PI^{2}RS: Public Infrastructure Information Retrieval System for Metro Cities

Sarthak Kaner
MT23081
Indraprastha Institute of Information Technology
Delhi, India
sarthak23081@iiitd.ac.in

Prerna Tyagi
MT23131
Indraprastha Institute of Information Technology
Delhi, India
prerna23131@iiitd.ac.in

Aman Gupta
MT23016
Indraprastha Institute of Information Technology
Delhi, India
aman23016@iiitd.ac.in

Sakshi Singh
MT23135
Indraprastha Institute of Information Technology
Delhi, India
sakshi23135@iiitd.ac.in

Mayank Singh Thakur
MT23123
Indraprastha Institute of Information Technology
Delhi, India
mayank23123@iiitd.ac.in

Rashi Khandelwal
MT23072
Indraprastha Institute of Information Technology
Delhi, India
rashi23072@iiitd.ac.in

***Abstract*—Public infrastructure data is one of the most important sources of information for collecting and analysing data regarding the development of a city and performing urban planning. Despite the ability of global processing systems like Google Maps being able to provide the location of nearby infrastructure such as schools and banks, there seems to be a gap in the ability to retrieve and gather information in relativity to gather and display the information regarding existing infrastructure and their facilities. Due to the scattered and unorganized nature of data regarding urban planning, collecting and fine-tuning data seems time-consuming and resource-intensive. Our proposed solution helps in gathering and organizing public infrastructure data in an organized format collected from trusted government and public portals that have made their data public. The proposed straightforward interactive system helps in information retrieval by deploying a select search mechanism that searches relevant information per the user's needs by implementing data processing and matching algorithms on the collected data while implementing query optimization on the questions the user asks. PI^{2}RS intelligently navigates through repositories of data, providing insightful information for the stakeholders such as business owners, policymakers, infrastructure planners and general citizens, empowering them in understanding and providing efficient public service to the target audience by culminating diverse data into one system through this interactive question-answering system.**

***Index Terms*—Public Infrastructure, Information Retrieval, Data Mining, Query Optimization**

I. Problem Statement

Accessing information regarding public infrastructure and urban planning data for different cities is a time-consuming and resource-extensive task as these types of data seem scattered in nature, where the information is a bit too complex to be understood by those unfamiliar with the relevant data. To resolve this issue, we are proposing an interactive system that will help in retrieving relevant information regarding the public infrastructure in a metropolitan city.

The aim of the project is to integrate information regarding all possible government and public infrastructure that helps in the day-to-day functioning of the society in the given cities. Our target stakeholders are not limited to the general public only who are in search of places to live and do their daily chores when migrating from their hometown, but also business owners and policymakers in understanding and evaluating the state of the society in which they are presently residing and operating in, and help improve the quality and quantity of public facilities that are available for regular usage.

II. Motivation

Public infrastructure includes all kinds of structures accessed by people around themselves to ensure the smooth functioning of society. Hospitals, banks, and metro stations are a few examples that can be categorized as public infrastructures that are being accessed on a frequent basis. It is a time-consuming and resource-exhaustive task for a person or an organization to gather information regarding the public infrastructure on an individual basis as per needs when they need to know about multiple public infrastructures in a particular region.

This constant and widespread need for information about public infrastructure across different regions and cities highlights the significance of developing a centralized system that can streamline the process of data collection and dissemination. Creating such a comprehensive system can help handle the issues of manually acquiring data from different sources. This would save time and resources while ensuring readily accessible and up-to-date data. Additionally,

it would enable cross-referencing and comparative analysis of public infrastructure availability and quality across different geographical areas.

The proposed system would be beneficial for a wide range of stakeholders, such as government agencies (to monitor and enhance public service delivery), urban planners (to identify areas needing infrastructure upgrades), researchers (to study infrastructure patterns and impacts), and the general public (to easily locate essential services nearby).

