Universität Zürich[UZH]

Institut für Informatik

Enabling Efficient Use of Citizens' Feedback with Large Language Models

Master's Thesis in Informatics

submitted by

Xiaohui Wu

Student ID no. 21-738-604

Department of Informatics
of the University of Zurich
Prof. Dr. Gerhard Schwabe

Supervisor: Kilian Sprenkamp

Submission of the thesis: October 21, 2024

Abstract

In recent years, digital platforms have been increasingly adopted by governments to enhance citizen participation in policy making. A notable example is the European Commission's Have Your Say platform, which allows citizens to submit feedback and related documents. However, despite improvements to the platform, it still lacks effective tools to process and analyze the vast amounts of unstructured feedback. This results in valuable information from the feedback not being fully extracted, limiting its impact on policy making. To address this issue, this thesis explores the application of Large Language Models (LLMs) and Generative Artificial Intelligence to enhance the feedback processing workflow. A Retrieval Augmented Generation (RAG) model that combines information retrieval with text generation is proposed. This study adopts the Design Science Research Method (DSRM) and identifies the limitations of the existing platform through user analysis and presents targeted solutions. Based on these solutions, a web application called *Citizen Feedback Enhancer*, developed using the RAG model, is introduced to efficiently process and analyze large volumes of feedback. The performance and user experience of the *Citizen Feedback Enhancer* are evaluated through a questionnaire. Thus, this thesis contributes by demonstrating how advanced LLMs technology can be effectively applied to the field of citizen feedback processing to improve both the efficiency and effectiveness of information utilization in the policy making process.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Kilian Sprenkamp, for his guidance, support and valuable advice during the writing of my thesis. He also supervised my Master Project and Master Thesis during my master's degree. I am very happy to have had the opportunity to work and learn with him. I would also like to thank Professor Gerhard Schwabe and Liudmila Zavolokina of informatics department for their helpful suggestions and ideas during the process. My thanks also go to Professor Gilbert Friedgen in Luxembourg, who provided the initial idea for this thesis. His comments and our discussions helped us to determine the overall direction of the research. Also, thanks to Luo Jiayu and others for their help with the application evaluation, they provided a lot of interesting ideas to help me improve the application.

Special thanks to my family, who have given me consistent support, help and encouragement throughout my master's degree. My family will always be my strong backing.

Special thanks to my friends, Wang Yibing, Deng Zhingyuan, Xu Ruomeng and others, whose company allow me to have a colourful life and positive attitude in Switzerland.

Finally, I would like to thank the University of Zurich for providing me with a valuable learning platform and enabling me to progress in a strong academic environment.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
DSMR	Design Science Research Methodology
ICT	Information and Communication Technologies
IR	Information Retrieval
LLMs	Large Language Models
MMR	Maximum Marginal Relevance
NLP	Natural Language Processing
RAG	Retrieval Augmented Generation
UTF-8	Unicode Transformation Format (8-bit)
VSM	Vector Space Model
XML	eXtensible Markup Language

1 Introduction

In recent years, governments over the world have become increasingly interested in the methods and processes of citizen participation in policy making process. Many sociological studies have shown that the development and popularisation of digital tools has affected the scope and quality of citizen participation [37]. On the one hand, anyone can use these platforms and tools to express their suggestions and feedback. On the other hand, governments collect the data from citizens to promote policy formulation and development.

The Have Your Say platform, developed and managed by the European Commission, is one such democratic digital platform where citizens can provide a wealth of data to support government policy formulation by posting feedback and uploading relevant documents [9]. The platform has been updated several times and has become more and more user-friendly and functional. It offers a wealth of filter boxes to make it easier for citizens to find the initiatives that interest them. This greatly simplifies the process and cost of citizen participation, as the name of the website suggests, highlighting the democratic politics of the European Union.

However, while this digital platform lowers the threshold for citizen participation, there are also some limitations. Observing the data of different types of initiatives, it is found that some initiatives related to hot topics such as environmental protection and youth often receive tens of thousands of different citizen feedback. Since these feedback are unstructured text data containing multiple EU official languages, and some citizens may also submit relevant attachments at the same time, this reduces the efficiency of policy makers in extracting useful information from the feedback. The consequence is that a significant amount of useful information in the citizen feedback is not fully utilised, and some citizens' concerns cannot be conveyed through the feedback. In addition, the data in the existing platform lacks interactivity. For decision-makers, it is time-consuming and laborious to try to find the common demands of citizens from a large amount of feedback. Although the platform has analysed the distribution of countries and organisations of the commentators using visualisation methods such as charts, there is no direct and effective analysis of the text itself. Thus, this thesis ask the following research question:

RQ: How can a system be designed that incorporates citizen feedback from the Have Your Say into the policy making process effectively ?

To address the above limitations, this thesis proposes to use Large Language Models (LLMs) and Generative AI to improve the efficiency of using citizen feedback. Extensive experiments and tasks have shown that large language models (LLMs) such as GPT-4o are outstanding in terms of creativity [17]. With just the integration of text retrieval techniques and suitable prompts, it can generate complete and relevant responses. The Retrieval Augmented Generation (RAG) model combines the advantages of information retrieval and text generation, making it possible to develop more intelligent chatbots [20]. Although such chatbots have been widely used in business, medicine, customer service and other scenarios, their application in the field of citizen participation has not yet been fully explored [3]. Therefore, it is necessary to introduce this technology into the field of citizen feedback processing to improve the efficiency and effectiveness of the policy making process.

To answer the given RQ, a Design Science Research Methodology (DSRM) has been utilized in this thesis [27]. Problem identification and solution objective were first carried out through user-based analysis of the Have Your Say website. Based on the solution objectives, a chatbot integrated within a web application - *Citizen Feedback Enhancer* - has been developed based on the RAG model to improve the efficiency of processing and utilizing citizen feedback. We then demonstrated and evaluated the functions and application scenarios of this web application.

The thesis is structured as follows. Chapter 2 is a literature review summarising the use of LLMs in citizen participation and the development of the RAG. Chapter 3 presents the DSRM methodology. Chapter 4 identifies the problem in detail from four perspectives. Chapter 5 looks for solutions based on the identified problems. Chapter 6 presents the development process of the web application. Chapters 7 and 8 demonstrate and evaluate the web application. Finally, the thesis is summarised and future work is proposed.

2 Literature Review

2.1 The Evolution of Citizen Participation

Citizen participation in politics has a long history. In ancient Rome (509 BC-27 BC), institutions such as the Roman Senate allowed citizens to participate directly or indirectly in public decision-making. Similar things happened in China's Tang dynasty (618-907), where there was also an advisory system that provided a channel for the public to express their opinions. The emergence and development of these organisations provided a reference for a more robust civic participation system in the future. Since the 20th century, the importance of civic participation has received more attention due to the spread of democratic ideas. Carole Pateman believes that citizens are key of democratic politics and that citizen participation helps to generate a better democratic system [26]. Entering the 21st century, with the vigorous development of digital technology, citizens have gained more opportunities to participate in political decision-making, such as through electronic voting, online meetings and social media [15]. These methods share one thing in common: they involve democratic decision-making through electronic platforms and tools. This approach is known as e-participation [35]. E-participation lowers the threshold for citizen participation and provides people with a more convenient and faster way to express their opinions and feedback. However, due to uneven data resource allocation, this form of citizen participation faces difficulties such as the digital divide.

With the development of Artificial Intelligence (AI), some data has become open source and shared, removing some of the limitations of e-participation. In their 2018 study, Savaget et al. highlight the potential of AI in conjunction with open data, claiming that it could be applied in the field of policy making. [31]. Similarly, Jérôme Duberry also mentioned that AI can not only enhance the government's understanding of citizen feedback through data analysis, but also improve the efficiency of policy making through digital AI tools [7]. AI technologies such as text translation have become popular on e-participation platforms, making it easier for citizens to read documents and materials in different languages. Other AI technologies such as text clustering and sentiment analysis have also been integrated and used on some platforms. For example, 'Decidim' in Spain uses machine learning algorithms to cluster citizen feedback [36]. This helps policy makers to better identify common themes in the feedback. The 'CitizenLab', meanwhile, relies on NLP technology to perform sentiment analysis on different types of citizen feedback which helps governments to identify deep emotional information [6].

Apparently platforms mentioned above have integrated AI technology into the citizen participation process and improves the efficient use of citizen feedback. However, the features been integrated in those platforms focus on a single aspect, such as text clustering or sentiment recognition. These functions perform well in solving specific problems, but lack detailed analysis and summary of the whole text content. To improve the analysis of long text feedback on the platform, this paper proposes that the RAG system can be integrated into the citizen participation platforms. The RAG system has been widely used in question-answering systems, customer support and other scenarios [42]. However, RAG still has great application prospects for citizen participation platforms. By integrating retrieval and generation techniques, RAG can accurately search for relevant information from a large amount of citizen feedback even though the texts are unstructured and lengthy, and generate a summary at the same time. More importantly, the platforms can achieve more intelligent interactions with integrating the RAG system. It will greatly increase the efficiency of using feedback and improve the quality of citizen participation.

2.2 Information Retrieval and Generative AI

2.2.1 Information Retrieval Technology

Information retrieval (IR) has a long evolutionary history. It is studied to help users extract the required information from a large amount of literature. Traditional IR techniques convert text into vectors in a multidimensional space and use the similarity between vectors for key words matching [33].

The Vector Space Model (VSM) is a traditional model commonly used in the field of IR which represents both texts and queries as multidimensional vectors. It first generates a lexicon containing all the words based on the text, where the number of word categories represents the dimension of the vector and the weight of the word represents the element in each vector [32]. Then, the text data most similar to the query is found by similarity calculation. In VSM, cosine similarity is usually used to calculate the similarity between the query vector and the document vector. VSM is a relatively simple IR method that performs well for large-scale and low-vocabulary document collections [28]. However, it cannot extract semantic relationships in the text because it only uses word frequency for matching.

Best Match 25 (BM25) is another traditional IR method based on the probabilistic model [30]. It is considered an improvement on VSM. Specifically, in VSM method, words with high frequency have a greater impact on the similarity score. BM25 solves this problem by introducing new parameters. It prevents words with excessively high frequency from having too great an impact on the document score and allows the weight of word frequency to be adjusted appropriately when dealing with documents of different lengths, thus avoiding the problem of long documents being excessively punished or short documents being excessively rewarded.

