# CSIT6000R Group 8 Project Report

**SHAO Guocheng**
SHEN Hengshuo
LIU Zhongnuo
{gshao,}@connect.ust.hk

## Abstract

Conversational AI systems have witnessed significant advancements in recent years, with Retrieval-Augmented Generation (RAG) emerging as a promising approach for question-answering tasks. RAG systems combine retrieval mechanisms with generative models to provide contextually relevant responses by leveraging external knowledge sources. In this project, we present a detailed exploration of RAG-based question-answering pipelines, focusing on key components such as document retrieval, response generation, and conversation memory management. We investigate different pipeline configurations and evaluate their performance using diverse evaluation methodologies, including the use of large language models as judges. Our study sheds light on the strengths and limitations of RAG systems and provides insights into their design, implementation, and evaluation. Through comprehensive experimentation and analysis, we aim to advance the understanding of RAG-based conversational AI systems and contribute to the ongoing development of effective question-answering solutions.

## 1 Introduction

Conversational AI systems have witnessed remarkable advancements in recent years, empowering users to interact with machines in natural language and facilitating a wide range of applications, from customer service chatbots to virtual assistants. Among the various approaches to conversational AI, Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm for question-answering tasks, combining the strengths of both retrieval and generation models to provide informative and contextually relevant responses.

In this paper, we present a comprehensive exploration of RAG-based question-answering pipelines, focusing on the integration of document retrieval, response generation, semantic filtering and conversation memory mechanisms. Our study aims to elucidate the intricate workings of RAG systems, evaluate their performance across different configurations, and provide insights into their effectiveness in handling diverse user queries and scenarios.

The proliferation of large-scale language models, such as GPT and BERT, has significantly contributed to the advancement of conversational AI by enabling more sophisticated natural language understanding and generation capabilities. However, while these models excel in generating fluent and coherent text, they may lack access to external knowledge sources, limiting their ability to provide accurate and contextually relevant responses to user queries.

RAG addresses this limitation by augmenting generation models with retrieval mechanisms that enable access to external knowledge repositories, such as document collections or knowledge graphs. By leveraging the complementary strengths of retrieval and generation models, RAG systems can effectively combine pre-existing knowledge with generative capabilities to produce informative and contextually relevant responses.

In this project, we delve into the design and implementation of RAG-based question-answering pipelines, exploring key components such as document retrieval mechanisms, response generation strategies, and conversation memory management. We conduct an evaluation of different pipeline configurations, assessing their performance in terms of response quality, relevance, and coherence.

In summary, our work offers a comprehensive exploration and experiment of RAG-based question-answering pipelines, highlighting their potential for enhancing user interactions and providing valuable insights into their design, implementation, and evaluation.

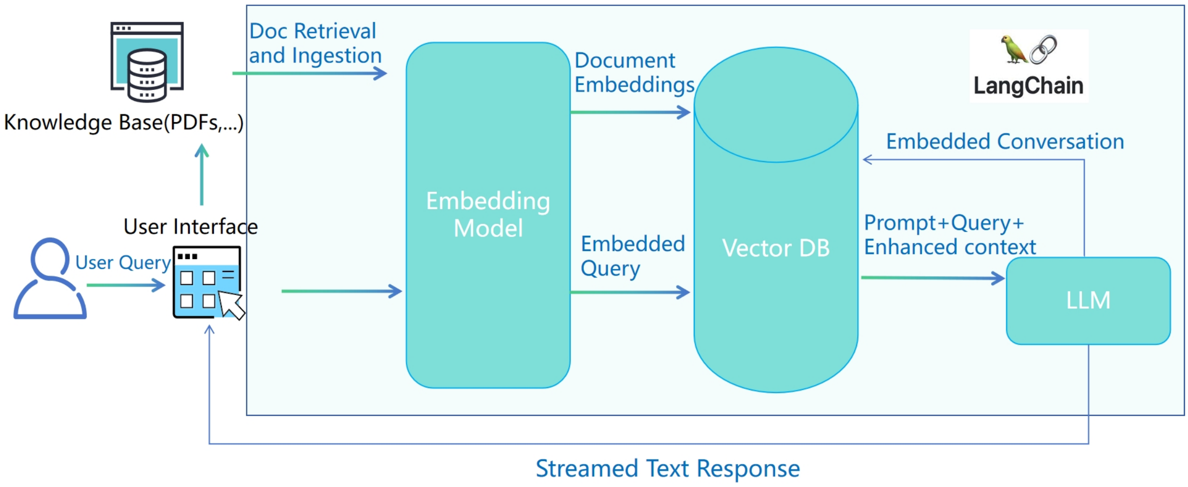


Figure 1: Overall framework

## 2 Related Works

The development of Retrieval-Augmented Generation (RAG) systems for question-answering tasks has garnered significant attention in the field of conversational AI. Researchers have explored various approaches to integrating retrieval and generation models to enhance the quality and relevance of responses. Below, we provide an overview of related work in this domain.