III. Literature Review

This section describes the literature survey done to develop a comprehensive system that can help in the information retrieval of public infrastructure data at the given platform.

Due to the presence of the vast amount of documents present on the web, work done by Houtinezhad et al. [1] talks about information retrieval methods applied to web crawlers which use a combination of vector space modelling and language statistical models to improve the retrieval of related documents. The proposed model calculates the similarity of the input term based on the shortest path between each term and the next term. Finally, the similarity using the path feature and cosine similarity is measured.

One such work by Esposito et al. [2] talks about the challenge of efficiently managing and retrieving a large volume of digital documents, particularly within the context of civil engineering projects and tender responses. The proposed solution aims to offer advanced document retrieval capabilities without requiring a strong background in computer science. By leveraging RESTful Web Services and integrating with systems like Alfresco and Solr, the solution enhances document retrieval through semantic search techniques, overcoming challenges such as synonymous terms and singular/plural forms.

The research [3] looked at making a Hindi chatbot. It gives info on mothers' and kids' health in areas with few resources. The chatbot works with FAQs and a health database. It understands questions in the Devanagari script and Latin script. It used Dynamic Time Pruning, Sentence Pair Classification, and Cosine Similarity to find alike question-answer pairs. These check how alike sentences are. One hard part was dealing with words that can mean different things. Query expansion methods helped some, but not a lot. The pre-trained encoders and fine-tuning plans really affected how well the chatbot worked.

On the other hand, this paper [4] focuses on the importance of text pre-processing in Cross-Language Information Retrieval (CLIR) systems and analyzes the impact of different pre-processing techniques on the performance of CLIR models. Tokenization divides text into individual tokens, which could be sentences, words, or characters. Stop words, which are common and frequently occurring words in a language, are removed as they do not contribute significant meaning to the content.The study utilizes Regexp Stemmer and regular expressions for stemming tokens. TF-IDF (Term Frequency-Inverse Document Frequency) is used to represent each document as a feature vector, assigning relevance weights to terms based on their frequency in the document and inverse frequency across the corpus.

IV. Novelty

The proposed system presents a novel approach to address gaps in reflected by research in public infrastructure retrieval by providing user-friendly, easily accessible, and optimised search functionalities. The system intends to offer a smooth and straightforward user experience, in contrast to existing systems with complicated interfaces and complex data, enabling users with different levels of technical competence to easily access public infrastructure data. It simplifies information retrieval by allowing users to ask questions in natural language with the use of Natural Language Processing techniques.

The proposed system makes access easier by combining data from several publicly available sources into a single platform, removing the need to browse between sources. It targets specific issues associated with urban settings, specifically in metropolitan areas, and provides personalized solutions for the location of adjacent medical facilities, educational institutions, and other critical infrastructure. Overall, the proposed system presents a comprehensive solution bridging the gap between existing scattered data and users seeking information about public infrastructure.

V. Methodology

For the proposed solution, we are defining a two-mechanism system whose utility mostly depends on the choice made by the users, whether the user wants to have a restricted query between themselves and the system or a free talk with the chatbot that would be doing the same thing as the mechanism with restricted querying. Hence, we are following the below-mentioned methodology to develop the proposed solution:

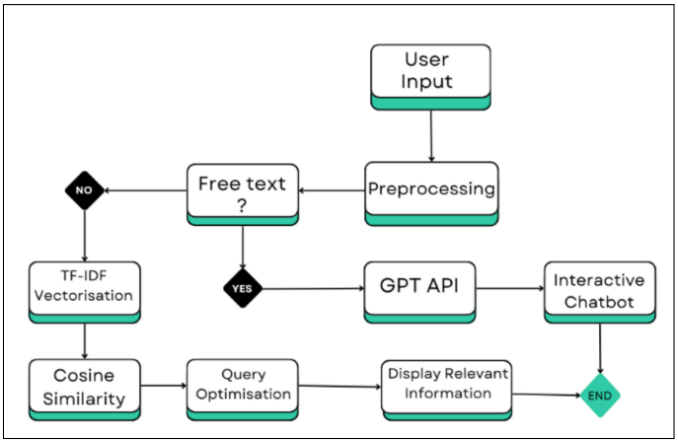

Fig. 1: Methodology

A. Pre-processing:

The question is pre-processed. This involves data cleaning, handling missing values, and the normalization process, which includes removing punctuation, stemming or lemmatization.