With the development of computer technology and the explosive growth of data, IR technology has evolved from the initial traditional method based on the vector space model to modern methods based on complex vector embedding techniques and semantic understanding. As shown in Table 1, traditional IR and embedding-based IR have significant differences in various aspects. Vector embedding is the process of converting words, sentences or documents into dense vector representations [16]. Each piece of data then represents a point in a multidimensional space, with similar data closer together. By calculating the similarity between these vectors (usually cosine similarity), IR systems can retrieve data that is semantically and structurally related. There are many sub-classifications of vector embedding, such as word embedding, sentence embedding and document embedding. The most commonly used today are pre-trained models based on deep learning, such as BERT and OpenAI Embedding [39]. Their advantage over other vector embedding methods is that they have been trained on a large amount of text data and can capture more complex semantic relationships [41]. This approach not only improves retrieval accuracy, but also enables the system to better understand complex natural language input, and is suitable for semantic search, recommendation and question answering systems in various scenarios. In this thesis, the OpenAI Embedding model is used to generate vector embedding for the query data and the user query, and IR is performed by cosine similarity comparison.

2.2.2 Generative AI

The term "generative AI" is used to describe techniques that employ deep learning to train data and utilize generative models to generate human-like creations [11]. Early generative models were mainly in the field of speech, using encoding or deep learning

Feature	Traditional IR	Embedding IR
Model Basis	Geometric model	Neural network model
Similarity	Cosine similarity/Euclidean disdance	Cosine similarity/Vector distance
Term Frequency	TF-IDF weighting	Implicit representation
Length	Not considered	Implicit representation
Semantic Meaning	Low	High
Flexibility	Limited	High
Complexity	Low	High
Use Case	Text similarity	Semantic understanding

Table 1: Comparison of Traditional IR and Embedding-based IR

techniques to achieve speech synthesis and text-to-speech. Nowadays, generative models are used in a wider range of fields and there are very well-known generative models in fields such as images, text, audio and video. For instance, DALL-E 2 is capable of generating high-quality images and artwork. Concurrently, LLMs such as the GPT series have become the core technology for generating natural language texts [14].

The field of generative AI has developed rapidly in recent years, with an increasing variety of models being used and their performance improving. From the early introduction of models such as generative adversarial networks (GANs) and variational autoencoders (VAEs), to the more recent development of generative models based on the Transformer architecture, all have contributed greatly to the improvement of AI generation capabilities [22]. The emergence of the Transformer has totally changed the landscape of deep learning in NLP. It is because its self-attention mechanism has greatly improved the efficiency of processing long-range dependencies [40]. As a result, the current most popular pre-trained LLMs, such as BERT and GPT, are all designed and trained based on the principles of the Transformer [4].

Thanks to the development of LLMs, many AI interaction tools based on them have emerged. These tools are designed to answer user questions through real-time conversations. These tools, collectively referred to as chatbots, are often designed to handle complex questions and provide personalised services [1]. These LLM-based chatbots can understand and generate coherent natural language responses, and can even replace humans in providing services in certain scenarios [5]. But while generative AI can generate high-quality responses and realize fluent conversation, t also has a serious problem: the data becomes outdated. This is because they use pre-trained data, and over time their databases may contain outdated or incomplete information. This outdated data can cause chatbots to generate inaccurate responses, which can affect the user's judgement and experience [14]. To address this issue, the RAG model, which introduces external data sources, has been proposed to improve the accuracy and reliability of responses generated by generative AI [20].

2.3 RAG Framework

The RAG system was created to solve the problem that LLMs do not perform well when faced with data from different domains[20]. It is possible to combine external knowledge bases with the powerful generation capabilities of LLMs in order to produce useful responses [13]. The RAG system is mainly constructed of an efficient and accurate retriever and a powerful generator. Futhermore, a preprocessed external data source is also necessary for it. The process first begin with a search for target-related information in the external data source and then provides this information as context to the generator. The generator uses state-of-the-art LLMs and combines the retrieved information to generate complete responses. RAG is widely employed in scenarios that require human-computer interaction, such as dialogue generation and question-answering systems. As a result, chatbots based on RAG and special knowledge bases are being developed rapidly to align with the requirements of users in different fields. Figure 1 depicts the basic structure and processes of the RAG.

As the development and application of LLM has matured, the majority of existing RAG systems have adopted open-source LLM as their generation component, such as GPT series from OpenAI and Llama series from Meta. These RAG systems is possible to generate personalised output that meets the user's needs just by employing simple fine-tuning or prefix-tuning [21]. The initial deployment of RAG relied on traditional IR techniques such as TF-IDF and BM25, which depend on the frequency of keywords and hard to capture semantic relationships effectively [28, 30]. To enhance the precision of retrieval, researchers have employed vectorisation techniques such as the Dense Passage Retrieval (DPR) model [19]. This approach maps data and queries into a vector space and performs retrieval via vector similarity. Currently, pre-trained embedding models are widely adopted. They transform lengthy texts into high-dimensional vectors, which greatly enhances the accuracy of IR.

Although RAG systems are widely used in chatbots development, they still face some challenges. The first challenge is to ensure consistency between the retrieved results and the generated responses [18]. To overcome this, LLMs can be fine-tuned with specific data. Additionally, the creation of appropriate prompt templates can

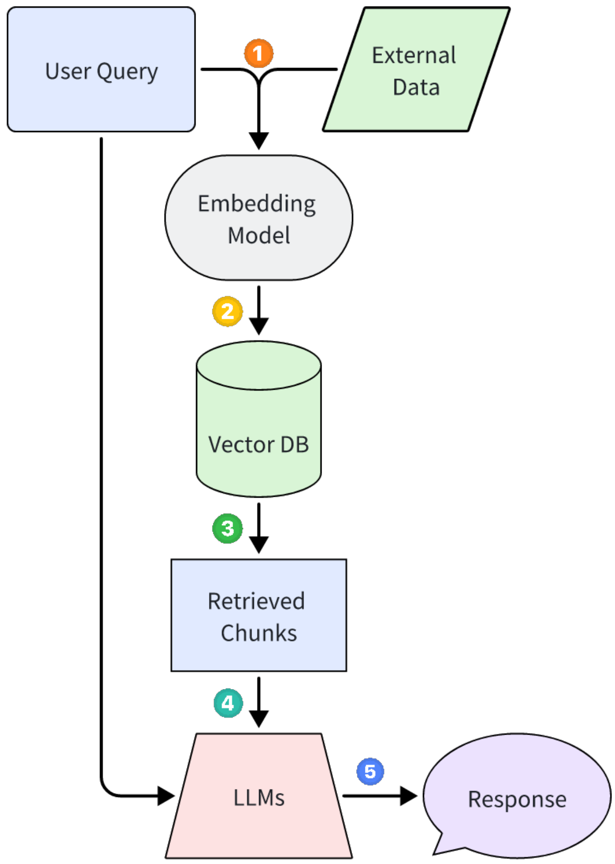

Figure 1: 1. Pass the query and external data to the embedding model to generate embedded vector; 2. Store the embedded data into database and transfer the embedded query to database; 3. Fetch retrieved chunks from database using embedded query vector based on different retrieval technology; 4. Send chunks as well as query text to pre-trained LLM model; 5. LLM will generate relevant responses based on given chunks and query.

assist in guiding the LLMs to more effectively utilize the retrieved data. Secondly, the integration and maintenance of external knowledge bases is a time-consuming and resource-intensive process. To solve this problem, the system needs to establish an automated data maintenance and update pipeline to ensure the timeliness and validity of the data.

3 Methodology

This thesis follows the DSRM proposed by Peffers et al. [27] to address the challenges raised in the Research Questions section. DSRM is a widely used framework for information systems to design and evaluate innovative IT solutions that address real-world problems. The DSRM approach has four different ppssible entry points for research, respectively are problem centered approach, objective centered solution, design and development centered approach and observing a solution. As the aim of this thesis is to provide an interactive solution to the problem that it is difficult to make efficient use of citizen feedback on the Have Your Say website, the problem centered solution was selected as the entry point to start the research.

The DSRM approach is structured into six phases which shows in Figure 2. Following these phases of DSRM, the main research methodology of this thesis are also divided into six phases, including (1) problem identification and motivation, (2) objectives of a solution, (3) design and development, (4) demonstration, (5) evaluation, (6) communication:

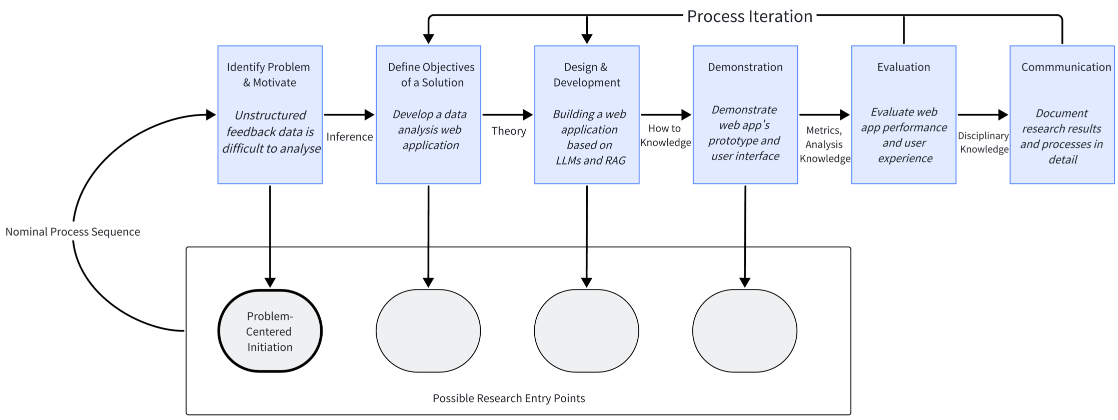

Figure 2: DSRM Model with Problem Centered Solution Entry Point

(1) **Problem Identification and Motivation:** The European Commission's Have Your Say website provides a platform for citizens and organisations to express their views and feedback. Citizens can submit text feedback or upload

relevant materials in a low-threshold manner. These platforms influence the decision-making process by collecting feedback from citizens and organisations on initiatives. However, extracting useful information from a large amount of citizen feedback is difficult and time-consuming. This is because the feedback texts are not only voluminous, but also written in multiple languages and with multiple attachments. These characteristics make the processing and analysis of feedback extremely complex.

(2) **Objectives of a Solution:** Inspired by the research of Kilian et al, this project develops an interactive web application for analysing feedback data [34]. The application effectively integrates citizen feedback into the policy making process. To increase the depth of textual data processing and analysis, the web application integrates a complete RAG model. Vector retrieval techniques are embedded in the RAG model and combined with the text generation capabilities of advanced LLMs (such as GPT-4o). To enhance the interactivity of the application, the entire user interface is designed as a chatbot. While interacting with the chatbot and receiving relevant responses, users can also view the content of related initiatives and the source of the feedback information. The aim is to enable users to quickly and accurately locate the most relevant information among hundreds of feedback, thereby improving the ability to extract key information.