- RAG Models: (Lewis et al., 2020) introduced the RAG framework, which combines a retriever module for document retrieval with a generator module for response generation. This work laid the foundation for subsequent research in RAG-based question-answering systems.

- Dual Encoder Models: (Karpukhin et al., 2020) proposed the use of dual encoder models, where both the query and the document are encoded separately before being fed into a decoder for response generation. This approach improves the alignment between queries and documents, leading to more accurate responses.

- Conversation Memory: In the context of conversational question-answering, (alice. publications@ cern. ch et al., 2021) explored the use of conversation memory mechanisms to maintain context across multiple turns of dialogue. By integrating conversation history into the response generation process, these systems can produce more coherent and contextually relevant responses.

- Large Language Models: Recent advancements in large language models, such as GPT-3 (Brown et al., 2020), have further propelled research in conversational AI. These models offer powerful generation capabilities and can be fine-tuned for specific question-answering tasks, either independently or as part of RAG pipelines.

## 3 Methods

In this section, we provide a detailed overview (Fig.1) of the methodologies employed in our study to explore Retrieval-Augmented Generation (RAG) pipelines for question-answering tasks. We delve into the design and implementation of key components such as document retrieval, response generation, and conversation memory management. Our methodologies aim to elucidate the intricate workings of RAG systems and evaluate their performance across different configurations. Through rigorous experimentation and analysis, we seek to gain deeper insights into the effectiveness of RAG pipelines in providing informative and contextually relevant responses to user queries.

### 3.1 Embedding Documents and Storage

**Embedding Documents** In this subsection, we detail the process of embedding documents using the NVIDIA embeddings model. The NVIDIA embeddings model serves as a powerful tool for converting textual information into dense vector representations, capturing semantic meaning and contextual information within the document content. These embeddings enable efficient comparison and retrieval of documents based on semantic similarity.

Our approach involves leveraging the pre-trained NVIDIA embeddings model, specifically tailored for question-answering tasks. This model is trained on a large corpus of text data, allowing it to capture intricate relationships and nuances present in natural language. By utilizing this model, we transform each document into a high-dimensional embedding space, where documents with similar semantic content are represented by vectors located closer together.

**Document Storage** In parallel with document embedding, we employ the FAISS (Facebook AI Similarity Search) index for efficient storage and retrieval of documents. The FAISS index facilitates fast and scalable similarity search operations on large collections of high-dimensional vectors. It organizes document embeddings into a data structure optimized for nearest neighbor search, enabling rapid retrieval of relevant documents given a user query.

Our document storage system integrates the FAISS index with an in-memory document store, allowing seamless integration of newly embedded documents into the existing collection. This ensures that our document storage remains dynamic and adaptive to changes in the document corpus over time.

By combining document embedding with the FAISS index and in-memory document storage, we establish a robust foundation for effective document retrieval in our question-answering pipeline. This enables our system to efficiently process user queries and retrieve relevant documents based on semantic similarity, facilitating accurate and informative responses to user inquiries.

This detailed explanation highlights the technical intricacies of our document embedding and storage methodology, showcasing the sophistication and effectiveness of our approach within the context of question-answering research.

### 3.2 Document Retrieval

In this subsection, we elucidate the process of document retrieval, which forms a crucial component of our question-answering pipeline. Our document retrieval mechanism aims to efficiently retrieve relevant documents from the document corpus based on user queries or uploaded documents.

We leverage the FAISS index, integrated with our in-memory document store, to perform fast and accurate document retrieval. When a user submits a query or uploads a document, our system utilizes the embedded representations of documents stored in the FAISS index to identify documents that are semantically similar to the query or uploaded content. This process involves conducting nearest neighbor search operations in the high-dimensional embedding space to identify candidate documents for further processing.