B. Decision Point:

A decision point is reached where the user decides which system mechanism determines whether the question is a free text question or not.

C. TF-IDF Vectorization:

If the decision points tend to give a decision that it is not a free text, it will go towards the restricted question-answering mechanism. Next, we performed TF-IDF vectorization. TF-IDF is a statistical method used to determine the importance of words within a document in a training corpus. Words that are common across all documents in the corpus would have a lower weight or score, while words that are unique to the specific document being processed would have a higher weight or score. This allows the system to identify the most important keywords within the question.

D. Cosine Similarity:

Here, the TF-IDF vector from the user's question is compared to the TF-IDF vectors for documents in a corpus using a cosine similarity algorithm. Cosine similarity is a metric used to determine how similar two documents are. The higher the cosine similarity score, the more similar the documents are.

E. Query Optimization:

The system then optimises the query developed after the input is taken from the users. On the basis of these queries, the data from the dataset is selected.

F. Display Relevant Information:

Based on the cosine similarity scores, the most relevant information is then displayed to the user.

G. GPT API Path:

If the question is not a free-text question, then the system takes a different path that utilizes the GPT API. GPT stands for Generative Pre-training Transformer and is a large language model chatbot developed by OpenAI. One such generative AI model is used for creating a chatbot that is an interface for communication between the user and system in a free-text manner,

H. Interactive Chatbot:

Once a response is generated, the chatbot enters an interactive mode to continue communicating with the user.

VI. Dataset

The dataset used for the project is collected from multiple trusted government and non-government portals and sources that provide public infrastructure data. The collected data is classified into three cities and five public infrastructural categories.
The three selected metro cities are as follows:

1) Delhi
2) Hyderabad
3) Mumbai

Five of the most used public infrastructures and facilities were selected for the given selected cities, and the data regarding those were collected. The selected datasets are as follows:

A. Schools:

In this category, the data regarding schools and colleges are collected, which consists of their location, address, classes, type of institute, whether central or state government and their contact info.

B. Hospitals:

Hospital and health facilities data were collected for all these cities, mostly consisting of which department they belong to, address, number of beds, and their names as well.

C. Transportation:

Transportation is one of the most important public facilities in metro cities, and they are bound to have infrastructure supporting these facilities. Four major transportation modes- Metro, Railways, Bus, and Local trains - were taken into consideration. Their stoppages, lines, source, destination, and station location data were added to the dataset.

D. Banks:

Bank data were also added to the public infrastructures as they are one of the most important public infrastructures. Bank data such as branch name, IFSC code, and branch location data were included in the dataset.

E. Government Offices:

Government Office information is also included in the dataset, which includes information such as location, the officer operating the office, the use for which the office is being allocated, and their contact info.

VII. Code

The PUBLIC INFRASTRUCTURE is a Streamlit application designed to provide information about various public infrastructure facilities in Delhi, Mumbai and Hyderabad. It begins with setting up the Streamlit app; User can first choose the city name from the options "Delhi", "Mumbai", and "Hyderabad". Then, users can choose different categories of public infrastructure using a dropdown menu (e.g., Banks, Hospitals, Transportation, etc). Users interact with the app upon selecting a category by providing

inputs relevant to their queries. Depending on the selected category, these inputs could include region names, school names, IFSC codes, etc.