(3) **Design and Development:** During this phase, the design and development of the web application prototype will be carried out. The prototype is developed using a variety of technology stacks, including MongoDB for data storage and management, Python for backend development, and the LangChain framework to implement complex NLP tasks. In addition, the project uses OpenAI's Embedding technology and GPT model to support text generation and analysis. The front end is developed using React, and the interface design is prototyped and iterated using Figma. To ensure efficient deployment of the application, the frontend is deployed via Netlify, while the backend is hosted on the AWS cloud service to ensure system reliability and scalability.

    In the DSRM model, this phase is iterative in nature. The technology stack and development environment used were selected after multiple rounds of screening and testing. They ensure that the project's requirements and costs are optimally met, and that the application demonstrates good performance and flexibility in practice.

(4) **Demonstration:** This phase will demonstrate the functionality of the application. The prototype will be tested using real data from the Have Your Say

website to demonstrate its ability to process and analyse feedback. This will include demonstrating the app's ability to handle large amounts of data, respond to different languages and interact with the user. A number of questions will be presented to the user during the demonstration, and the user will be able to ask the app further questions based on their own experience and interests. The app will display the questions and their responses and provide the source of the responses for reference. In addition, the demonstration will display how different output results can be achieved by adjusting different model parameters, such as increasing the number of contexts returned by retrieval process to improve the detail of the answers.

(5) **Evaluation:** The evaluation will consist of two iterative processes. In the early stages of the project, the evaluation focuses on the project's solutions and prototypes. The evaluation is mainly carried out by the project partners, such as my supervisor Kilian Sprenkamp. This evaluation improves and supplements the solutions and application prototypes by collecting their feedback. In the later stages of the project, the evaluation focuses on the performance of the application in real-world scenarios. We designed a user experience feedback questionnaire on the web application to collect opinions from stakeholders, academics, and organisations and people related to policy formulation. Through the questionnaire, we obtained their evaluation of system functions, ease of use, and practicality, and further optimised the design and functions of the application based on the feedback. Afterward, we analysed the results of around 20 questionnaires, discussed and summarised the benefits and limitations of the application in terms of design principles and development process.

(6) **Communication:** The final phase involves the documentation and communication of the research results. This includes a Master's thesis and a presentation of the research motivation and questions, the solution developed, the application development, the application evaluation and a summary and outlook of the whole research.

4 Problem Identification and Motivation

This chapter presents the first part of the thesis methodology. It will begin by discussing the development and three main functions of the Have Your Say website. It will then summarise its limitations and use these as motivation for the subsequent development of the thesis.

4.1 Have Your Say

The European Commission's Have Your Say platform is an online consultation tool for citizens and organisations [9]. It allows citizens and organisations to provide feedback at key stages of the policy making process and aims to increase the influence of citizens' decisions on policy formulation. Since its launch, the platform has undergone several updates and upgrades to make it more user-friendly [8]. The platform has improved its search function, making it easier for users to find topics of interest using keywords and drop-down boxes. At the same time, key consultation documents have been translated into all official languages to facilitate reading and reference by users in different regions. These improvements have lowered the threshold for citizen participation, with the aim of improving the quality of decision-making and encouraging wider social engagement. The main features of this website include the following:

Information filtering: The website provides many drop-down boxes to help users find relevant topics of interest, including but not limited to initiative topics, feedback opening hours and the initiative stages. A keyword search is also available to help users search for relevant topics rapidly and accurately.

Provide feedback: Users can click on the initiative they are interested in to download and view the details. Once successfully registered, users will be able to post feedback on the initiative within the specified period of time and also upload relevant attachments. This feedback will be displayed in text form on the website for easy reference by others. In addition, users may also express their suggestions by participating in public consultations and questionnaires.

Data visualization: Users can read the feedback they and others have posted. The content displayed for each feedback includes, but is not limited to, the name, identity and country of the feedback provider, but does not include account privacy. For each individual initiative, the website generates a category distribution map of respondents and a country distribution map of feedback providers.

4.2 Identify the Problem

However, while this digital platform lowers the threshold for citizen participation, there are also some limitations. Through literature reviews, data analysis, and direct interaction with the platform, these problems have been identified:

(1) **Analysis and evaluation of feedback are inefficient and insufficient:** The platform receives a large amount of feedback from citizens in various formats, most of which is unstructured free text. While some feedback includes attachments, the platform lacks the functionality to systematically analyze and evaluate this feedback. This forces decision-makers to manually review large volumes of text and files, which is time-consuming and resource-intensive, potentially leading to important information being overlooked. Furthermore, existing tools are limited to handling single texts or documents, making it insufficient to process large-scale feedback from multiple sources. These limitations reduce the efficiency of information utilization in the policy making process and weaken the connection between decision-making and public feedback.

(2) **Interaction and effective utilisation of feedback are limited:** The platform's current tools for interacting with citizen feedback are insufficient for decision-makers. While basic tools such as filter boxes help users locate topics of interest, they provide limited support for identifying the content and sentiment of the feedback. Although the platform includes visualization tools like geographical distribution maps of feedback providers, it does not offer content analysis of the feedback itself. This leads to the neglect of regional and organizational differences, making it difficult for policy makers to draw meaningful insights. This lack of interactivity diminishes the effective use of feedback data, which indirectly reduces the impact of citizen participation.

(3) **The quality and readability of feedback are inconsistent:** Although

the platform has lowered the threshold for citizen participation, it has also created challenges related to the quality of feedback. The feedback received from EU citizens is diverse, with some contributors providing authoritative, data-based feedback, while others post more emotional or repetitive comments. The inconsistency in feedback quality complicates the decision-making process. Additionally, the multilingual nature of the feedback and attachments, which sometimes include dialects or grammatical errors, further complicates the task of translation and interpretation, slowing down the review process.

(4) **The volume of feedback varies significantly between initiatives:** The platform experiences an imbalance in the amount of feedback received for different initiatives. Popular topics such as environmental protection and youth policy tend to attract thousands of responses, while less prominent or newly released topics receive only a few. This imbalance poses a challenge for decision-makers: on the one hand, when facing with large volumes of feedback, they may ignore a lot of key information and become tired of reviewing. On the other hand, it is hard to obtain the concerns of citizens with only a few feedback. Also without proper tools, consolidating and analyzing feedback across related topics is challenging.

5 Objectives of a Solution

This chapter proposes solutions to the four identified problems in the previous chapter. To address these issues, a complete RAG (Retrieval-Augmented Generation) framework was developed and integrated into a functional chatbot. The chatbot retrieves relevant feedback from the database in response to the user's question and provides the most relevant information as context. The LLM then uses this context along with the user's question to generate an accurate summary. Each specific solution corresponds to different functionalities of the chatbot and user interface, which are explained in detail below.

1. **Solution for problem (1):** To efficiently and sufficiently analyze and evaluate the feedback submitted by citizens, unstructured texts were processed and converted into high-dimensional vector embeddings using advanced NLP technologies. This structured data is indexed in a database, enabling efficient retrieval of feedback relevant to user queries. The system utilizes OpenAI's GPT model, which generates a well-reasoned analysis and summary based on the retrieved feedback and context. The combination of vector embeddings, which capture semantic relationships, and the LLM allows for a more nuanced and efficient analysis, addressing the inefficiency and insufficiency of feedback evaluation.

2. **Solution for problem (2):** In contrast to the information filtering approach used by the Have Your Say website, this RAG system focuses on filtering and analyzing feedback for decision-makers. The chatbot provides a direct interaction with the feedback data via a conversational interface. By summarizing and categorizing feedback based on criteria such as country, organization, and content, the chatbot enhances the effective use of information compared to manual review. Prompts are designed to ensure logical and consistent responses, allowing decision-makers to explore correlations and trends in the data, thereby addressing the limitations of current feedback interaction.

3. **Solution for problem (3):** Improving feedback quality and readability involves two key approaches. First, data processing removes duplicates and

ensures unique feedback entries. For feedback containing attachments, the system integrates the textual comment with the relevant attachment content. Additionally, the retrieval process prioritizes relevant feedback, filtering out noise that is not pertinent to decision-making. Language barriers are addressed by allowing the LLM to detect and respond in multiple languages. Even when feedback contains multiple languages, the system can generate responses in the user's target language, significantly improving readability and efficiency.

4. **Solution for problem (4):** The varying volume of feedback for different initiatives is addressed by enhancing the search and retrieval process. The web interface provides a keyword search feature that allows users to locate initiatives of interest and displays the number of feedback entries for each initiative. Additionally, users can adjust model parameters, such as the number of records to retrieve, ensuring more comprehensive responses when dealing with initiatives that have large amounts of feedback. This flexibility in retrieval helps balance detailed responses for heavily discussed topics and meaningful insights for less-engaged initiatives.

6 Design and Development

6.1 Dataset Creation and Management

6.1.1 Create the Dataset

The data utilised in this thesis as an external dataset for RAG is all from user feedback and its attachments on the Have Your Say website. The creation of the dataset involved steps such as scraping, text processing, and downloading files.

Scraping: In this process, the thesis designed three key functions to scrape the data step by step, ultimately obtaining the required metadata format. Each function accesses a specific URL, retrieves text content typically in JSON format. It then extracts the necessary content and outputs it as either a list or a dictionary for the next processing step.

- `get_init_id`: This function takes a set of parameters as input to retrieve the ID of each initiative from the URL `https://ec.europa.eu/info/law/better-regulation/brpapi/searchInitiatives?`. The format of this set of parameters is shown below:

```
params = {
    'topic': topic,
    'size': size,
    'language': language,
    'page': page
}
```

  This set of parameters allows the function to control the initiative topic during scraping, the number of initiatives displayed per page, the language, and the page number. The final output of the function is a list of IDs representing the initiatives, which are then passed as input to the next function. For example, the initiative "Energy security architecture – fitness check" is identified by the ID `14392`.