Additionally, we employ techniques such as long context reordering and text splitting to enhance the effectiveness of document retrieval. Long context reordering ensures that longer documents are processed effectively by prioritizing the central portions of the text, where the most relevant information is often located. Text splitting enables the segmentation of lengthy documents into smaller, manageable chunks, facilitating more granular retrieval and analysis.

### 3.3 Response Generation and Conversation Memory

By combining retrieval-based and generative approaches, our system ensures comprehensive and contextually relevant responses. This strategy enables the system to provide accurate information while maintaining flexibility in response generation.The generative approach entails:

- Contextual Analysis: The system analyzes user queries, retrieved documents, and conversation history to comprehend context and information requirements thoroughly.
- Response Formulation: Leveraging Code Llama 13B from NVIDIA's API, the system formulates responses that are fluent, coherent, and informative. It synthesizes information from multiple sources to craft responses that address user queries effectively.

**Conversation Memory** Our system maintains a conversation memory to retain contextual information from previous interactions, enabling personalized and coherent responses over time. This memory stores key details such as user queries, system responses, timestamps, and relevant document excerpts, facilitating continuity and coherence in the conversation. The context retention process involves the following steps:

- Data Storage: Contextual information from each interaction is stored in the conversation memory, ensuring that past exchanges are readily accessible for reference.

- Contextual Analysis: When generating responses to new queries, the system analyzes the conversation history stored in the memory to understand the context and tailor the response accordingly.

- Contextual Integration: Relevant information from past interactions is seamlessly integrated into the response generation process, enhancing the relevance and coherence of the generated responses.

By leveraging the conversation memory, our system develops a long-term contextual understanding of user preferences, information needs, and conversational dynamics. This understanding evolves over time as the system accumulates knowledge and experience from past interactions. The long-term contextual understanding enables the system to adapt its response generation strategies dynamically, improving user engagement and satisfaction over time.

### 3.4 Semantic Guardrail

**Classifier Training** In this subsection, we detail the process of training the semantic guardrail classifier (Fig.2), which plays a pivotal role in assessing the quality of user queries before generating responses. The classifier is trained to distinguish between high-quality queries suitable for response generation and low-quality queries that may be irrelevant or potentially harmful.

**Dataset Construction** We curate a labeled dataset comprising a diverse range of user queries, categorized based on their suitability for answering by a document chatbot. Queries are annotated with labels indicating whether they are deemed suitable or unsuitable for response generation.

- Good Responses Dataset: This dataset includes queries that would be reasonable for a document chatbot to answer. Queries cover various topics such as technology, research, gaming, language modeling, and graphics. Each query is representative of a different context or phrasing to ensure diversity.

- Poor Responses Dataset: Conversely, this dataset consists of queries that would be unreasonable for a document chatbot to answer. These queries are either irrelevant or potentially harmful to the reputation of owner of the chatbot.

**Model Selection and Training** Given the requirement for inference efficiency, we opt for a Multilayer Perceptron (MLP) as the classifier architecture. MLPs are computationally efficient and well-suited for text classification tasks.

- Model Architecture: The MLP classifier comprises multiple fully connected layers, with non-linear activation functions applied between layers to capture complex patterns in the data.

- Training Process: The labeled datasets are split into training, validation, and test sets to train and evaluate the classifier's performance. We employ techniques such as mini-batch gradient descent and backpropagation to optimize the model parameters.

- Hyperparameter Tuning: We fine-tune hyperparameters such as learning rate, batch size, and number of hidden units in the MLP to maximize classification performance while ensuring computational efficiency.

**Guardrail Integration** Once trained, the semantic guardrail classifier is integrated into our question-answering pipeline to assess the quality of user queries in real-time. When a user submits a query, the classifier evaluates its suitability for response generation based on learned patterns and features. Queries deemed to be of high quality are forwarded to the response generation component of the pipeline, while low-quality queries are flagged for further consideration or filtering.

The semantic guardrail acts as a proactive measure to ensure that only relevant and informative responses are generated in response to user queries. By filtering out low-quality queries before response generation, we enhance the overall quality and relevance of the responses provided by our question-answering system, improving user satisfaction and usability.

This detailed explanation illuminates the development and application of the semantic guardrail within our question-answering pipeline, highlighting its role in enhancing the quality and effectiveness of our system.

