processing the query 9
2.4 Use case for mapping the results 10

3.1 Diagram of Proposed System 13

4.1 Pipe and Filter Architecture for proposed system 16
4.2 Activity Diagram for the proposed system 17

5.1 Complete Graph 19
5.2 Top Keywords extracted 19
5.3 Keyword specific multi-domain structured data 20
5.4 Structured data with infobox 20
5.5 Knowledge Graph 21
5.6 Comaprision of similarity measures used 21
5.7 (a)BFS traversal and clusters (b) Closeness Centarlity and Betweenness Centrality values 22
5.8 Edges in knowledge graph 22
5.9 All pairs Shortest path 23
5.10 Contextual Space I 23
5.11 (a)Pratyaksha: Our results (b)Pratyaksha: OpenAI results 24
5.12 (a)Anumana: Our results (b)Anumana: OpenAI results 24
5.13 Contextual Space II 25
5.14 Contextual Space II: Pradhana(Unstructured data) 25
5.15 Contextual Space II: Prakriti(structured data) 26
5.16 Contextual Space III 26
5.17 Contextual Space III: Sattva(Accurate, needed information) 27

A.1 Project management 32

Chapter 1

INTRODUCTION

1.1 Preamble

The Contextual Search Anatomy represents a transformative leap in information retrieval. With a three-module approach, the project aims to refine search outcomes by understanding user queries more deeply, mapping semantic relationships intelligently, and constructing a knowledge graph. By going beyond traditional keyword matching, the system strives to provide users with not just relevant results but also a more personalized and deeply understanding of their queries. This initiative aligns with the broader vision of shaping a more intelligent and context-aware internet, promising a excellent search experience for users. The goal is to provide contextually correct results and personalized outcomes. This project is one such step towards that goal, focusing on extracting context for a given dataset and delivering relevant contextual results.Later mapping the knowledge graph to predefined contextual spaces – Pratyaksha, Anumana and Upamana[2] – along with qualities like Rajas, Tamas and Sattva[1]5.

1.2 Motivation

Motivated by the visionary principles underlying Google's semantic search engine, the Contextual Search Space project aspires to transcend conventional search limitations. Drawing inspiration from Google's commitment to understanding user intent and delivering contextually relevant results, our project seeks to redefine the search experience. We aim to break free from the constraints of keyword-centric approaches, envisioning a future where search engines comprehend user queries deeply, map semantic relationships intelligently, and construct knowledge graphs for a more nuanced understanding of information.

Google's semantic search engine has demonstrated the power of context-aware searches, inspiring our project to strive for a similar level of sophistication. By aligning with this motivation, we intend to enhance the precision and personalization of search results, ultimately contributing to the evolution of a more intelligent and user-centric search paradigm. The goal is to create a contextual search engine that not only meets but exceeds user expectations,

offering a more intuitive, efficient, and satisfying search experience.

1.3 Objectives of the project

- To construct a knowledge base using semantic relationships.
- To crawl and extract keyword-specific multidomain data.
- To realize/map the results with respect to query-specific contextual spaces.

1.4 Literature Survey

1. Query Suggestion and Data Fusion in Contextual Disambiguation -
    - Authors
        - Milad Shokouhi (Microsoft), Marc Sloan (University College London), Paul N. Bennett (Microsoft), Kevyn Collins-Thompson (University of Michigan), Siranush Sarkizova (Harvard University)
    - The paper explores the use of contextual query suggestions and data fusion techniques to improve the effectiveness of search queries. It develops a framework for mining context-sensitive query reformulations and evaluates the performance of these suggestions against a baseline, showing significant improvements in query quality. The authors also address the issue of injecting new relevant documents into search results based on user context, developing a classifier that determines when to inject new search results. This context-sensitive result fusion approach improves retrieval quality for ambiguous queries.
    - Future works suggested:
        - Further exploration of context-sensitive query reformulations and evaluation against different baselines to improve the quality of query suggestions.
        - Investigation of additional features and techniques for the classifier that determines when to inject new search results, aiming to enhance the ranking of injected documents in search results.
2. The Anatomy of a Large-Scale Hypertextual Web Search Engine
    - Authors:
        - Sergey Brin and Lawrence Page (Stanford University)

- The paper presents Google, a large-scale search engine that efficiently crawls and indexes the web, providing improved search results by utilizing the structure of hypertext. It addresses the challenges of scaling traditional search techniques to handle large amounts of data and explores how to leverage the additional information in hypertext to produce better search results. Future works suggested:
  - Support for user context, such as the user's location, and result summarization.
  - Extension of the use of link structure and link text.
  - Ongoing investigations into solving the problem of sub-optimal search results and improving response time.
  - Experimentation with sorting hits according to PageRank to improve search results.
  - Leveraging the vast amount of usage data available from modern web systems for interesting research purposes.

3. Situational Context for Ranking in Personal Search

   - Authors:
     - Hamed Zamani (University of Massachusetts Amherst), Michael Bendersky, Xuanhui Wang, Mingyang Zhang (Google Inc.)
   - The paper focuses on developing context-aware ranking models based on neural networks to understand the relationship between user behavior and situational context features in search logs. Two supervised models are proposed: one that considers a deep context representation learned from all contextual features, and another that incorporates binarized contextual features along with their deep dense abstractions.
   - Future works suggested:
     - Exploration of other types of context, such as short- and long-term search history, user profiles, and user devices, in personal search scenarios.
     - Study of pairwise and listwise context-aware networks to extend the pointwise approach presented in the paper.
     - Investigation of situational features in other search scenarios, such as web search.

4. Placing Search in Context: The Concept Revisited

   - Authors:
     - Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, Eytan Ruppin (Zapper Technologies Inc.)

- The paper introduces a new paradigm for search called IntelliZap, where search is initiated from a text query marked by the user in a document and guided by the surrounding text ("the context"). The use of context in search improves the match to the user's current needs and provides an advanced search tool on the web, even for inexperienced users.
- Future works suggested:
  - Focus on expanding augmented queries in a disambiguated manner.
  - Determination of the most relevant extent of context for processing specific queries.
  - Tailoring the approach to maximize the context-guided capabilities of individual search engines.
  - Further research in utilizing context to focus search and counteract the overwhelming amount of information on the web.

5. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

- Authors:
  - Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language)
- BERT is a language representation model that pretrains deep bidirectional representations from unlabeled text by conditioning on both left and right context in all layers. It can be fine-tuned with just one additional output layer to achieve state-of-the-art results on various natural language processing tasks, including question answering and language inference.
- Future tasks suggested:
  - Generalizing the findings of BERT to deep bidirectional architectures for various language understanding tasks.
  - Exploration of multitask fine-tuning to potentially push the performance of BERT even further by training on multiple tasks simultaneously.