- `get_publication_id`: This function accepts a list of IDs as an argument and retrieves detailed information about each initiative from the URL. The output is a list with multiple elements, including the initiative ID, short title, publication ID, frontend stage, and total feedback number.
  - **Initiative ID**: The unique identifier assigned to each initiative.
  - **Short Title**: A concise description or name of the initiative.
  - **Publication ID**:The unique identifier for the official publication related to the initiative.
  - **Frontend Stage**: Indicates the current phase or stage of the initiative as it appears in the public-facing systems. The frontend stage helps users understand where the initiative is in its lifecycle, such as init planning, public consultation, or adoption.
  - **Total Feedback Number**: The total count of feedback submissions received for the current initiative.
- `get_feedback_info`: The purpose of this function is to fetch all feedback data related to a specificc publication ID. It does so by accessing the URL `https://ec.europa.eu/info/law/better-regulation/api/allFeedback?publicationId={}&keywords=&language=EN&page={}&size=10&sort=dateFeedback,DESC`, where the parameters within curly brackets represent the publication ID and the number of data pages to retrieve. This function ultimately generates a metadata format containing multiple columns for each initiative. The metadata is stored in the database for subsequent processing.


```
{
  "_id": "669e23aecc365e75faab59f4",
  "id": 12054,
  "shortTitle": "Support to European farmers - Transitional provisions",
  "topic": "Agriculture and rural development",
  "publicationId": 6017296,
  "frontEndStage": "ADOPTION_WORKFLOW",
  "totalFeedback": 13,
  "feedback": [
    {
      "language": "PT",
      "id": 502081,
      "country": "PRT",
      "organization": "Secretaria Regional da Agricultura e
```

```
                Florestas",
            "surname": "Planeamento",
            "scope": "REGIONAL",
            "status": "PUBLISHED",
            "dateFeedback": "2020/01/10 16:24:05",
            "publication": "WITHINFO",
            "userType": "PUBLIC_AUTHORITY",
            "companySize": "LARGE",
            "feedback": "O Gabinete de Planeamento da Secretaria Regional da Agricultura e Florestas agradece a oportunidade de comentar...",
            "login": "",
            "attachments": [
              {
                "id": 6232166,
                "fileName": "subvencoes_agricolas_-_regras_transitorias.pdf",
                "documentId": "090166e5cafc1019",
                "links": "https://ec.europa.eu/info/law/better-regulation/api/download/090166e5cafc1019"
              }
            ],
            "firstName": "Gabinete",
            "historyEventOccurs": false,
            "referenceInitiative": "COM(2019)581",
            "isMyFeedback": false,
            "isLikedByMe": false,
            "isDislikedByMe": false,
            "publicationId": 6017296,
            "publicationStatus": "CLOSED"
          },
          {
            // Other feedback objects...
          }
        ]
      }
```

For instance, for the initiative with ID `13695`, the metadata includes information such as the initiative's title, topic, publication ID, frontend stage, total feedback count and feedback. In this example, 13 pieces of feedback were received. However, for brevity, only the detailed format of one feedback item is shown. The `feedback` column is particularly important as it includes `attachments`, which represent any files submitted with the feedback. These attachments are typically PDF files, and the `links` field contains the download URLs for these files.

This metadata format ensures that all relevant information (including feedback and related attachments) is systematically stored and easily accessible. In subsequent operations, this metadata is extended, i.e. each feedback is stored as a separate record.

Data processing with attachment files: After obtaining the metadata in the previous step, the data needs to be expanded into a separate record for each feedback item to facilitate data retrieval and filtering. This process will cleanse and merge the data. Additionally, the metadata collected during the scraping process includes feedback items and associated attachments, typically in the form of PDF files. Often, feedback providers will upload the main content as an attachment, while the text area may only contain a prompt such as "See attachment." If this type of data is not processed, it may be ignored during data retrieval, or incomplete information may be extracted. Therefore, to make effective use of this information, several key processing steps need to be taken:

- **Attachment Downloading:** Some feedback messages may include one or more attachments. These attachments, usually in PDF format, are essential for fully understanding the feedback message. Information about the attachments is stored in the `attachments` column, which includes the attachment ID, name, and download link. The program identifies these attachments by parsing the metadata, specifically looking for the URL provided in the `links` field of each attachment object. Once identified, the program downloads the PDF files to a temporary location for further processing.

- **Text Extraction from PDFs:** Once downloaded, the contents of the PDF file need to be extracted. The system uses the PyMuPDF library, which can extract text from each page of a PDF. The processing of the extracted text in this program is to merge it directly into the original `feedback` column. This step is crucial to ensure that the full text content of the feedback is included in the final dataset and that no information is lost.

- **Text Preprocessing:** The text extracted from the raw feedback and attachments goes through a series of pre-processing steps. This includes removing unnecessary spaces, converting the text to lower case and handling line breaks. The aim is to store the text as a long string for analysis and embedding.

- **Embedding Generation:** The pre-processed text from the feedback and attachments is then used to generate an embedding using an artificial intelligence model. Here the program uses the 'text-embedding-3-small' text embedding model from OpenAI. This model supports text inputs of up to 8,191 characters and is relatively inexpensive, making it ideal for this type of high-volume text embedding task. By generating embeddings, the semantic information of the text is converted into a vector representation in a high-dimensional space that can capture the deeper meaning of the text content. These embeddings are used for information retrieval by creating a vector index via MongoDB Atlas.

- **Data Integration and Storage:** Finally, the pre-processed feedback text, extracted attachment content, and generated embeddings are combined into a structured format. Each feedback entry is stored as a separate record in the database. In addition to the existing metadata, a `combined` column is generated to store the merged data. The `combined` column is a single string that includes the following content:
  - **Title**: The short title of the initiative associated with the feedback.
  - **Content**: The pre-processed feedback text along with the extracted and preprocessed content from attachments.
  - **UserType**: The type of user providing the feedback.
  - **Country**: The country of the user providing the feedback.
  - **Organization**: The organization of the user providing the feedback.

After the above processing, the data is stored in the database in a structured form. Similarly, take the initiative with ID 13695 as an example. The data structure below is the first feedback of this initiative with feedback ID 502081.


```json
{
  "_id": "66ae3db0f29b0de6175aeae6",
  "feedback_id": 502081,
  "attachments": [
    {
      "id": 6232166,
      "fileName": "subvencoes_agricolas_-_regras_transitorias.pdf",
      "documentId": "090166e5cafc1019",
      "links": "https://ec.europa.eu/info/law/better-regulation/api/download/090166e5cafc1019"
    }
  ],
  "combined": "Title: Support to European farmers - Transitional provisions; Content: o gabinete de planeamento da secretaria regional da agricultura e flor...; UserType: PUBLIC_AUTHORITY; Country: PRT; Organization: Secretaria Regional da Agricultura e Florestas",
```

```
  "country": "PRT",
  "dateFeedback": "2020/01/10 16:24:05",
  "embedding": [Array (1536)],
  "feedback": "o gabinete de planeamento da secretaria regional da
     agricultura e flor...",
  "firstName": "Gabinete",
  "frontEndStage": "ADOPTION_WORKFLOW",
  "language": "PT",
  "organization": "Secretaria Regional da Agricultura e Florestas",
  "publicationId": 6017296,
  "shortTitle": "Support to European farmers - Transitional
     provisions",
  "status": "PUBLISHED",
  "surname": "Planeamento",
  "topic": "Agriculture and rural development",
  "totalFeedback": 13,
  "userType": "PUBLIC_AUTHORITY"
}
```

6.1.2 Data Storage and Management

6.1.2.1 Data Persistence and Storage

To achieve efficient data storage and retrieval, MongoDB Atlas was selected as the main database solution for this study. MongoDB Atlas is a cloud-hosted NoSQL database with high scalability and a flexible data model [23]. It can easily handle unstructured and semi-structured data. The `pymongo` extension package can be used in the code to enable efficient interaction between Python and MongoDB.

Throughout the data processing, Python scripts are used to first scrape and pre-process the data, converting the raw data into a structured format. The data is then stored in collections in the MongoDB atlas. To ensure persistent data updates, Github Actions are set up for database update operations. The Github Actions script is set up to automatically run the data scraping and database update scripts at the beginning of each month. This interval is determined based on the update frequency of the data source website and the cost of running the script. This persistent data update allows users to obtain the latest information on initiatives and feedback.

6.1.2.2 Vector Index Creation

To optimise data retrieval performance, search indexes were created in the database for this study to ensure fast and accurate responses when processing user queries. In practical terms, two different methods were considered: Atlas Search and Atlas Vector Search. The former is a full-text search engine suitable for exact matching and language processing tasks. The latter allows similarity searches of data in high-dimensional spaces and performs well when processing embedded vectors. Since both the user input query and the database data in this thesis were processed using high-dimensional embeddings, it was decided to use the Atlas Vector Search method to create a vector index.

When a user asks a question, the system first converts the user's query into a vector representation. Then, by comparing it with the pre-stored vectors in the database, the most similar documents are calculated. This process uses cosine similarity as a measurement standard. When the database is updated, the index is also automatically updated, which ensures data consistency. The specific index definition can be seen in the following data.

```
{
  "fields": [
    {
      "numDimensions": 1536,
      "path": "embedding",
      "similarity": "cosine",
      "type": "vector"
    }
  ]
}
```

6.1.2.3 Data Persistence and Storage Pipeline

The data persistence and storage pipeline, shown in the Figure 3, ensures efficient data management by automating data update and storage processes in MongoDB Atlas. Regular data scraping and updates are performed using GitHub Actions, and vector indexes are created to improve search speed and accuracy.

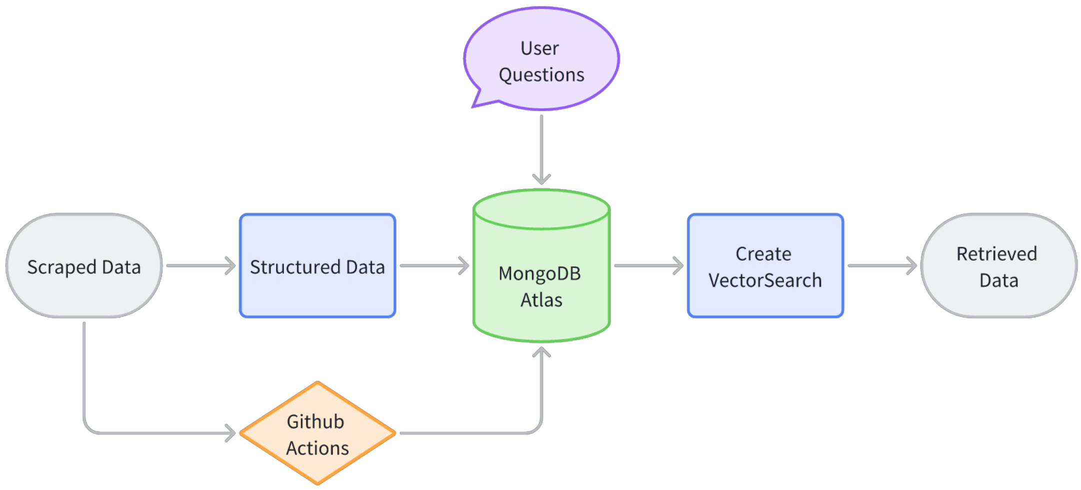

Figure 3: Data Persistence and Storage Pipeline Diagram

6.2 Web Application Development

This section describes the whole development process of the web application. In order to solve the problem raised in Chapter 1, i.e. *how to design a system that effectively incorporates citizen feedback into the policy making process*, we designed and developed a web application called *Citizen Feedback Enhancer*. Its main body is a chatbot and it integrates functions such as data retrieval, data analysis and data visualisation. These functions are designed to meet the solutions to the four problems proposed in Chapter 5. The following will be explained from the three aspects of backend development, frontend development and cloud deployment.