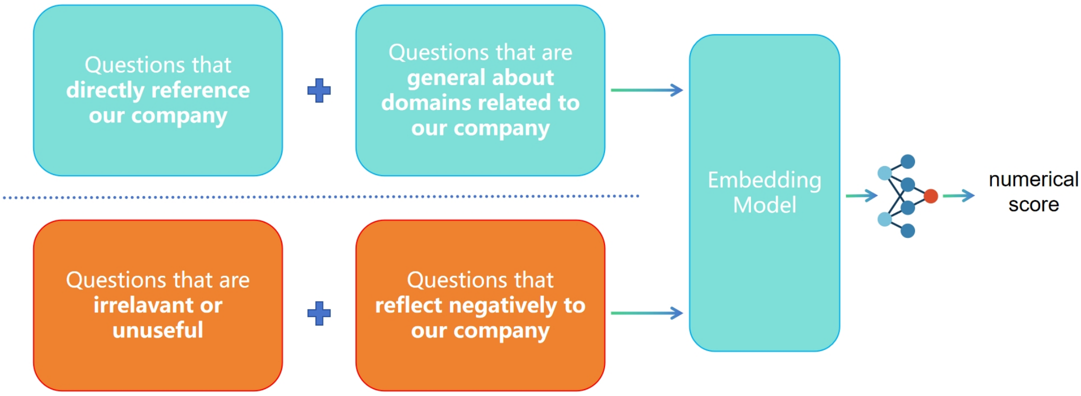


Figure 2: Semantic filtering classifier

### 3.5 User Interface

Our system leverages the Gradio library to develop an intuitive and user-friendly interface for interacting with the question-answering pipeline. Gradio provides a seamless way to create web-based interfaces for machine learning models, allowing users to input queries, upload documents, and receive responses in real-time. Through the Gradio interface, users can effortlessly engage with the system, making it accessible to both experts and non-experts alike.

**Document Upload Handling** In addition to text queries, our system supports document upload functionality, allowing users to submit PDF documents for analysis and question-answering.

- PDF Content Parsing: Upon receiving a document upload, our system parses the PDF content, extracting text and metadata information. This process involves segmenting the document into smaller units if necessary, ensuring efficient processing and analysis.

- Integration into Document Corpus: The parsed content is then integrated into the document corpus, where it becomes available for subsequent retrieval and analysis alongside other documents. This seamless integration enhances the comprehensiveness and relevance of information available for question-answering tasks.

By seamlessly integrating document retrieval with user input processing, our system provides a holistic solution for addressing user inquiries effectively. This methodology enables users to interact with the system using both text queries and document uploads, enhancing the accessibility and usability of the question-answering pipeline.

## 4 Evaluation

### 4.1 LLM as a Judge Formulation

In the realm of conversational AI, leveraging Large Language Models (LLMs) as evaluators or "judges" has emerged as a valuable approach for configurable automatic testing of natural language task performance. This approach involves using an LLM to simulate various interaction scenarios, generate synthetic data, and evaluate the performance of a chatbot or conversational system. By parsing the chatbot's responses on synthetic data, developers can assess its correspondence, retrieval accuracy, and overall performance against predefined criteria.

This concept, known as "LLM-as-a-Judge," allows for easy test specifications aligned closely with human judgment and can be fine-tuned and replicated at scale. Rather than using pre-existing LLM chains, we aim to expand on these ideas and evaluate our system with a more customized solution tailored to our specific requirements. Our pairwise evaluation process involves the following steps:

1. Sample two document chunks from the RAG agent's document pool.

2. Generate a synthetic "baseline" question-answer pair using the sampled document chunks.

3. Utilize the RAG agent to generate its own answer based on the provided question and document context.

4. Use an LLM as a judge to compare the two responses, considering the synthetic generation as "ground-truth correct."

By implementing this pairwise evaluator, we can quantitatively evaluate the performance of our RAG pipeline and gain insights into its ability to provide accurate and relevant responses compared to baseline chatbot systems.

An example Q&A pair is shown as following:

**Question**: How do latent diffusion models (LDMs) improve the visual fidelity of high-resolution image synthesis compared to pixel-based diffusion models (DMs)?

**RAG Answer**: According to the paper "High-Resolution Image Synthesis with Latent Diffusion Models," LDMs improve the visual fidelity of high-resolution image synthesis compared to pixel-based DMs in several ways. Firstly, LDMs are trained in the latent space of powerful pretrained autoencoders, which allows them to reach a near-optimal point between complexity reduction and detail preservation. This is in contrast to previous work that trained diffusion models directly in pixel space, which often resulted in a trade-off between the two.