1.5 Problem Definition

To design and implement a contextual search system where for entered user queries, the Query Processing module extracts keywords from user input to understand intent, creating a structured representation. Semantic Relationship Mapping constructs a graph to capture relationships between keywords, with a sub-process pruning unnecessary edges for efficiency. The final module integrates a knowledge graph, enhancing contextual understanding by mapping semantic relationships onto broader informational spaces and ultimately aligning the results with one of the contextual spaces.

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Overview of SRS

A system requirement specification (SRS) is a document which has a detailed description of features of a system or software application. This has the elements which define the required functionalities required by the customer. This includes the details of the technical team who build the system. The purpose is to collect and analyse the ideas that define the system. It gives the complete overview of the project which will be used by everyone and fulfill the customer. It also describes the non-functional requirements, constraints and factors that are necessary for understanding the requirements of the software.

2.2 Requirement Specifications

A system requirement specification (SRS) is a document which has a detailed description of features of a system or software application. This has the elements which define the required functionalities required by the customer. This includes the details of the technical team who build the system. The purpose is to collect and analyse the ideas that define the system. It gives the complete overview of the project which will be used by everyone and fulfil the customer. It also describes the non functional requirements, constraints and factors that are necessary for understanding the requirements of the software.

The goals of the SRS are :

- To provide an overview of the project to anyone who needs to understand the system created.
- It helps both users and any other developers who are working on it and hence simplify the problem.
- To provide an overall insite of the problem domain, users, scope and requirements.
- To provide the requirements specification (FR and NFR).

2.2.1 Functional Requirements

- The system should identify and extract keywords from the entered search queries.
- The system should construct a complete graph, map the relationships, and remove unwanted edges.
- The system should search in the knowledge graph and return relevant results based on the entered query.
- The system should map the search results to one of the contextual spaces.

2.2.2 Use case diagrams

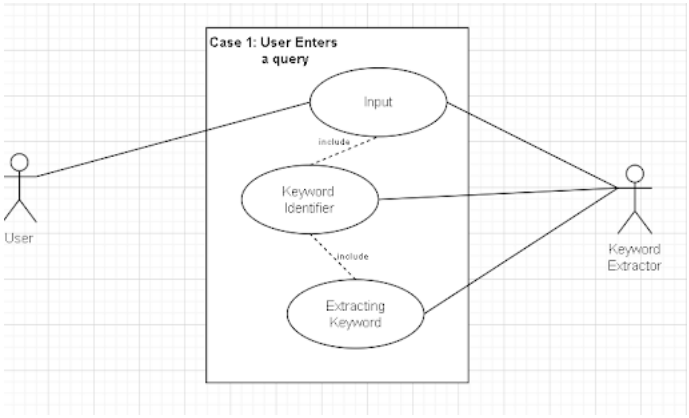

Figure 2.1: Use Case for user enters a query

Actors: User, Keyword Extractor.
Pre-condition:

- Queries are entered.

Post-condition:

- Keywords are extracted.

Main Scenario:

1. User enters a query.
2. System identifies and extracts the keywords.
3. The keywords extracted are further provided for knowledge base generation.

Exception Scenario:

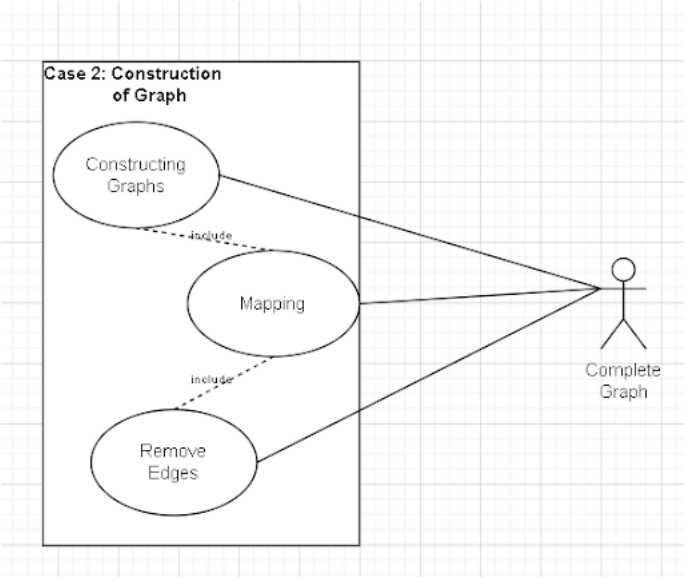

Figure 2.2: Use case for construction of graph

1. User does not enter a query.
2. Users are requested to enter keywords.

Actors: Complete Graph
Pre-condition:

- Keywords are extracted.

Post-condition:

- A complete graph is generated.

Main Scenario:

1. Extracted keywords are used in constructing graphs.
2. Relation between keywords is mapped.
3. Unwanted edges in the graph are removed.

Exception Scenario:

1. Extracted keywords aren't sufficient for graph formation.
2. Edge Removal must not delete the generated graph.
3. Mapping between edges is incorrect.

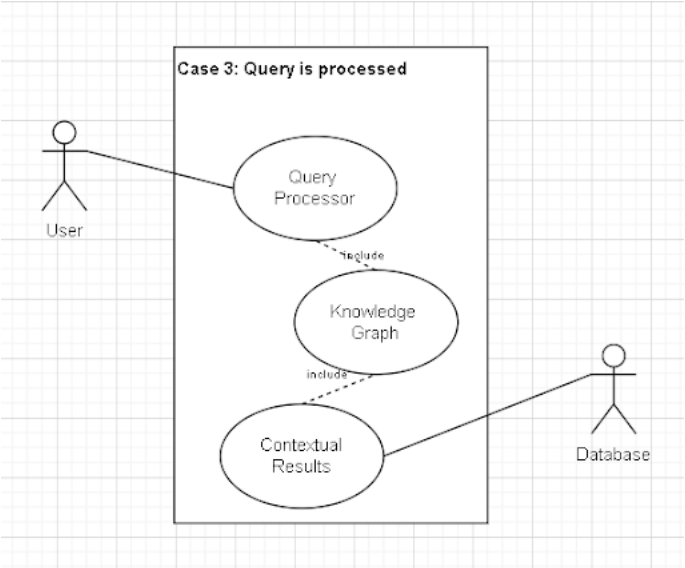

Figure 2.3: Use case for processing the query

Actors: User, Database
Pre-condition:

- User enters a query.

Post-condition:

- Return relevant results by searching in the knowledge graph.

Main Scenario:

1. Query results are extracted into keywords.
2. Graph is generated.
3. Relevant results are returned based on the query and graph.

Exception Scenario:

1. Query is not entered.
2. Knowledge graph constructed is incorrect.

Actors: User Interface
Pre-condition:

- Contextual spaces are created.