6.2.1 Backend Development

The backend of this web application is designed to process citizens' feedback on EU initiatives efficiently. This section states the technologies and methods used to develop the backend. The backend is responsible for retrieving data from database, interacting with the generation model, and passing the responses to the frontend.

- **FastAPI Framework:** Most of backend code is written in Python and uses FastAPI as the web framework for building APIs. FastAPI is easy to use, stable and have high performance in terms of speed, so it is ideal for combining with Python and Node.js as the web development framework [29]. The backend is built around multiple API endpoints, including:

  - **/query:** Handles user queries by invoking the retrieval and generation chains.
  - **/search_keywords:** Allows users to search for initiatives based on keywords and ID, returning topic, total feedback and links of the specific initiatives.

- **MongoDB Atlas Integration:** Feedback data is stored in the cloud-based MongoDB Atlas [23]. By integrating with MongoDB Atlas, the system can retrieve structured user feedback documents from several collections. In this step, the most important tool is the PyMongo package which is designing for establishing secure connection and accessing the relevant collection.

  Here, a *parse-parameters* function is constructed that can dynamically construct search filters as required. For example, the user can set parameters to return all data belonging to a specific topic. This function effectively narrows the scope of information retrieval and improves search accuracy.

- **Retrieval and Conversational Chains:** The core functionality of this web application is built on LangChain. This framework provides a variety of chain structures that simplify the interaction with LLM. In this project, LangChain is used to create retrieval question-answering (QA) chains and conversation chains.

  In the chain structure of LangChain, several key parameters are set to adjust the type of program and optimise the response results:

  - **llm:** This parameter is responsible for building LLM dynamically based on model name.
  - **retriever:** This parameter is responsible for creating retriever, defining the retrieval approach and adjust trieval granularity.
  - **return_source_document:** This parameter is responsible for determining whether to return the source document that generated the answer.
  - **chain_type_kwargs:** This parameter is responsible for passing extra param, such as prompt template.
  - **memory (optional):** This parameter is responsible for storing conversation history.

- **OpenAI API Integration:** OpenAI's embedding model and language model are integrated into this RAG system. The embedding model is used to convert the user's query into a high-dimensional vector that matches the embedding format of the data in the database. A language model such as *gpt-4o-mini* is used to construct the generator. The generator can be either fixed or dynamically constructed by adjusting parameters such as model type and temperature.

- **Prompt Templates:** To ensure that the responses generated have a fixed structure, this RAG system designs a prompt template. It guides the linguistic model to focus on specific aspects of the feedback, such as type of organisation, nationality and main concerns:

```
prompt_template = """
You are a helpful assistant responsible for providing
    detailed analysis and summaries of citizen feedback on
    EU laws and initiatives.

### Task:
You have been given a question: {question}. The person
    asking this question could be a policymaker, researcher
    , or anyone interested in public feedback. They need
    you to provide in-depth analysis and summaries based on
     citizen feedback to help them understand public
    opinions and concerns.

### Context:
The following contexts {context} have been provided to you,
     retrieved from a database of citizen feedback. Only
    use the information from the contexts provided to
    answer the question, and avoid speculation or using
    unprovided information. Due to the retrieval process,
    some contexts may be less relevant to the question;
    summarize these cautiously. If none of the contexts
    provide relevant information, politely express that you
     do not have enough information to answer the question.

### Context Structure:
Each context provided to you includes the following
    information:
- The type of user or organization: from context['UserType
    '] and context['Organization']
- The country of the feedback provider: from context['
    Country']
- A detailed and concrete summary of the feedback: from
    context['Content'] and context['Title']

```

```
    ### Requirements:
    Please summarize and analyze the content provided, paying attention to the following points:
    1. **Paragraph-based Summaries**: Avoid summarizing in a list format. Use paragraphs to separate different points when necessary.
    2. **Focus on Relevant Content**: Prioritize information that is directly related to the question. Briefly mention or skip content that is less relevant.
    3. **Polite Expression**: If you cannot provide a relevant answer, politely state "I'm not sure" or "Based on the provided context, I cannot answer this question."
    """
```

This template is the result of repeated experiments. It is the most complete and stable version that enables cluster analysis of the data and ensures a consistent response format.

- **Search and Retrieval Optimization:** To optimise document retrieval, the model's interface can receive different parameters to adjust the retrieval method, such as using similarity or MMR. In addition, adjustable parameters also include the model type, the amount of data returned and the type of search chain.

6.2.2 Frontend Development

The front end of this web application is mainly built using React [10]. The prototype is designed using Figma [12], and components are beautified using Material UI [38]. Functionally, the frontend interacts with the backend to retrieve and generate information. At its core is a chatbot through which users can ask questions and receive responses from the backend.

- **React Framework:** The React framework is the primary choice for frontend development. React is one of the most popular frontend development frameworks and is widely used in the production environments of many famous companies, such as Facebook and Netflix. Its component-based design allows developer to divide the interface into different modules. Each of these modules can be reused and managed independently. In this application, the interface is divided into the following components:
  - **Home:** This is the main interface of the application, which is for defining the overall structure. Other components are integrated into the main

interface.

  - **Navigation Container:** This is the navigation bar of the interface, including the sidebar buttons, title, and question input box.
  - **Chat Container:** This is the chat interface, which displays the chat content between the user and the chatbot.
  - **Source Container:** This is the interface that displays the retrieved data returned by the backend. It can list the title, content, user type, country and organization of the feedback.
  - **Sidebar Container:** This is the sidebar of the application, which is used to modify the settings of the search chain and implement keyword search.

  The distributed design of these components improves development efficiency and code maintainability. In addition, React also supports state management and logic reuse for components and parameters through custom hook functions (hooks).

- **Interaction with Backend:** The interaction between the frontend and backend is implemented using hooks. These functions use Axios to send HTTP requests and handle responses, thereby enabling communication between the user interface and the backend.

  When the user submits a query in the chat interface, the hook sends the query to the backend using an HTTP POST request. The selected topic, chain type, and generated model are also sent as parameters. The backend receives the query and returns a response, which includes the chatbot's responses and the retrieved resource.

  In addition, when the user wants to search for information about an initiative using keywords, an HTTP GET request is sent to the backend with the input. The database returns matching search results based on the input criteria, and the results are displayed in real time.

- **User Interface Design:** The user interface is designed as a streamlined chatbot, with a google-like light mode. The design is based on the most famous chatbot, *Chatgpt*. The main design elements include:

  - **Search Bar:** An intuitive search bar where users can enter and send queries, with support for multi-line input and voice input.

- **Floating Sample Question Buttons:** Interactive buttons that suggest sample questions to the user. It allows a quick and easy query generation.
- **Drawer-style Sidebar:** A collapsible sidebar that does not take up space on the main page and allows users to switch between different functions.
- **Result Display Area:** A specific area in the centre of the interface, which mainly displays the conversation between the user and the chatbot and the returned source data.
- **Rating Area:** The three-line rating stars is used to display the system's correlation between questions, responses and sources. The score is generated automatically, and the higher the correlation, the more stars there will be.
- **Interactive Elements:** The main interactive elements include dropdown menus, clickable links, and dynamically changeable parameters. Also there will be a button for generating summary of initiatives.

- **Key Functions Design:** The application includes several key functions to enhance the user interaction experience and data retrieval capabilities. These features include:
  - **Real-time Chat Interaction:** Allows users to submit queries and receive instant responses from the chatbot.
  - **Source Data Display:** The data sources used by the chatbot to generate responses are displayed next to the main conversation. The data sources are presented in a clear structure and paginated according to the size of the retrieved data.
  - **Data Filtering:** Allows dynamic filtering of the original data set by setting different topics. The content returned is all related to the selected topics, helping users to narrow their search.
  - **Retrieval Parameter Adjustment:** Users can dynamically adjust the parameters applied by the RAG system, including retrieval chain type, search methods, model name, number of results, etc.
  - **Keywords Search:** Allows users to perform a keyword search based on the initiative ID and title. A link to the initiative and the total number of feedbacks are displayed. Users can adjust their search strategy based on the number of feedbacks.

6.2.3 Cloud and Deployment

The web application is deployed using a combination of Amazon Web Services (AWS) services (for the backend) and Netlify (for the frontend) [2, 24]. This cloud deployment makes it easy to maintain and scale the application.

6.2.3.1 Backend Deployed on AWS

The backend is deployed on AWS. The reason for choosing AWS is that it offers flexible computing resources and strong security features. We mainly leverage the following key services:

- **EC2 (Elastic Compute Cloud):** AWS EC2 instances host the FastAPI application and use automated scripts to keep it up to date and running. These scripts are implemented as systemd services and ensure that the application is restarted immediately after any code update. The instances are configured to efficiently handle incoming API requests and provide stable performance.
- **API Gateway:** AWS API Gateway is used to manage API requests from the frontend. Although other API services such as load balancers were tried, API Gateway was chosen as the API service because it is better suited to handling multiple interfaces. After creating the API, resources and HTTP methods (e.g. GET, POST) are defined to correspond to different paths in the frontend. For example, define a /query resource and set the POST method to receive query requests from the frontend. In addition, the API Gateway must be integrated with the EC2 instance. This means that API Gateway sends data via an HTTP request to the FastAPI application running on the EC2 instance. By deploying API Gateway, a public HTTPs endpoint can be obtained for interaction between the frontend and backend to ensure data security.

6.2.3.2 Frontend deployment on Netlify:

The frontend is hosted on Netlify. This is a platform that is dedicated to deploying and hosting web applications such as React. Netlify's deployment process is very simple. Users only need to connect a GitHub repository, and whenever there is a code update pushed to GitHub, Netlify will automatically builds and deploys the latest version of the application. This continuous integration and deployment (CI/CD) process eliminates the need for manual deployment.

Another key benefit of Netlify is that it is free. For small and medium sized applications, Netlify's free hosting service is sufficient. It also provides free custom domain name services, so users can create custom domain names as needed. The web application *Civic Feedback Enhancer* is deployed on Netlify with a custom name *eufeedback-enhancer*[1].

[1] https://eufeedback-enhancer.netlify.app/

7 Demonstration

This chapter demonstrates the basic functions of the Citizen Feedback Enhancer. The main sequence of the demonstration will be the key features mentioned in Chapter 6. The demonstration will include real page screenshots and text descriptions.

- **Main Interface Demonstration:** Figure 4 displays the main page of the website, including the site name and page content. The website URL is automatically assigned by Netlify after customisation [24], providing a convenient and user-friendly solution for users who do not have a custom domain name. The overall style of the main interface is that of a chatbot, inspired by the most popular chatbot of the moment, ChatGPT [25]. While the app includes additional features, the design primarily encourages users to interact with the citizen feedback text. The page layout, from top to bottom, consists of the headline, prompt, input box and quick question-and-answer button. On the left side of the page is a collapsible sidebar, which is designed to save space on the main interface. The page is optimized for a screen resolution of 2560×1440 but also adapts to slightly larger or smaller screen resolutions.