Each category has associated functions imported from separate Python modules. These functions handle data retrieval and processing based on the user inputs provided through the Streamlit interface. For example, functions related to banks would retrieve bank details based on zone, branch name, IFSC code, or location, while functions related to schools would handle school-related queries. Text inputs provided by users are processed to extract relevant information and match them against the available data.

Functions preprocess text inputs and utilize techniques like cosine similarity to effectively match user queries with the stored data.

For example:
get_branch_details_by_zone(input_zone):
This function takes a zone name as input. It retrieves bank details such as bank name, branch name, IFSC code, and address for all banks located in the specified zone. The function likely queries a dataset or database containing information about banks in a city, filtering the data based on the provided zone name.

get_bank_details_by_ifsc(partial_ifsc_code):
Users input a partial or complete IFSC (Indian Financial System Code) code of a bank as input. The function searches for bank details based on the provided IFSC code. It retrieves details such as bank name, zone, branch name, IFSC code, and address corresponding to the specified IFSC code.

get_schools_in_region(region_name):
This function takes a region name as input. It retrieves and returns the list of schools located in the specified region. Internally, it likely queries a dataset or database containing information about schools, filters the schools based on the provided region name, and returns the filtered list of schools.

Retrieved information is displayed back to the user through the Streamlit interface. Results are presented in a structured manner, making it easy for users to interpret and utilize the information. Some basic error handling might be implemented to handle cases without matching information for the user's query.

VIII. Evaluation

For the proposed solution, we have introduced two evaluation metrics, the first one being the website's response time over the randomness of the question and the second one being the user satisfaction metric.

A. Response Time over Randomness:

In this evaluation, we are taking into account the time taken by the website to respond to the question asked by the user. The more random the question is, the more response time taken by the website due to the time taken to perform text and cosine similarity matching between response and query.

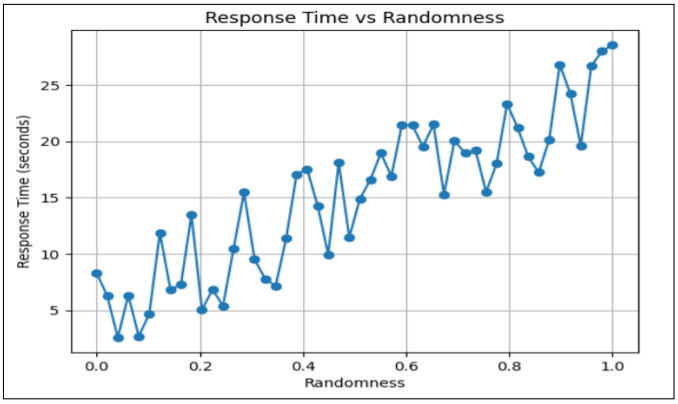

Fig. 2: Website Response Time v/s Question Randomness

The given graph shows that when there is the least randomness, it takes a significant amount of time as the system checks for the query for the first time. After some queries, it is seen to take less time comparatively as the system gets familiar with the dataset and query asked. However, for the upcoming queries, it is seen that there is a sudden spike in the graph. This is due to the change in the field of questioning, as it can be seen that the query generated is not very random.

The same trend can be observed as the randomness increases, but it is evident from the graph that randomness affects the website's response time, as the overall response time seems to increase with the increase in the randomness of a query. It also seems to be a bit less at complete randomness compared to a few randomness before.

B. User Satisfaction:

The second evaluation metric is User Satisfaction. The chart shows users' satisfaction with the system's ability to provide relevant answers to their questions. The green segment, making up 70.0% of the data, indicates that users were pleased with the answers received. This suggests that the system was contextually successful in understanding the user query and delivering useful responses within a specified time frame for the user's needs.

The red portion indicates that 30.0% of users were unsatisfied with the answers provided by the system. This dissatisfaction could be due to the system's inability to fully understand the user's query, giving irrelevant, incomplete, or unnecessary information, or being unable to fulfil the user's demand within the stipulated time frame.