Post-condition:

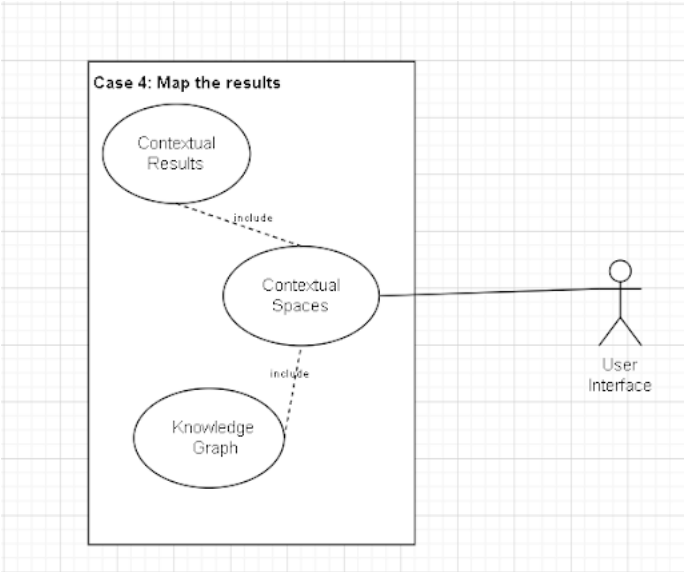

Figure 2.4: Use case for mapping the results

- Search results are mapped to one of the spaces.

Main Scenario:

1. Search results are mapped to the correct contextual spaces.

Exception Scenario:

1. Result is mapped to an irrelevant contextual space.
2. Contextual spaces are not generated.

- Availability of the system = 0.99
- Mean time of Failure (MTTF) = 100 hours
- Mean Time of Repair (MTTR) = 1 hour
- Mean Time Between Failure (MTBF) = 100+1 = 101 hours
- Availability MTTF / MTTR = 100 / 101

Performance requirements

The Response Time of the system is < 5 seconds. (15 processor, 100mbps).

2.3 Software and Hardware requirement specifications

Packages:

- networkx: Enables network analysis.
- matplotlib.pyplot: Supports data visualization.
- community: Employs community detection in networks.

Network Analysis:

- community.best_partition: Utilizes the Louvain method for community detection.
- nx.betweenness_centrality: Computes betweenness centrality.
- nx.closeness_centrality: Computes closeness centrality.

Visualization:

- matplotlib.pyplot: Utilized for graph visualization.

Third-party Libraries:

- wikipediaapi: Accesses Wikipedia data via API.
- rake_nltk: Extracts keywords using the Rapid Automatic Keyword Extraction algorithm.
- yake: Extracts keywords using Yet Another Keyword Extractor.
- spacy: Facilitates natural language processing.
- textblob: A library for processing textual data.
- networkx: Facilitates the creation, manipulation, and study of complex networks.
- keybert: Extracts keywords using BERT embeddings.

External APIs:

- Wikipedia API: Fetches information from Wikipedia.

Web Scraping:

- BeautifulSoup: A library for extracting data from HTML and XML files.

Text Distance Library:

- textdistance: Computes distances between sequences.

Additional Resources: Our code incorporates the "WordNetLemmatizer" from NLTK for lemmatization. Additionally, it utilizes the Damerau-Levenshtein distance from the "textdistance" library for distance computation.

Software/APIs:

- OpenAI GPT-3.5 Turbo API:
  - **Purpose:** Utilized for natural language processing and generating insights based on user prompts.
  - **API Endpoint:** "https://api.openai.com/v1/chat/completions"

Hardware Requirements:

1. Core i5 Processor
2. 8GB RAM

Chapter 3

PROPOSED SYSTEM

3.1 Description of Proposed System.

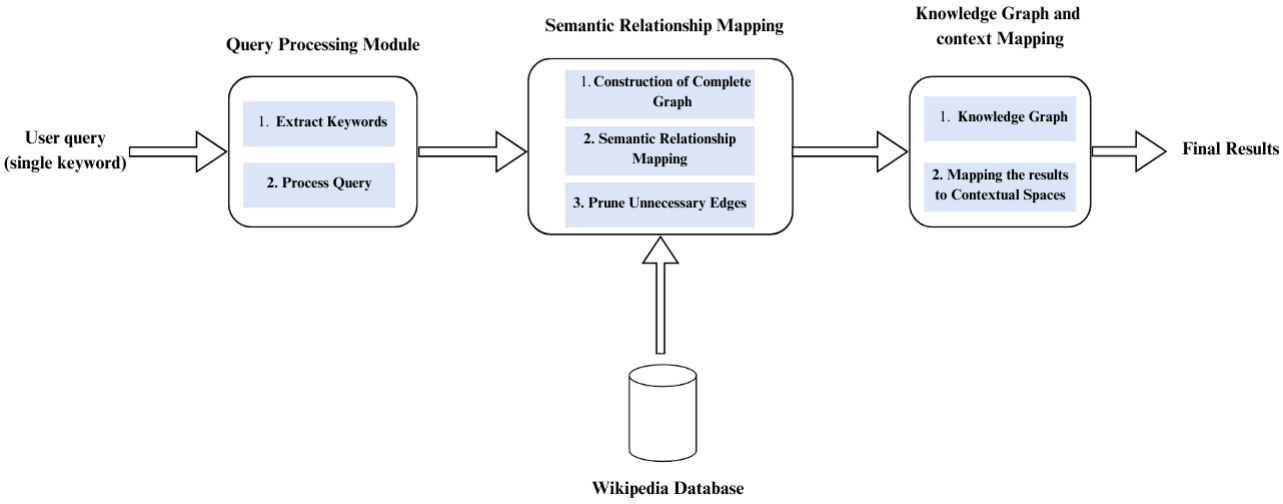

Figure 3.1: Diagram of Proposed System

The contextual search space system is a sophisticated approach to information retrieval. Unlike traditional methods focused on specific keywords, this system comprehends the connections between words, leveraging a vast knowledge database to provide detailed and meaningful answers. It involves multiple steps, including understanding user queries, creating a map of word connections, and utilizing an extensive knowledge repository. This approach excels in delivering answers that align with the broader context of user inquiries, making the search experience more insightful.

The proposed system comprises three main modules:

Module 1: Query Processing and Keyword Extraction

Module 2: Semantic Relationship Mapping

Module 3: Knowledge Graph and Mapping Results to Contextual Spaces

Module 1: Query Processing

The Query Processing module is responsible for the initial processing of user queries. It involves extracting relevant keywords from the input query to understand the user's intent. The goal is to create a structured representation of the query that can be interpreted by subsequent modules.

Module 2: Semantic Relationship Mapping

In Semantic Relationship Mapping, a complete graph is constructed to represent the semantic relationships between the extracted keywords. The graph serves as a comprehensive network that captures the connections between different terms in the context of the query. The semantic relationship mapping involves identifying and establishing links between keywords, creating a richer representation of the user's query. Unnecessary Edges Pruning is done further to optimize the graph's efficiency; unnecessary edges are pruned. This step involves removing connections that do not contribute significantly to the overall understanding of the query. Pruning helps streamline the graph, focusing on the most relevant relationships.