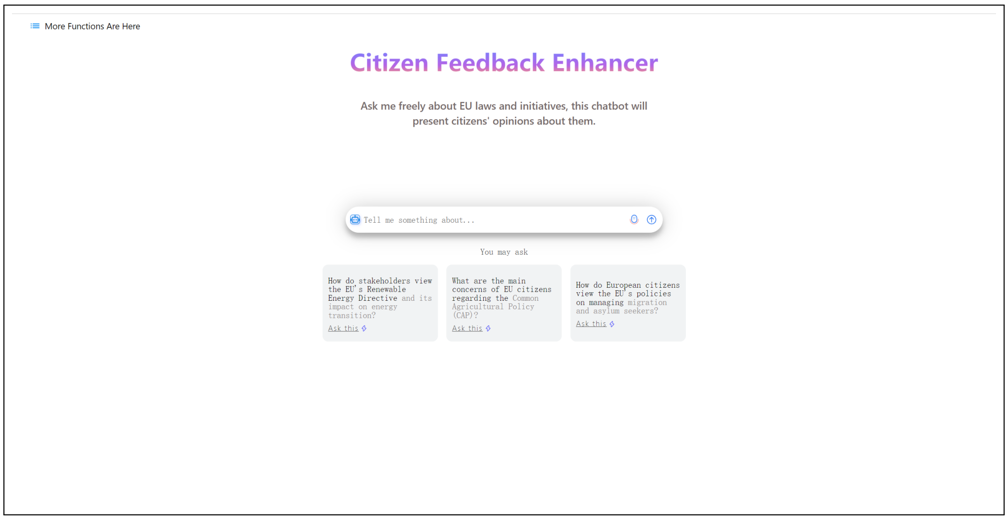

Figure 4: Main Interface

- **Real-time Chat Interaction and Source Data Display:** When the user inputs a query and clicks on the Send button, the web app will transitions from the main page to the chat page as shown Figure 5. The middle section of the chat page displays the conversation between the user and the chatbot, while the right section presents the source content used to generate the response, which is filtered from the database via IR.

For example, in the case depicted in Figure 5, the user asks the question, *"What is the public opinion on the EU's Erasmus+ programme and its impact on student mobility?"*. For this query, the source panel on the right shows feedback from citizens in countries such as Greece, Austria and Norway, covering issues such as *"the low number of participating students, teacher and student mobility, financial support and academic compatibility"*. By reviewing this text, users can trace the sources that informed the answer. Below the question, the chatbot generates a summary response based on this feedback, highlighting the benefits and challenges of the Erasmus+ programme and suggesting some improvements, such as *"the need for stricter selection criteria and more adequate financial support"*.

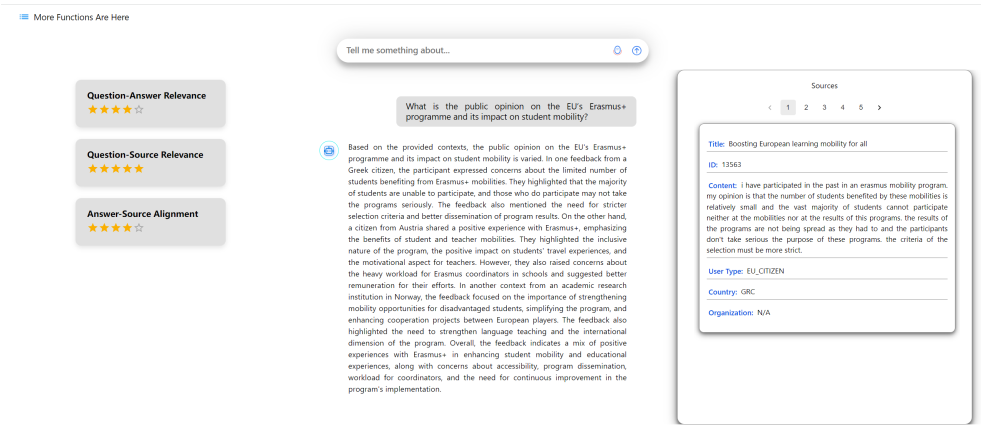

Figure 5: Chat Interface

- **Data Filtering and Retrieval Parameter Adjustment:** This section demonstrates the main features of the web application sidebar, including filtering data and adjusting retrieval parameters. As illustrated in Figure 5, we first set the 'Topics' to 'Education and training (EAC)' and then adjust the 'User Type' parameter to 'Academic Research Institution', which means we would like to know what academic institutions' attitude towards this initiative. This instructs the database to retrieve 5 pieces of feedback data that are most relevant to the user's query and fall under both the EAC topic and user type academic research institution. Based on the retrieved results, it can be concluded that the RAG system has successfully returned 5 pieces of data that match the specific topic and user type that are highly relevant to the user's question. Additionally, adjusting other parameters can influence the model's response in different ways.

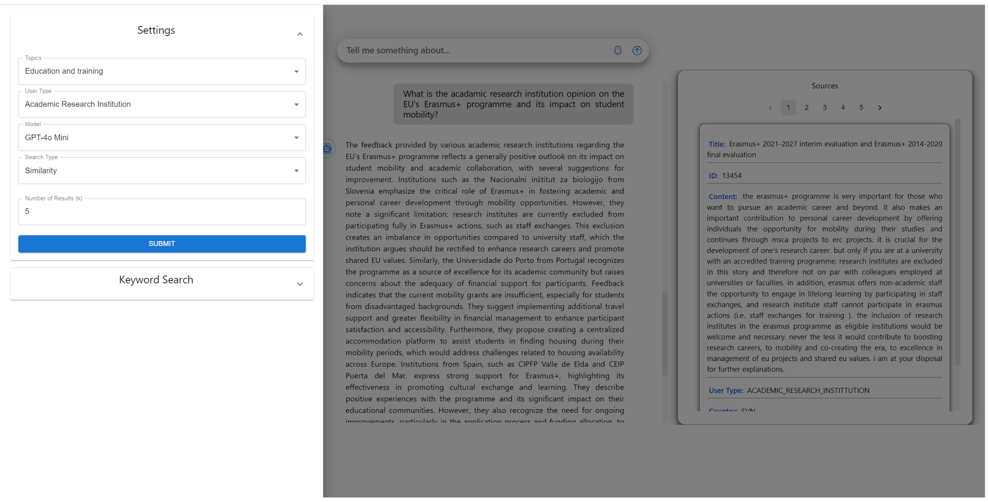

Figure 6: Data Filtering and Retrieval Parameter Adjustment in Sidebar

- **Keywords Search and Summarization:** The keyword search functionality is demonstrated in the Figure 7. By entering the title or ID of a relevant initiative, users can access basic information about it, including the topic, the number of feedback entries, and a website link. Moreover, users can generate a summary to gain an initial understanding of the initiative's motivations and approaches. Users can adjust the number of returned results based on the feedback count and the brief summary, which help to enhance the relevance and accuracy of the response. For more detailed information, users can follow the provided link, as illustrated in Figure 8.

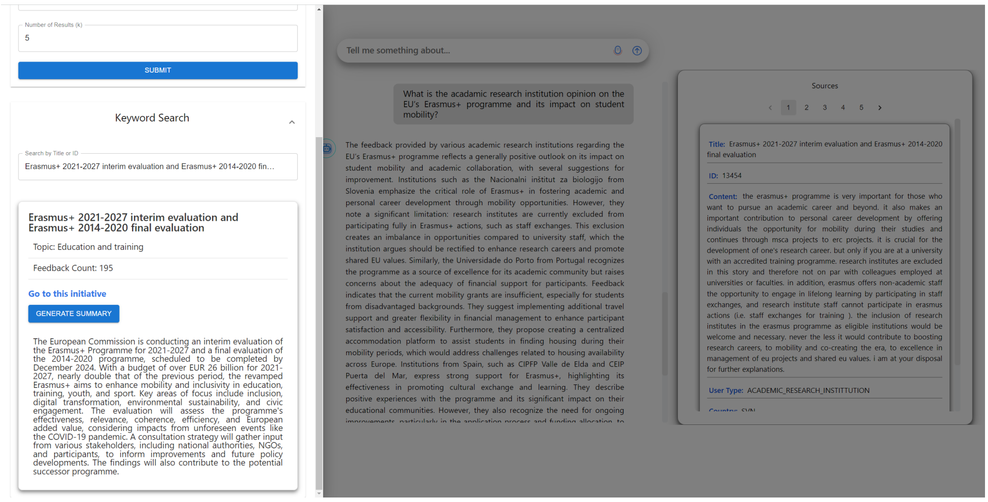

Figure 7: Keywords Search and Summarization in Sidebar

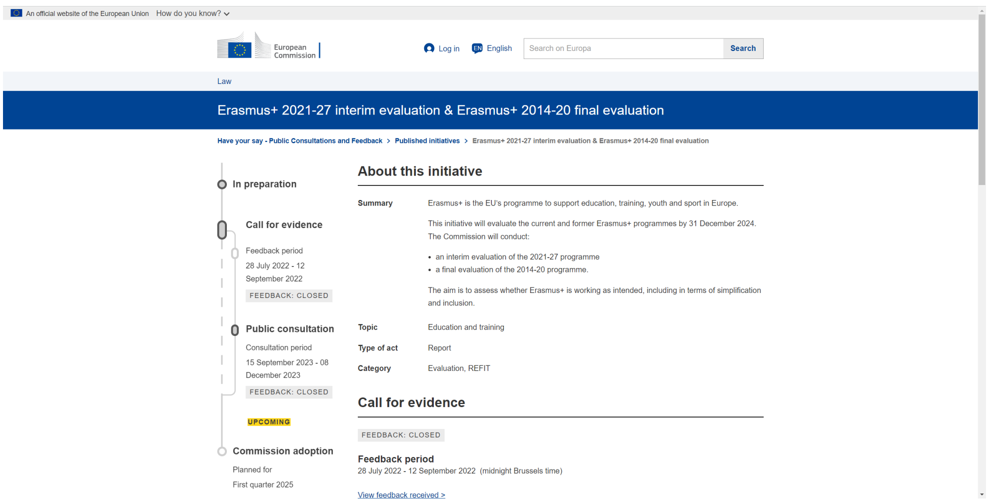

Figure 8: Website of the initiative *Erasmus+ 2021-27 interim evaluation & Erasmus+ 2014-20 final evaluation.*

8 Evaluation

In this section we present a online evaluation of the developed *Citizen Feedback Enhancer*, which mainly collects user feedback in the form of questionnaires [1]. The questionnaire includes a survey of the interviewees' educational and professional backgrounds, a Likert scale for evaluating the application feather, and several open-ended questions. This evaluation was carried out using a targeted release, and a total of about twenty feedbacks from different educational and professional backgrounds were collected. Below is a summary and analysis of each question in the questionnaire.