Secondly, LDMs introduce cross-attention layers into the model architecture, turning them into powerful and flexible generators for general conditioning inputs such as text or bounding boxes. This convolutional manner of high-resolution synthesis is made possible by the use of latent spaces, which enables LDMs to generate high-quality images while significantly reducing computational requirements compared to pixel-based DMs.

Lastly, the paper notes that LDMs trained in VQ-regularized latent spaces sometimes achieve better sample quality, even if the reconstruction capabilities of the first stage models slightly fall behind those of their continuous counterparts. This suggests that LDMs are able to generate higher quality images than pixel-based DMs, even with some loss in reconstruction capabilities.

In summary, LDMs improve the visual fidelity of high-resolution image synthesis compared to pixel-based DMs by training in the latent space of powerful pretrained autoencoders, introducing cross-attention layers, and achieving better sample quality in some cases.

An example LLM evaluation is presented below.

**Question**: How do latent diffusion models (LDMs) improve the visual fidelity of high-resolution image synthesis compared to pixel-based diffusion models (DMs)?

**Synth Evaluation**: [1] Justification: The second answer is more detailed and provides additional information about how latent diffusion models (LDMs) improve visual fidelity compared to pixel-based diffusion models (DMs). It explains the use of latent spaces, cross-attention layers, and the impact on computational requirements. Furthermore, it discusses the benefits of training LDMs in VQ-regularized latent spaces. The second answer does not introduce any inconsistencies and provides a better understanding of the comparison between LDMs and pixel-based DMs.

**Evaluation Instructions** The evaluators are instructed to compare the two answers and provide a score based on the following criteria:

[0]: The second answer is deemed inferior, contains false information, does not address the question adequately, or introduces inconsistencies.

[1]: The second answer is considered better than the first, maintaining consistency and providing accurate information.

We repeat the experiment for 10 times and take the frequency that the second answer(RAG answer) outperforms the default answer as a quantitative metric (see the results in Table. 1).

| RAG Pipeline | Preference Metric |
|---|---|
| w/o conversation retrieval | 0.8154 |
| w/o semantic guardrail | 0.7213 |
| Full pipeline | **0.8667** |

Table 1: Preference Metrics for Different RAG Pipelines

We have some reasoning for the result. For RAG without Conversation Retrieval, the absence of conversation retrieval capabilities limits the system's ability to contextualize responses based on previous interactions. Without access to past conversations, the system may struggle to maintain coherence and relevance in its responses, leading to lower preference metrics compared to the full RAG pipeline. Users may perceive the lack of contextual understanding as a limitation, resulting in lower preference ratings. While the absence of semantic guardrail in the RAG pipeline could lead to the inclusion of lower-quality or irrelevant questions in the conversation history. This could impact the performance of the system in the following ways:

- Inclusion of Bad Questions: Without the semantic guardrail, the RAG pipeline may not filter out questions that are irrelevant, inappropriate, or of poor quality. As a result, these questions could be included in the conversation history, affecting the relevance and coherence of subsequent responses.
- Degraded Conversation Context: The presence of bad questions in the conversation history could distort the contextual understanding of the system. Subsequent responses may be influenced by irrelevant or misleading information, leading to lower-quality interactions with users.
- Impact on User Experience: Users may become frustrated or disengaged if the system responds inadequately to their queries or introduces irrelevant topics into the conversation. This could result in lower satisfaction ratings and preference metrics for the RAG pipeline without the semantic guardrail.

And the full RAG pipeline incorporates both conversation retrieval and semantic guardrail components, enhancing the system's ability to generate high-quality responses. By leveraging conversation history and semantic constraints, the system can produce more coherent, relevant, and accurate responses, resulting in higher preference metrics.

### 4.2 Document retrieval Evaluation

We test several use cases to validate if the pipeline could correctly retrieve the document(s) relating to the question. An example test is presented below.

**User**: <Uploaded a PDF document 'Learning Transferable Visual Models From Natural Language Supervision'>

**Agent**: I have received your document 'Learning Transferable visual Models From Natural Language supervision'. I'm glad to help if you have any question regarding it.