Module 3: Knowledge Graph and Context Mapping

Knowledge Graph and Mapping Results to Contextual Spaces involve the integration of a knowledge graph to further enhance the contextual understanding of the query. The system leverages existing knowledge to map the results of the semantic relationship mapping onto contextual spaces. This integration provides a broader and more nuanced perspective by incorporating external information and relationships.

3.2 Description of Target Users

A contextual search engine serves a diverse user base, with specific benefits for various categories. Content Creators rely on it to find fresh ideas and trends for their work, ensuring that their content remains engaging and up-to-date. Scientists use it to access scholarly articles and research data, staying informed about the latest developments in their fields. E-commerce Sites utilize the engine to optimize product listings, monitor market trends, and understand consumer behavior , enhancing their online presence and sales.

Data Scientists rely on it to find relevant datasets and analytical tools for their research and analysis needs. Students and Researchers use the engine to access academic resources and literature, improving the efficiency of their studies and investigations.

3.3 Advantages of Proposed System

- Contextual search engines deliver highly relevant search results by considering user context, enhancing the user experience.
- They offer personalized recommendations and content, catering to individual preferences and needs.
- By understanding user intent and context, these engines reduce irrelevant search results and improve precision.
- Users save time by quickly finding what they need without sifting through irrelevant search results.

3.4 Scope of proposed system

- The search engine can only process and search through text-based information, excluding any audio, video, or image content.
- It can only accept text-based queries as input and provide textual results as output.
- The search engine does not accept user feedback.
- Console-based application, no UI.
- The data is from Wikipedia only.

Chapter 4

SYSTEM DESIGN

The system design describes the interface and satisfies the specified functional requirements. This defines the architecture, context, and activity diagrams of the system.

4.1 Architecture of the system

We have chosen the pipe-filter architecture because it best suits our system. In the pipe-filter architecture, all processes and their intermediate outcomes are connected like a pipe. The outcome of one process acts as the input to another process, forming a connected flow. Therefore, all processes, along with their outcomes and inputs, are interconnected. The input to the system is a single keyword, and the output is multi-domain data specific to the keyword mapped to contextual results.

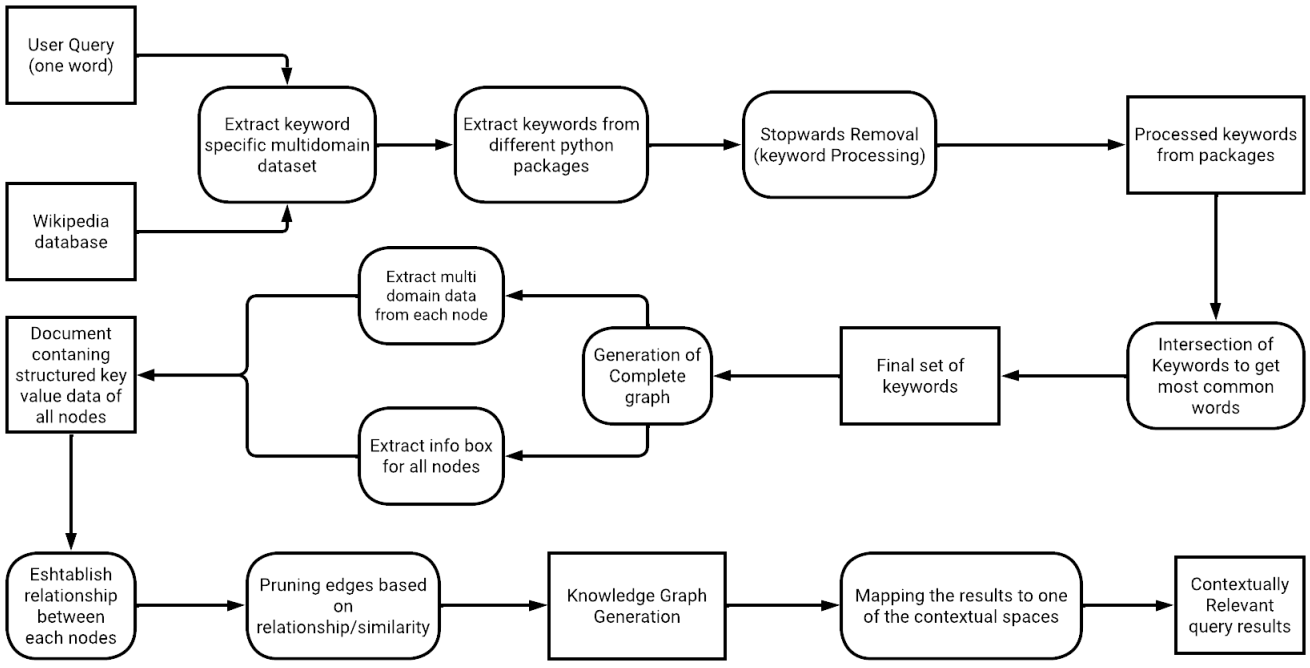

Figure 4.1: Pipe and Filter Architecture for proposed system

4.2 Activity Diagram

Activity diagram is an important behavioral diagram in UML diagram as it gives the dynamic behavior of the system. It is one of the advanced versions of flow charts that helps in giving an overview on the flow of the system from activity to activity. Additionally, it enables them to identify constraints and conditions that lead to specific events. If complex decisions are being made, a flow chart becomes an activity diagram.

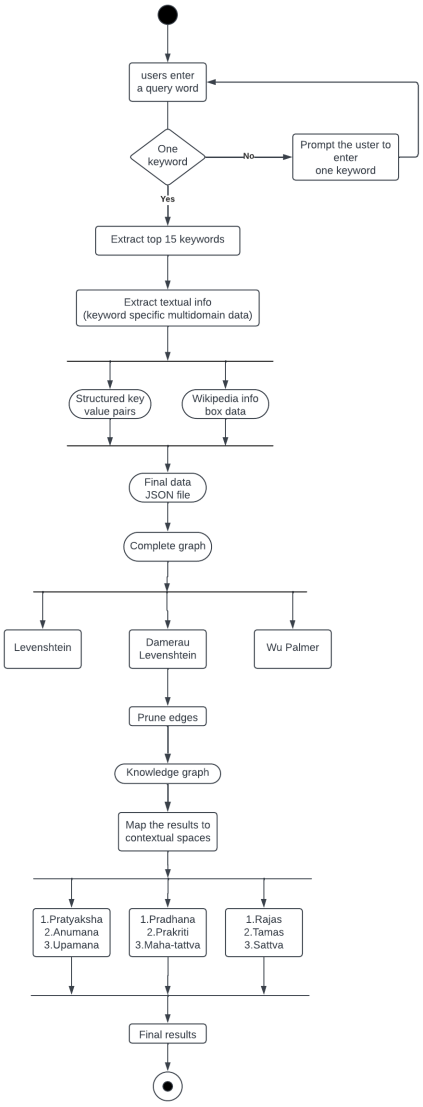

Figure 4.2: Activity Diagram for the proposed system

Chapter 5

IMPLEMENTATION

Implemented in Python 3.11.1, the system relies on Wikipedia as its primary database and utilizes five libraries—Rake, Yake, SpaCy, KeyBERT, and TextBlob—for keyword extraction. After preprocessing, including removing stopwords, the system identifies the 15 most common keywords across all sets.A complete graph is constructed using 'networkx,' and structured JSON data is extracted for each node, encompassing Wikipedia and infobox information. This yields a final document with organized key-value data, capturing keyword-specific multi-domain details.