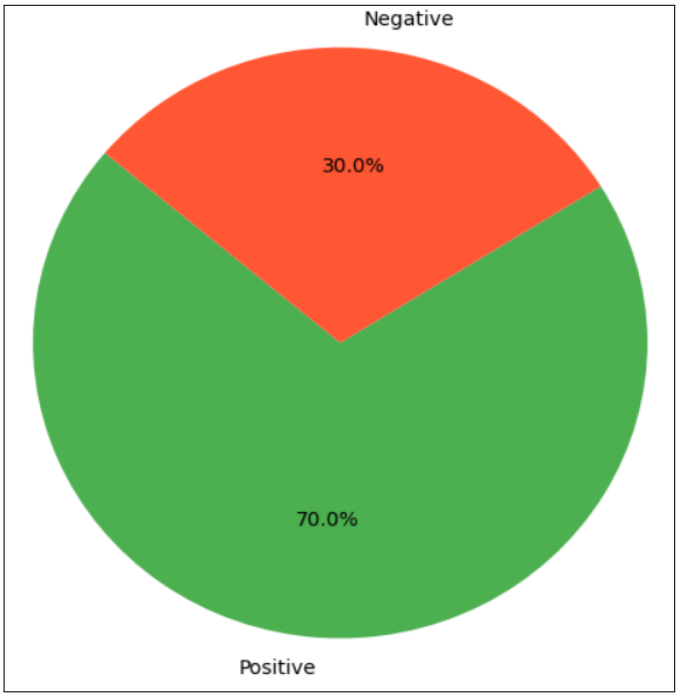

Fig. 3: User Satisfaction Pie

The graph depicts the human evaluation of the different types of questions asked to the system for which it can respond satisfactorily for about 70% of queries, whereas, for the rest, it could have been the case of unnecessary or incomplete data. It could also have been possible that the application was not able to retrieve the data even though it had the data in the search dataset.

IX. Conclusion and Future Work

In conclusion, this project introduces a reliable and creative way to solve long-standing problems with public infrastructure information systems. The proposed solution performs much better than existing systems by using advanced natural language processing methods like TF-IDF vectorization and cosine-similarity comparisons. It improves data organization and accessibility, as well as optimizes search capabilities.

This task aims at creating a better system for people. It helps them find information about public projects. This is very important for making good choices. The new system lets people easily access details. It makes searching simple and user-friendly. Overall, this project is a big step forward. It gives individuals power when deciding about infrastructure plans. With improved access and search abilities, the system can drive positive change. It can help advance public projects on a larger scale.

In the future, this project can be expanded to multiple cities and more public infrastructure, not limited to only the most important ones but also the ones that are most frequently used but not considered while developing the given prototype. Updating technologies and integrating such techniques are the most important future works that need to be done to develop such effective systems.

References

[1] M. Houtinezhad and H. R. Ghaffary, "Effective retrieval of related documents based on spelling correction to improve information retrieval system," in *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*. Bam: IEEE, Mar. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8405418/

[2] C. Esposito and O. Tamburis, "An Effective Retrieval Approach for Documents Related to Past Civil Engineering Projects," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. Napoli, Italy: IEEE, Jun. 2019, pp. 295–300. [Online]. Available: https://ieeexplore.ieee.org/document/8795368/

[3] R. Mishra, S. Singh, J. Kaur, P. Singh, and R. Shah, "Hindi Chatbot for Supporting Maternal and Child Health Related Queries in Rural India," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, T. Naumann, A. Ben Abacha, S. Bethard, K. Roberts, and A. Rumshisky, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 69–77. [Online]. Available: https://aclanthology.org/2023.clinicalnlp-1.9

[4] S. V. S and P. R, "Text Pre-Processing Methods on Cross Language Information Retrieval," in *2022 International Conference on Connected Systems & Intelligence (CSI)*. Trivandrum, India: IEEE, Aug. 2022, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9923952/