(1) **Demographic information:** In this part, we asked three questions about the user's basic information. The following are the specific statistics and analysis for each evaluation item:

- **Please state your age**: The respondents were aged between 24 and 50, with a concentration in the 25 to 28 age group.
- **Please state your highest degree**: According to the statistics on educational background, most of the respondents have a Master's degree or higher and only three of them own a bachelor's degree (including those studying for a master's degree), indicating that this group of users has a relatively high level of education.
- **What is your profession**: In terms of occupation, the respondents come from a wide range of professions, including professionals such as Masters and PhD students, academics, researchers, AI engineers and psychotherapists. It is worth noting that most of the respondents' professional fields are related to technology, research and data analysis, which enables them to evaluate the system's functions and performance from a professional perspective.

(2) **Likert Scale Ratings:** In this part, we have designed a Likert scale to evaluate users' experience with the *Citizen Feedback Enhancer*. The Likert

[1] `https://forms.gle/kzRfps7ZnoDD85tJ7`

scale contains ten questions, each with five levels of degree, namely “strongly disagree”, “disagree”, “neutral”, “agree” and “strongly agree”. The following are the specific statistics and analysis for each evaluation item:

- **I think that I would like to use this system frequently**: Most users agree with this statement, indicating that they are willing to use the system frequently. However, a minority of users still express neutrality or disagreement, indicating that they have a lower level of reliance on this e-democracy tool or that some functions have not fully met their needs.
- **I found the system unnecessarily complex**: Most users strongly disagree or disagree that the system is too complicated, indicating that the system is generally easy to use. A small number of users are neutral on this issue, which may indicate that some functions of the system are difficult to get used to.
- **I thought the system was easy to use**: Users generally agree that the system is easy to use, which reflects that the user interface and interaction design are satisfactory to most people.
- **I think that I would need the support of a technical person to be able to use this system**: Most users are disagreed to this, indicating that the system is user-friendly for new users and can be used without extensive learning.
- **I found the various functions in this system were well integrated**: Almost all users agree with this statement, indicating that the system is highly complete and very functional. This allows users to seamlessly switch between and use different functions.
- **I thought there was too much inconsistency in this system**: Half of the users disagreed with this statement and felt that the system was relatively consistent. However, forty percent of the users were neutral, indicating that in some cases they expected the results from the application to be inconsistent with their expectations.
- **I would imagine that most people would learn to use this system very quickly**: Most users agree with this statement, indicating that they find the application easy to use and able to adapt to different user groups.
- **I found the system very cumbersome to use**: The vast majority of users disagree with this statement, indicating that the application is user-friendly in most cases.

- **I felt very confident using the system**: Most users expressed confidence in using the system, indicating that when the system provides a verifiable source of information, users are more confident in the reliability and accuracy of the results.

- **I needed to learn a lot of things before I could get going with this system**: Most users were against this, indicating that the system is user-friendly for new users. However, some users still believe that they need to learn some operating skills to use the system well, which suggests that we may need to provide more user guides or help documents.

(3) **What is your favorite feature of Citizen Feedback Enhancer? Why?**: User feedback on this issue are varied, and we have summarised them as follows for elaboration and analysis:

- **System professionalism:** The professionalism of the system was affirmed by many users. User 1 said: *'Quickly identifying opinions on EU legislations.'* In addition, User 2 mentioned: *'My favorite feature is its professionalism and relevance. It is related to the EU laws and initiatives.'* This shows that users appreciate the professionalism and convenience of the system in answering questions related to specific citizen feedback and policy analysis.

- **Source visualization and credibility enhancement:** Several users mentioned that their favourite feature was the source information provided by the system when generating an response. For example, User 3 said, *'I appreciate that the system provides the sources it relies on to answer my questions, as it adds support to the response.'* User 4 added, *'It show you the information source. Because in this case, I can trust your system.'* Users felt that this feature was particularly important in policy analysis, as they could verify the accuracy of the system based on the information source.

- **Topic selection and key words search:** Some users particularly appreciated the system's topic selector and search function. This function allows users to narrow down the scope of the data according to their interests and needs in order to obtain more accurate feedback. For example, User 5 pointed out: *'The various topics make it useful for different people.'* Similarly, User 6 emphasised the convenience of keyword search: *'I can use the keyword search and get summarization.'* This feedback reflects the users' need for accurate data retrieval and their hope that the system can locate problems quickly and accurately.

(4) **Which function of Citizen Feedback Enhancer do you think is unnecessary? Why?**: User feedback on this issue is relatively concentrated, and similarly, we have summarised and analysed the issue as follows:

- **Technical parameter adjustment function:** Many users mentioned that adjusting technical parameters (such as generating model selection and retrieving technology types) is not necessary for the average user. User 7 believes that *'I assume the audience using this system isn't made up of experts who know the differences between the models available. So, it would be best to use the top model for the system and remove the model selection option from the settings.'* This feedback indicates that for most users with a non-technical background, adjustments to technical parameters add complexity to the system. They would prefer the system to automatically adjust these settings and operate efficiently without the need for manual and complex parameter adjustments.

- **Scoring function:** Some users pointed out that the scoring function of the system is complex and difficult to understand. Although the original intention of this feature was to allow the AI to evaluate the answers it generated for the user's reference, many users reported being confused by the lack of explanation. User 8 mentioned: *'The relevancy and alignment scores on the left are a nice idea, but as they are very intrans parent I would not rely on them.'* Similarly, User 9 added, *'I do not really get the scores on the right? Is the system rating its own answer?'* This feedback indicates that the current scoring function is not intuitively explained, and users have difficulty understanding the actual meaning and application scenarios of these scores. Therefore, adding an explanation of the scoring mechanism or simplifying the scoring display may improve the user experience.

(5) **Do you experience any problems during use?**: The aim of this question is to collect operational problems and difficulties encountered by users when using the application, so as to better understand the shortcomings of the system. Through user feedback, we found that problems mainly focus on the following two points.

- **System response speed and operation problems:** Many users mentioned that the response speed of the system was sometimes slow and that they had to wait a long time for answers to be generated. For example, User 10 mentioned: 'I should wait the answer for several times.' In addition, some users also said that the system were not convenient enough. User 11 pointed out: 'You need to press the button and can not simply hit enter (small remark).' The feedback shows that system response speed and convenience are the key factors affecting user experience and will be the focus of future optimisation.

- **Display problems:** Several users mentioned that the system's replies were too long. User 12 mentioned: *'The chatbot's reply is just toooo long. No one wants to read such a long paragraph. The mobile experience is not optimized.'* This is because the system requires as much detail and completeness as possible when generating replies, which leads to the generated content is too long, which affects the user's reading experience. User 13 added: *'When reading the answers, I felt that the spacing between the words was inconsistent, and the reading experience was average.'* This feedback indicates that users want the system's responses to maintain a fixed word spacing for a better reading experience.

(6) **What additional features would you like to see in future versions of the web app to improve the overall efficiency of feedback analysis?**: All users provided valuable feedback to help us improve the application. The following suggestions for improving the functionality of future versions of the application have been analysed and summarised.

- **Improve response speed and streaming output:** Response speed was a focus for many users. User 14 suggested *'I think it would save time if the answers could be streamed out, like popular chatbot does.'* to reduce waiting time and allow users to see the system's working process in real time. There were also other similar suggestions to output responses as if they were typed. This shows that using streaming output when generating long text is a key future improvement.

- **Structured answers and concise expressions:** Many users wish the system-generated answers to be more concise and structured. User 15 said: *'This app seems to design for professionals in policy areas, but I suggest to make answer look more organized, such as list each points or highlight important part.'* In fact, generating long responses that are entire paragraphs seriously affects the user's reading experience. This shows that users prefer answers with a clearer structure. Future optimisation plans will include adjusting the output format and displaying key content to make it easier for users to read and capture key information.

- **Optimise UI/UX design:** Users have made several suggestions for improving the UI/UX design. User 16 suggested, *'Maybe add an info icon for the "more functions are here" section to explain the features a bit, improving the user experience (UX).'* This indicates that some unnecessary text descriptions affect the user experience, and some of the text displayed in the application will be replaced with images. Secondly, some users suggested adjusting the layout of the built-in filtering and key word search function. Due to their location, these functions are often overlooked, and it needs to be opened and closed each time, which is inconvenient to use. This indicates that users want to improve the interactivity of the system and hope to get a more relaxed and convenient interactive environment.

- **Background information supplement:** Many users suggested that the system add more filtering options, such as filtering feedback content by date, region, etc., to more accurately retrieve and analyse citizen feedback. And some users wanted more detailed policy context explanations. User 17 said, *'I don't know much about EU laws and regulations, so I would like to have a section that explains the current policy status before displaying citizens' feedback.'* This suggestion prompts us to consider the wider user base, so providing some explanation of the policy background is necessary.

9 Discussion

This thesis aims to fill the gap in the Have Your Say platform by integrating advanced NLP techniques to provide an efficient method for analysing user feedback [9]. The thesis focuses on the application of advanced NLP techniques such as information retrieval, text embedding and text generation [3, 7], and data-driven web application development. Following the DSRM methodology [27], the thesis begins by identifying a central research question and formulating four key design points around this question for which appropriate solutions are sought. Guided by the design principles, we developed a RAG-based web application [20], the *Citizen Feedback Enhancer*, and demonstrated and evaluated the application's functionality and future prospects. Thus, the core research question proposed in this thesis - *How can a system be designed that effectively incorporates citizen feedback from the Have Your Say into the policy making process?* - was effectively answered.

First, as more and more e-participation platforms are established to promote the development of digital democracy, many NLP methods have been gradually applied to these platforms [6, 37, 31]. For example, common applications include text classification, sentiment recognition, and interactive systems based on chatbots. However, these NLP applications are usually single-function and mainly used to improve the efficiency of interaction between the public and the platform, and rarely directly serve the needs of policy makers. In addition, although RAG has received widespread attention in the NLP field in recent years, it has not been fully promoted and used on citizen engagement platforms. For large-scale citizen feedback platforms such as Have Your Say, there are no examples of the use of RAG technology or similar advanced tools to efficiently process and analyse feedback. By developing the Citizen Feedback Enhancer and applying RAG technology to citizen engagement platforms, we have contributed to the application of this technology on the platform.

Second, when defining the research questions, we divided the core questions into four sub-questions. At the same time, each sub-question was analysed in depth to formulate feasible design principles. During the system design process, we specifically considered how different artefacts could affect the effectiveness of solving these problems. For example, to solve the problem of interactivity of feedback data, we designed the web application as an interactive chatbot and added many features to display additional information to enhance interactivity. To solve the problem that feedback data cannot be efficiently analysed and evaluated, we integrated text embedding technology with state-of-the-art text generation technology so that semantic information in textual data can be efficiently extracted and analysed. Furthermore, the retrieval technology can simultaneously search for multiple data related to the query, which greatly improves the efficiency of data use. The application of these technologies can be extended to other similar citizen participation platforms, replacing to some extent the manual analysis of large amounts of feedback text and improving the efficiency and accuracy of the decision-making process.