For the knowledge graph, edges are pruned based on similarity measures like Normalized Levenshtein distance and Wu Palmer similarity. Various graph algorithms enhance structure and relationships. Results are mapped to contextual spaces, ensuring a comprehensive representation of multi-domain insights based on specified keywords.

5.1 Proposed Methodology

The proposed work consists of three modules: Query Processing module, Semantic Relationship Mapping, and Knowledge Graph Construction, along with Contextual Search Mapping.

5.1.1 Generating Keywords and Complete Graph:

In this module, the system processes user queries by extracting keywords to comprehend user intent. Utilizing the Wikipedia database, keyword-specific multidomain documents are fetched. Keywords are then extracted using Python packages such as Rake, Yake, SpaCy, KeyBERT, and TextBlob. The processed keywords are used to create a complete graph with the 'networkx' package. This graph serves as the foundation for subsequent analysis.

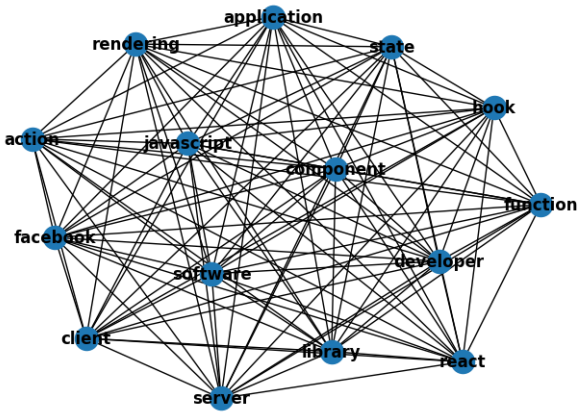

Figure 5.1: Complete Graph

```
Top 15 keywords are:
1. react
2. component
3. facebook
4. application
5. hooks
6. function
7. software
8. javascript
9. rendering
10. server
11. action
12. state
13. developer
14. library
15. client
```

Figure 5.2: Top Keywords extracted

5.1.2 Generating Keyword-Specific Multi-Domain Data:

Following the extraction of keywords, this module focuses on gathering detailed information about each keyword from the multidomain documents. The system employs various Python packages to process and refine the extracted keywords. The intersection of keyword sets is utilized to identify the most repeated keywords, forming the basis for further processing.

```
"function": {
        "wiki": {
            "title": "function",
            "summary": "Function or functionality may refer to:",
            "details": {
                "Computing": [
                    "Function key, a type of key on computer keyboards",
                    "Function model, a structured representation of processes in a system",
                    "Function object or functor or functionoid, a concept of object-oriented programming",
                    "Function (computer programming), or subroutine, a sequence of instructions within a larger computer program"
                ],
                "Function (music), a relationship of a chord to a tonal centre": [
                    "Function (musician) (born 1973), David Charles Sumner, American techno DJ and producer",
                    "\"Function\" (song), a 2012 song by American rapper E-40 featuring YG, Iamsu! & Problem",
                    "\"Function\", song by Dana Kletter from Boneyard Beach 1995"
                ],
                "Function (biology), the effect of an activity or process": [
                    "Function (engineering), a specific action that a system can perform",
                    "Function (language), a way of achieving an aim using language",
                    "Function (mathematics), a relation that associates an input to a single output",
                    "Function (sociology), an activity's role in society",
                    "Functionality (chemistry), the presence of functional groups in a molecule",
                    "Party or function, a social event",
                    "Function Drinks, an American beverage company"
                ],
                "Function field (disambiguation)": [
                    "Function hall",
                    "Functional (disambiguation)",
                    "Functional group (disambiguation)",
                    "Functionalism (disambiguation)",
                    "Functor (disambiguation)"
                ]
            }
        },
        "infobox": null
    },
```


Figure 5.3: Keyword specific multi-domain structured data


```
"javascript": {
        "wiki": {
            "title": "javascript",
            "summary": "JavaScript (), often abbreviated as JS, is a programming language and core technology of the World Wide Web,
              alongside HTML and CSS. As of 2023, 98.7% of websites use JavaScript on the client side for webpage behavior,
              often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on
              users' devices.\nJavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.
              It has dynamic typing, prototype-based object-orientation and first-class functions. It is multi-paradigm, supporting event-driven,
              functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular
              expressions, standard data structures, and the Document Object Model (DOM).\nThe ECMAScript standard does not include any input/output
              (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs
              for I/O.\nJavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of
              applications. The most popular runtime system for this usage is Node.js.\nAlthough Java and JavaScript are similar in name, syntax, and
              respective standard libraries, the two languages are distinct and differ greatly in design."
        },
        "infobox": {
            "Paradigm": "Multi-paradigm: event-driven, functional, imperative, procedural, object-oriented programming",
            "Designed by": "Brendan Eich of Netscape initially; others have also contributed to the ECMAScript standard",
            "First appeared": "December 4, 1995; 28 years ago (1995-12-04)[1]",
            "Stable release": "ECMAScript 2021[2] \n   / June 2021; 2 years ago (June 2021)",
            "Preview release": "ECMAScript 2022[3] \n   / 22 July 2021; 2 years ago (22 July 2021)",
            "Typing discipline": "Dynamic, weak, duck",
            "Filename extensions": ".js\n.cjs\n.mjs[4]",
            "Website": "ecma-international.org/publications-and-standards/standards/ecma-262/"
        }
    },
```


Figure 5.4: Structured data with infobox

5.1.3 Building the Knowledge Graph (Similarity Measures):

The system constructs a knowledge graph by pruning edges from the complete graph based on similarity measures. This involves calculating values such as Normalized Levenshtein distance, Wu Palmer similarity, Path similarity, and Jaro-Winkler similarity between each pair of nodes. Edges with weights below a set threshold are pruned, resulting in the creation of the knowledge graph.