**User**: So how to learn a visual model from natural language supervision in detail?

**Agent**:To learn a visual model from natural language supervision,you can follow the approach detailed in the paper"Learning Transferable visual Models From Natural Language supervision."The authors propose using raw text about images as a broader source of supervision compared to the traditional fixed set of predetermined object categories. First,create a sufficiently large dataset by collecting(image,text) pairs,as the authors did with a dataset of 400 million pairs from the internet.Then,use the simple pre-training task of predicting which caption goes with which image.This method has proven to be efficient and scalable for learning state-of.the-art (SoTA) image representations from scratch. After pre-training,fine-tune the model for specific tasks using labeled data.The authors demonstrate that this natural language supervision approach enables flexible zero-shot transfer,as the representation is connected to language. Additionally,consider exploring creative and advanced techniques such as dialog-based learning,semantic parsing,and feature representations produced by conditioning a deep contextual language model on natural language explanations and descriptions of relations.These methods can further improve performance on specific tasks.
Sources: Radford, A, Kim, J.W, Hallacy, C, Ramesh, A, Goh,G, Agarwal, S… &Sutskever,I.(2021). Learning Transferable Visual Models From Natural Language Supervision.arXiv preprint arXiv:2103.00020.

We could clearly see the agent retrieved the related document and refer to it for response.

### 4.3 Semantic Guardrail Evaluation

Experiments show that our agent with semantic guardrail mechanism can successfully handle different types of questions and be aware of the organization it represents(NVIDIA, in our case) after an efficient training, as introduced in Sec. 3.4. Our reason for generating embeddings from different groups of questions hinges on the assumption that they would be useful for semantic filtering. To help confirm this, we use some classical machine learning approaches like principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) for dimensionality reduction. These techniques essentially transform high-dimensional data into lower-dimensional representations while trying to keep the important statistical properties intact.

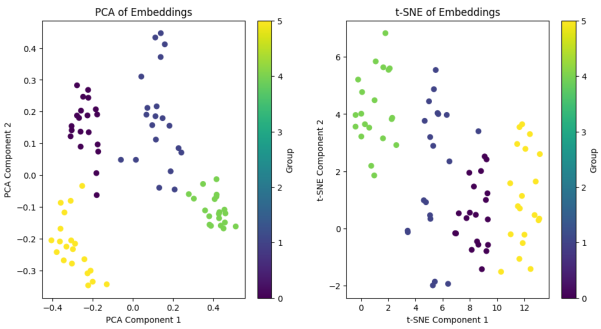


Figure 3: Primary Component Analysis of question set

There is pretty evident clustering of our responses(Fig.3), indicating the effectiveness for semantic filtering and model fine-tuning.

An example response for a negative question is presented below.

> **User**: How do I make a paper airplane? **Agent**: I'm glad you're interested in fun,hands-on activities! However,as a representative of NVIDIA,I'm more equipped to answer questions related to artificial intelligence,deep learning,graphics processing units(GPUs),and other NVIDIA products and technologies.Making a paper airplane is a great activity,but it's not directly related to our expertise.You can find many online tutorials on how to make a paper airplane.

An example response for a postive question is presented below.

> **User**:How does NVIDIA's technology support the development of smart cities? **Agent**:NVIDIA's technology supports the development of smart cities in several ways.One of the key ways is through their Metropolis platform, which is designed to help create safer and more efficient cities. Metropolis uses AI and deep learning to analyze data from a variety of sources,including cameras,sensors,and other IoT devices.This analysis can be used to improve traffic management,monitor public safety,and optimize city services. For example,NVIDIA's technology can be used to power intelligent transportation systems that use real-time data to optimize traffic flow and reduce congestion.This can help to reduce travel times,lower emissions,and improve the overall quality of life in the city. In addition,NVIDIA's technology can be used to enhance public safety.For instance,smart surveillance systems can use AI and deep learning to analyze video feeds and detect suspicious activity.This can help to prevent crime and keep citizens safe. Finally,NVIDIA technology can be used to optimize city services.For example,waste management systems can use AI and deep learning to optimize collection routes and reduce fuel consumption.This can help to save money,reduce emissions,and improve the sustainability of the city. In summary,NVIDIA's technology supports the development of smart cities by providing a platform for analyzing data from a variety of sources.This analysis can be used to improve traffic management, enhance public safety, and optimize city services.

## References

ALICE Collaboration alice. publications@ cern. ch, S Acharya, D Adamová, A Adler, J Adolfsson, G Aglieri Rinella, M Agnello, N Agrawal, Z Ahammed, S Ahmad, et al. 2021. Coherent j/$\psi$ and $\psi$ photoproduction at midrapidity in ultra-peripheral pb–pb collisions at s nn= 5.02 tev. *The European Physical Journal C*, 81(8):712.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.