Third, we followed the several design principles of digital platform and web application during the development process to ensure that the system not only solves the core problem, but also has the ability to evolve in a sustainable way. Firstly, the application should be user-centred. By analysing the needs of stakeholders, we designed an intuitive and easy-to-use interface that allows users to conveniently analyse feedback and access information. Moreover, the web application is data-driven. All data comes from the actual feedback content and other related features of the target website and is stored in MongoDB database. The system accesses the database via several API to retrieve and search for data. To ensure the sustainability and scalability of the system, we have adopted a modular system architecture, which allows the system to adapt and expand flexibly in the face of future technological advances and changes. For the data in the database, the system is regularly updated via GitHub Actions to ensure that the data is up to date. Privacy and security are also key concerns when dealing with large amounts of citizen feedback. The system only uses known data from the target website and does not use personal information such as name and gender as features. Collaboration and openness are also important aspects of the design. All code is open source and available in a public repository on GitHub [1]. Finally, the entire project uses an iterative development approach to gather user feedback and make continuous improvements. During the evaluation process, the focus is on optimising the user experience and system functionality. The above design principles have enabled *Citizen Feedback Enhance* to have stable functionality and flexibility. It not only efficiently processes and analyses large volumes of citizen feedback from the Have Your Say platform, but also has the capacity for continuous improvement and expansion.

However, after analysing the user experience evaluation, it was found that the *Citizen*

[1] `https://github.com/Xiaohw1999/eufeedbackapp.git`

Feedback Enhancer still had some limitations in its design. The response time of the system was sometimes slow, which affected the overall user experience. Although the system could process data effectively, the lack of streaming output for instant feedback remained a challenge that needed to be addressed. Furthermore, some features (such as relevance and consistency ratings) were considered complicated and difficult to understand. Users indicated that the system did not provide sufficient explanations for these features, which made the experience of using them less than smooth. In addition, users suggested reducing the number of adjustable technical parameters (e.g. LLMs settings) as these options were not necessary for the target users. Users would prefer the system to automatically select the most appropriate algorithm, rather than having to select it themselves. There were also some minor issues with the layout of the main interface, such as the search function, which should be placed more prominently, rather than hidden in the sidebar, so that it is easier for users to find and use. These issues affected the overall user experience of the system and suggest that these aspects need to be further optimised in future improvements.

10 Conclusion

Due to the large amount of citizen feedback on the Have Your Say website and the lack of effective digital methods for automating and analysing this feedback, the key information in the large amount of feedback has not been used effectively. To address this problem, this thesis develops a *Citizen Feedback Enhancer* using the DSRM methodology [27]. This web application is based on the RAG system and combines state-of-the-art retrieval technology with text generation technology to quickly retrieve users' concerns from a large feedback database and generate complete and accurate responses. The application successfully applies RAG technology to digital democracy platforms, improving the efficiency of using citizen feedback and helping policy makers to better understand and respond to public needs.

The core contribution of this thesis is the development of a RAG-based web application —— *Citizen Feedback Enhancer*. We used DSRM to identify the problem, explore possible solutions and complete the development, deployment and evaluation of the web application. This application effectively compensates for the shortcomings of the Have Your Say website in terms of inefficient use of user feedback [9]. Through this application, policy makers will be able to extract key information from user feedback more quickly and accurately, thereby improving the efficiency of policy making. This contribution has a positive impact in several areas. First, the application enables policy departments to efficiently analyse and synthesise public opinion. Secondly, it improves the user experience and allows for better incorporation of citizen feedback, thereby increasing the effectiveness of citizen participation. In addition, the app provides a practical example of the application of advanced retrieval generation modelling technology in public affairs, and is expected to promote innovation in digital democracy technology.

However, our research has some limitations. First, there was insufficient in-depth communication with the core user group when defining the research questions, and no comprehensive needs analysis was conducted through interviews or workshops. Although the target platform has yet to adopt advanced technologies like RAG for processing user feedback, conducting a thorough needs study will be essential for future iterations. Second, there are limitations in data processing. Since much of the feedback is unstructured and subjective, and lacks standard answers, the diversity and complexity of the data hinder the feasibility of extensive pre-processing. Over-processing may weaken important information and reduce the accuracy of the analysis. Lastly, the evaluation process is constrained by the lack of standardized metrics to assess the system's effectiveness. Because the evaluation mainly relied on user experience and satisfaction to the system's functionality and user interface. As a result, these evaluations are more likely to be subjective. More objective metrics are therefore needed in the future to assess the accuracy and efficiency of the system.

In summary, this thesis finds that there is great potential for the introduction of LR and LLM techniques into the citizen feedback analysis process. Through the development of the *Citizen Feedback Enhancer* web application, this research demonstrates how LR and LLM techniques can be used to quickly retrieve and analyse citizen feedback to provide policy makers with an accurate basis for decision making. In future work, there are several directions in which the performance and scope of the Citizen Feedback Enhancer can be further improved. First, we can better understand the specific needs of core users through in-depth stakeholder needs analysis. Second, we should also focus on collaboration between system functions, reduce cumbersome user operations and improve the user experience. In addition, the scalability of the system is also an important development direction. Currently, the main NLP technologies used in the system are information retrieval and text generation. In the future, technologies such as sentiment analysis will be introduced to provide users with richer analysis perspectives. I believe that these suggestions will further improve the user experience of web applications and provide more convenient support for feedback analysis to policy makers.

References

[1] E. Adamopoulou and L. Moussiades. An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations*, pages 373–383. Springer, 2020.

[2] Amazon. Amazon web services (aws). `https://aws.amazon.com/`, 2023.

[3] D. Analytics and D. Sukhwal. Retrieval augmented generation: An evaluation of rag-based chatbot for customer support. 2024.

[4] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi. The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges. *Future Internet*, 15(8):260, 2023.

[5] Q. Chen, Y. Gong, Y. Lu, and J. Tang. Classifying and measuring the service quality of ai chatbot in frontline service. *Journal of Business Research*, 145: 552–568, 2022.

[6] CitizenLab. Citizenlab - digital democracy made easy. `https://www.citizenlab.co`, 2015. Accessed: 2024-09-23.

[7] J. Duberry. *Artificial Intelligence and Democracy*. Edward Elgar Publishing, 2023. URL `https://creativecommons.org/licenses/by-nc-nd/4.0/`. Open Access, licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.

[8] EU Monitor. European commission revamps its 'have your say' portal - eu monitor. `https://www.eumonitor.eu/9353000/1/j9vvik7m1c3gyxp/vla1g0soroqw?ctx=vjyqenw8m1ig`, July 3 2020.

[9] European Commission. European commission - have your say. https://ec.europa.eu/info/law/better-regulation/have-your-say_en, 2024.

[10] I. Facebook. React: A javascript library for building user interfaces. `https://react.dev/`, 2023.

[11] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech. Generative ai. *Business & Information Systems Engineering*, 66(1):111–126, 2024.

[12] Figma. Figma: The collaborative interface design tool. `https://www.figma.com/`, 2023.

[13] P. Finardi, L. Avila, R. Castaldoni, P. Gengo, C. Larcher, M. Piau, P. Costa, and V. Caridá. The chronicles of rag: The retriever, the chunk and the generator. *arXiv preprint arXiv:2401.07883*, 2024.

[14] F. Fui-Hoon Nah, R. Zheng, J. Cai, K. Siau, and L. Chen. Generative ai and chatgpt: Applications, challenges, and ai-human collaboration, 2023.

[15] A. Fung. Varieties of participation in complex governance. *Public Administration Review*, 66(s1):66–75, 2006.

[16] M. Grohe. word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–16, 2020.

[17] J. Haase and P. H. Hanel. Artificial muses: Generative artificial intelligence chatbots have risen to human-level creativity. *Journal of Creativity*, 33(3): 100066, 2023.

[18] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

[19] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[20] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[21] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[22] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*, pages 2391–2400. PMLR, 2017.

[23] I. MongoDB. Mongodb atlas: Cloud database as a service. https://www.mongodb.com/atlas, 2023.

[24] Netlify. Netlify: Build, deploy, and manage modern web projects. https://www.netlify.com/, 2023.

[25] OpenAI. Chatgpt: A large language model. https://chat.openai.com/, 2023. Accessed: 2024-09-26.

[26] C. Pateman. *Participation and Democratic Theory*. Cambridge University Press, 1970.

[27] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[28] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[29] S. Ramírez. Fastapi: Fastapi framework, high performance, easy to learn, fast to code, ready for production. https://fastapi.tiangolo.com/, 2018.

[30] S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389, 2009.

[31] P. Savaget, T. Chiarini, and S. Evans. Empowering political participation through artificial intelligence. *Science and Public Policy*, 46(3):369–380, 2019.

[32] O. Shahmirzadi, A. Lugowski, and K. Younge. Text similarity in vector space models: a comparative study. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 659–666. IEEE, 2019.

[33] A. Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[34] K. Sprenkamp, L. Zavolokina, M. Angst, and M. Dolata. Data-driven governance in crises: Topic modelling for the identification of refugee needs. In *Proceedings of the 24th Annual International Conference on Digital Government Research*, pages 1–11, 2023.

[35] Sæbø, J. Rose, and L. S. Flak. The shape of eparticipation: Characterizing an emerging research area. *Government Information Quarterly*, 25(3): 400–428, 2008.

[36] D. Team. Decidim - free open-source participatory democracy for cities and organizations. `https://decidim.org`, 2016. Accessed: 2024-09-23.

[37] Y. Theocharis and J. W. Van Deth. The continuous expansion of citizen participation: A new taxonomy. *European political science review*, 10(1): 139–163, 2018.

[38] M. UI. Material ui: React components for faster and easier web development. `https://mui.com/`, 2023.

[39] I. Van de Poel. Embedding values in artificial intelligence (ai) systems. *Minds and machines*, 30(3):385–409, 2020.

[40] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[41] J. Xian, T. Teofili, R. Pradeep, and J. Lin. Vector search with openai embeddings: Lucene is all you need. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1090–1093, 2024.

[42] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, and B. Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

Universität Zürich[UZH]

Institut für Informatik

Declaration of independence for written work

I hereby declare that I have composed this work independently and without the use of any aids other than those declared (including generative AI such as ChatGPT). I am aware that I take full responsibility for the scientific character of the submitted text myself, even if AI aids were used and declared (after written confirmation by the supervising professor). All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Zürich, Xiaohui Wu

____

Signature of student