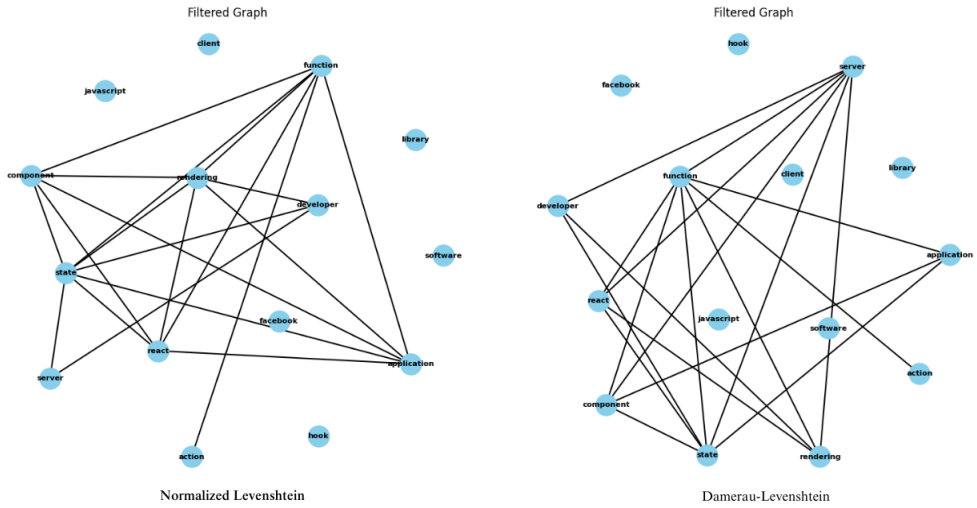

Figure 5.5: Knowledge Graph

Node pairs	Levenstein	Wu-Palmer	Damerau-levenstein
component ,react	0.47	0.20	0.29
component ,facebook	0.018	0.005	0.018
component, application	0.45	0.21	0.46
component, hook	0.041	0.011	0.041
component, function	0.375	0.187	0.37
component, software	0.025	0.013	0.025
component, javascript	0.026	0.017	0.026
component, rendering	0.41	0.0183	0.029
component, server	0.24	0.25	0.30
component, action	0.25	0.102	0.22
component, state	0.51	0.24	0.51
component, developer	0.15	0.26	0.24
component, library	0.02	0.008	0.02
component, client	0.05	0.038	0.05

Figure 5.6: Comaprision of similarity measures used

5.1.4 Applying Graph Traversals and Graph Algorithms:

This module focuses on applying graph traversals and algorithms to gain deeper insights into the knowledge graph. Techniques such as BFS Traversal systematically explore nodes at different levels, while DFS Traversal uncovers hierarchical relationships. Cluster Analysis examines thematic coherence, and Centrality Analysis identifies important nodes. Path Analysis utilizes shortest path algorithms to uncover quick connections between concepts[3].

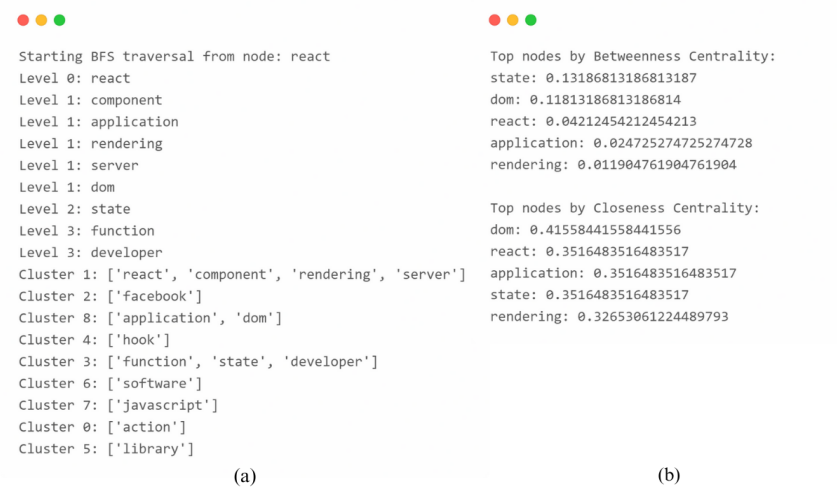

Figure 5.7: (a)BFS traversal and clusters (b) Closeness Centarlity and Betweenness Centrality values

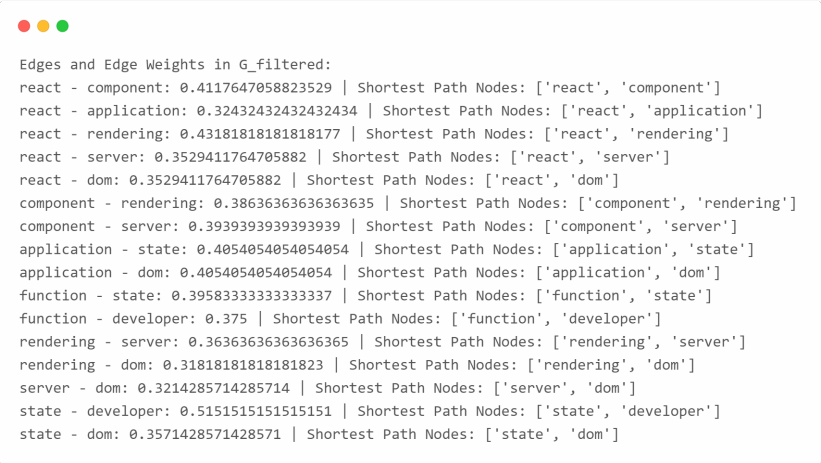

Figure 5.8: Edges in knowledge graph

5.1.5 Contextual Space Mapping:

The final module involves mapping the results obtained from the knowledge graph to predefined contextual spaces. These spaces, such as Pratyaksha, Anumana, and Upamana, provide different perspectives on the information. Additionally, contextual spaces like Pradhana, Prakriti, and Maha-tattva are defined, each representing different aspects of information. The system incorporates qualities like Rajas, Tamas, and Sattva, reflecting the dynamic nature, static information, and accuracy of the presented results. OpenAI is integrated to generate contextual suggestions for the user, enhancing the overall contextual mapping experience.

```
All Pairs Shortest Path Lengths:
Shortest path from react to react: 0 | Shortest Path Nodes: ['react']
Shortest path from react to application: 0.32432432432432434 | Shortest Path Nodes: ['react', 'application']
Shortest path from react to server: 0.3529411764705882 | Shortest Path Nodes: ['react', 'server']
Shortest path from react to dom: 0.3529411764705882 | Shortest Path Nodes: ['react', 'dom']
Shortest path from react to component: 0.4117647058823529 | Shortest Path Nodes: ['react', 'component']
Shortest path from react to rendering: 0.43181818181818177 | Shortest Path Nodes: ['react', 'rendering']
Shortest path from react to state: 0.7100840336134453 | Shortest Path Nodes: ['react', 'dom', 'state']
Shortest path from react to function: 1.1059173669467786 | Shortest Path Nodes: ['react', 'dom', 'state', 'function']
Shortest path from react to developer: 1.2252355487649604 | Shortest Path Nodes: ['react', 'dom', 'state', 'developer']
Shortest path from component to component: 0 | Shortest Path Nodes: ['component']
Shortest path from component to rendering: 0.38636363636363635 | Shortest Path Nodes: ['component', 'rendering']
Shortest path from component to server: 0.3939393939393939 | Shortest Path Nodes: ['component', 'server']
Shortest path from component to react: 0.4117647058823529 | Shortest Path Nodes: ['component', 'react']
Shortest path from component to dom: 0.7045454545454546 | Shortest Path Nodes: ['component', 'rendering', 'dom']
Shortest path from component to application: 0.7360890302066773 | Shortest Path Nodes: ['component', 'react', 'application']
Shortest path from component to state: 1.0616883116883118 | Shortest Path Nodes: ['component', 'rendering', 'dom', 'state']
```

Figure 5.9: All pairs Shortest path

```
Pratyaksha (Direct Edge):
In navigating a knowledge graph, Pratyaksha is like taking the express route.
If there's a direct connection (edge) between two nodes, it's the shortest and
most straightforward path. It's about efficiently reaching from point A to
point B without any intermediate places.


Anumana (Intermediate Nodes):
When there's no direct edge between nodes, Anumana comes into play. Imagine your
knowledge graph as a network of roads. Anumana is like planning your journey by
identifying and traversing intermediate nodes, making step-by-step progress toward
your destination. It's the art of finding the best route when there isn't a direct path.

Upmana (Comparing Node Data)
Imagine each place on the map (node) has details. Upmana is about looking closely at the
information in each location. It's not just about getting to the place; it's understanding
how the places are connected. By comparing the details at each stop, you learn more about
how things relate to each other on the map (knowledge graph).
```

Figure 5.10: Contextual Space I

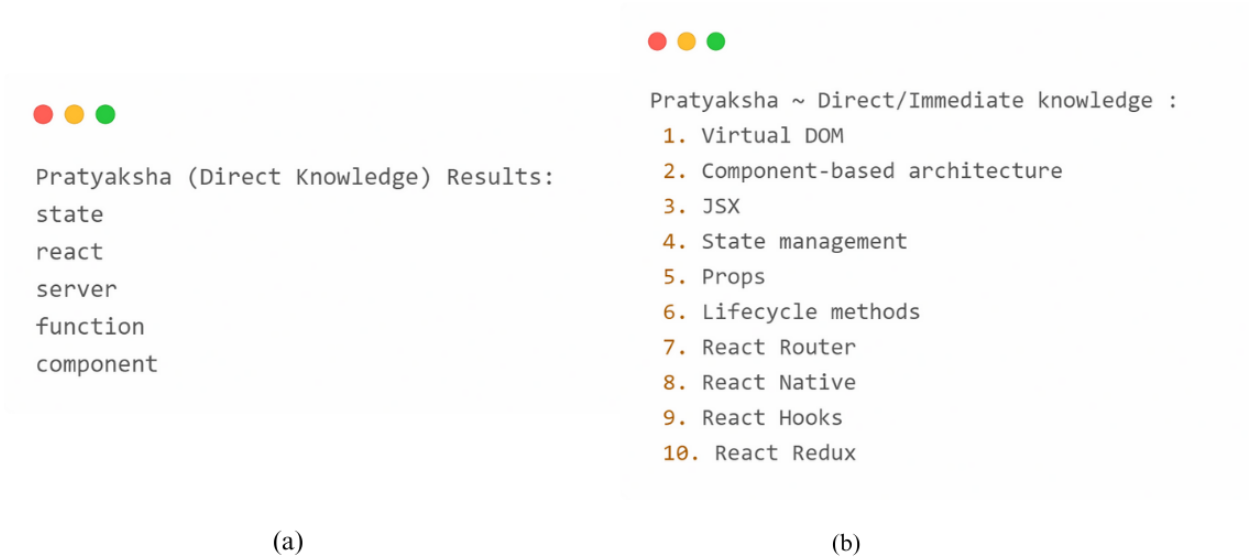

Figure 5.11: (a)Pratyaksha: Our results (b)Pratyaksha: OpenAI results

Figure 5.12: (a)Anumana: Our results (b)Anumana: OpenAI results

```
Pradhana:
In the search for information about each node, Pradhana refers to
the comprehensive textual file that encapsulates all possible data
available on the internet. It essentially represents keyword-specific
multi-domain data.

Prakriti:
Prakriti is the way we search for and organize information. It's the
process that takes what we find and neatly arranges it into a
structured file, like a well-organized list.

Mahatattva:
Mahatattva refers to the fundamental elements constituting the
discovered information. These essential elements are encapsulated in
a JSON file, where each piece of data is represented as key-value pairs,
ensuring a clear and organized presentation of the underlying information.
```

Figure 5.13: Contextual Space II

```
Node: application

title: application
summary: Application may refer to:
full_text: Application may refer to:

Mathematics and computing
Application software, computer software designed to help the user
to perform specific tasks Application layer, an abstraction layer
that specifies protocols and interface methods used in a communications
network Function application, in mathematics and computer science

Processes and documents
Application for employment, a form or forms that an individual seeking
employment must fill out College application, the process by which
prospective students apply for entry into a college or university
Patent application, a document filed at a patent office to support the
grant of a patent

Other uses
Application (virtue), a characteristic encapsulated in diligence
Topical application, the spreading or putting of medication to body surfaces
```

Figure 5.14: Contextual Space II: Pradhana(Unstructured data)

```
"application": {
        "wiki": {
            "title": "application",
            "summary": "Application may refer to:",
            "details": {
                "Mathematics and computing": [
                    "Application software, computer software designed
                     to help the user to perform specific tasks",
                    "Application layer, an abstraction layer that specifies
                     protocols and interface methods used in a communications
                     network","Function application, in mathematics and computer science"
                ],
                "Application for employment, a form or forms that an individual
                 seeking employment must fill out": [
                    "College application, the process by which prospective students apply
                     for entry into a college or university",
                    "Patent application, a document filed at a patent office to support
                     the grant of a patent"
                ],
                "Application (virtue), a characteristic encapsulated in diligence": [
                    "Topical application, the spreading or putting of medication to body
                     surfaces"
                ],
                "Apply": []
            }
        },
        "infobox": null
    },
```


Figure 5.15: Contextual Space II: Prakriti(structured data)

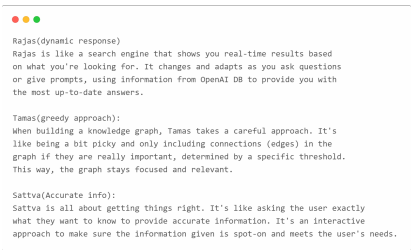

Figure 5.16: Contextual Space III

```
Information for 'facebook':
Wikipedia Info:
Title: facebook
Summary: Facebook is a social media and social networking service
owned by American technology conglomerate Meta Platforms. Created
in 2004 by Mark Zuckerberg with four other Harvard College students
and roommates Eduardo Saverin, Andrew McCollum, Dustin Moskovitz,
and Chris Hughes, its name derives from the face book directories
often given to American university students.

Infobox Data:
Type of site: Social networking service
Available in: 112 languages[1]
Founded: February 4, 2004; 19 years ago (2004-02-04) in Cambridge,
Massachusetts
Area served: Worldwide, except blocking countries
Owner: Meta Platforms
Founder(s): Mark ZuckerbergDustin MoskovitzChris HughesAndrew
McCollumEduardo Saverin
CEO: Mark Zuckerberg
URL: facebook.com
Registration: Required (to do any activity)
Users: 2.94 billion monthly active users
(as of 31 March 2022[update])[2]
Launched: February 4, 2004; 19 years ago (2004-02-04)
Current status: Active
Written in: C++, Hack (as HHVM) and PHP
```

Figure 5.17: Contextual Space III: Sattva(Accurate, needed information)

Chapter 6

TESTING

System Testing (ST) is a black box testing technique performed to evaluate the complete system's compliance against specified requirements.

In System testing, the functionalities of the system are tested from an end-to-end perspective. It includes both functional and Non-Functional testing.

6.1 Usability Testing

Usability Testing is a type of software testing where, a small set of target end-users of a software system, "use" it to expose usability defects. This testing mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives. It is also called User Experience Testing.

Test Case ID	Input Description	Expected Output	Actual Output
1	User enters a single keyword query	Complete Graph is generated	Complete Graph is generated
2	Generated complete graph is taken as input	Knowledge Graph is generated	Knowledge Graph is generated
3	Generated Knowledge Graph is taken as input	Mapping to predefined Contextual Spaces	Mapping to predefined Contextual Spaces

6.2 Performance Testing

Performance Testing is a type of testing to ensure software applications will perform well under their expected workload. Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, reliability, resource usage and scalability do matter. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks.

Test Case ID	Input Description	Expected Output	Actual Output
1	User enters a keyword	Graph generated in required time	Graph generated in required time
2	User doesn't enter keyword	System waits for user to enter keyword	System waits for user to enter keyword

6.3 Reliability Testing

Reliability testing is a type of testing to verify that software is capable of performing a failure-free operation for a specified period of time in a specified environment. Reliability means "yielding the same," in other terms, the word "reliable" means something is dependable and that it will give the same outcome every time.

Test Case ID	Input Description	Expected Output	Actual Output
1	User cuts off the internet	Program doesn't crash and reconnects after some delay	Program doesn't crash and reconnects after some delay

Chapter 7

CONCLUSIONS AND FUTURE SCOPE

7.1 Conclusion

In conclusion, the contextual search system effectively employs semantic relationships and efficient graph algorithms for knowledge retrieval. The advanced graph analysis techniques, including clustering, centrality measures, path analysis, and conceptual exploration, contribute to a comprehensive understanding of the knowledge graph. The systematic mapping of graph analysis results to predefined contextual spaces, coupled with OpenAI's additional insights, enhances the system's adaptability in delivering meaningful and contextually aligned search outcomes.

7.2 Future Scope

Looking ahead, exciting opportunities for improvement emerge. The integration of advanced NLP models like BERT or GPT will refine the system's understanding of user queries. Exploring data beyond Wikipedia aims to enrich the information landscape. The aspiration is to make edge weights dynamic, adapting to the evolving relevance and context of the knowledge graph.Dynamic adaptation of contextual spaces through user interactions, real-time updates, and user feedback will finely improve the contextual mapping process. There is a vision to process a broader range of information, breaking textual boundaries to include audio, video, and image content. Exploring multimodal search, incorporating image and voice inputs, along with deepening OpenAI integration for context-aware responses and dynamic suggestions, will enhance user-friendliness.As the system evolves, it seeks to align itself seamlessly with evolving information needs, offering a more comprehensive, adaptable, and user-centric search experience.

REFERENCES

[1] Author Name. *The 3 Gunas of Nature*. Year. URL: https://www.yogabasics.com/learn/the-3-gunas-of-nature/.

[2] Author Name. *Trust in Your Truth Through the Knowledge of Hinduism's Pramana*. Year. URL: https://www.yogapedia.com/trust-in-your-truth-through-the-knowledge-of-hinduisms-6-pramana/2/10822.

[3] Author Name. *Understanding the Levenshtein Distance Equation for Beginners*. Year. URL: https://medium.com/@ethannam/understanding-the-levenshtein-distance-equation-for-beginners-c4285a5604f0.

Appendix A

A.1 Context

Contextual search is an advanced search methodology that provides more individualized results by taking user purpose, location, and preferences into account. To accurately comprehend and translate user inquiries into executable commands, query processing entails methodically evaluating user queries. By finding significant relationships within a dataset, semantic relationship mapping improves understanding of contextually important relationships. A knowledge graph creates an organized representation of knowledge by arranging data into connected nodes and edges. These technologies allow for keyword extraction, automating the discovery of important terms or phrases from text for better summarization and categorization. They do this by utilizing Wikipedia databases as an extensive knowledge source.

A.2 Gantt Chart

TIMELINE	Week 1-4	Week 5-7	Week 8-11	Week 12-14
Literature Survey, Problem Identification, SRS				
Design and Training the model				
Implementation and Testing				
Report				

Figure A.1: Project management

A.3 Description of Tools and Technology used

A.3.1 Draw.io

An online editor called Draw.io is based on Google Drive. It facilitates the creation of flow charts, UML diagrams, use case diagrams, activity diagrams, as well as the system architecture. Since no software needs to be downloaded in order to use it, we can work with it directly in the browser. Additionally, since the data can be kept in Google Drive, it can be changed and many versions may be saved.

A.3.2 VS Code

Microsoft created the source code editor known as VS Code. It supports editing, constructing, and running various programmes and has a terminal for running commands and seeing xs1. It also contains a number of extensions that can be installed to support code optimization and improve the working environment for programmers, its sleek design and wide range of availability of language support integration with git and other version control systems it allows users to add customize their working environment and add additional functionalities, which makes it popular among programmers.

A.3.3 Git/Github

The main benefit of GitHub is that it stores various versions of the system, making it possible for users to access any version committed and retrieve the code. Git is an open source version control tool that aids in managing and keeping track of source code. Github can be used to store all the files and programmes necessary to run the system in a single repository and multiple users can read and edit the code, git repositories allow collaboration with other programmers, it helps to track, monitor and communicate. Github also provides features like continuous integration, code review.