UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

CARRERA DE CIENCIA DE LA COMPUTACIÓN

CrimeLens Chat: An LLM-Powered Natural-Language and Visual Analytics Interface for Local Crime Exploration

TRABAJO DE INVESTIGACIÓN

Para optar el grado de Licenciado en Ciencia de la computación

AUTOR(ES)

Jeremy Jeffrey Matos Cangalaya

Marcos Daniel Ayala Pineda

ASESOR(ES)

Cristian Lopez

Germain Garcia

Lima - Perú

2025

Contents

CHAPTER I

INTRODUCTION

1.1 Introduction to the Research Topic

1.2 Description of the Problematic Situation

In many large cities, concerns about crime have become increasingly common, especially in areas frequently visited by tourists [**?**]. Popular destinations are often associated with considerable levels of criminality, which can discourage visitors from exploring freely [**?**]. However, the impact of urban crime extends beyond tourists. Local residents, while more familiar with their surroundings, typically rely on personal experience, local news, and community word-of-mouth to decide which streets or neighborhoods to avoid. This informal awareness helps them navigate perceived danger zones, but it does not fully shield them from the risks that exist in their own cities.

In response to these concerns, several visualization tools ([**?**], [**?**], [**?**], [**?**], [**?**]) have emerged to increase awareness, validate hypotheses, and support safer decision-making. However, these solutions are often difficult to use for non-expert users. They usually provide sophisticated analysis using robust feature engineering processes, restricting their accessibility to a broader audience. As a result, both citizens and authorities are left with insufficient support to anticipate risks or act proactively.

A key challenge in crime prevention is the lack of accessible tools that help both citizens and authorities interpret complex crime data. While large volumes

of data are publicly available ([**?**], [**?**], [**?**]), they're often difficult to analyze without technical skills or time [**?**]. Crime patterns constantly shift across time and space. Some areas may be riskier at night, others during weekends or certain seasons. Without tools that clearly communicate these dynamics in real time, people struggle to make safe decisions, and authorities face obstacles in allocating resources effectively.

Although recent tools have improved the visual representation of crime data, they still present critical limitations in usability and applicability. Many platforms depend heavily on user expertise and offer mostly static or pre-defined visualizations that demand manual interpretation [**?**]. This limits their value in dynamic, real-world scenarios where fast decisions are required. Without systems that deliver clear, real-time insights tailored to a user's context, both citizens and authorities remain underserved in their efforts to understand and respond to urban crime.

1.3 Justification

Large language models (LLMs) offer a human-centred interface for interacting with complex data [**?**] [**?**]. Unlike, web dashboards, GIS (Geographic Information System) software or specialized tools, LLM-powered chatbot lets users ask questions like "Which streets saw the largest surge in robberies in the last month?" and instantly receive concise explanations. In addition, open-source models alleviate many of the privacy concerns often associated with proprietary systems and can be more cost-effective [**?**].

Adopting an LLM-based conversational layer directly addresses the two principal weaknesses of current crime-analytics platforms: usability and applicability. First, a chat interface removes the steep learning curves of existing tools. Second,

by supporting rapid, iterative exploration of temporal windows, geographic areas and crime types, it delivers immediate insights when users need them.

In this study, we aim to enhance crime-prevention efforts and raise awareness about the risks associated with routes used by tourists and residents by developing an intuitive chat interface. This interface will allow users to interact with crime data, presenting both textual summaries and dynamic visual feedback, to facilitate informed decision-making. Additionally, the work establishes a foundation for future investigations into the application of LLMs to more sophisticated crime-analysis tasks.

1.4 Research Objectives

1.4.1 General Objectives

To develop an chat-based system that leverages LLMs to enhance crime prevention efforts by enabling users to interact with spatio-temporal crime data through natural language queries, thereby improving decision-making for both citizens and authorities.

1.4.2 Specific Objectives

- Design and implement a chat-based interface that allows users to query crime data and receive both textual and visual feedback.
- Address privacy and security concerns by focusing on the use of open-source LLMs as alternative solutions to proprietary models.

- Implement and agent-based system that utilizes LLMs to process and analyze spatio-temporal crime data, providing real-time insights and recommendations.
- Create a question-answering dataset specifically tailored for crime data.

CHAPTER II

THEORETICAL FRAMEWORK

2.1 Spatio-Temporal Data

The spatio-temporal data is a type of data that contains information about the spatial and temporal dimensions of an event or phenomenon. This dual indexing enables the representation of complex relationships between spatial and temporal elements, allowing for a more comprehensive understanding of the data.

The spatio-temporal data can be formalized as a tuple $ST = \{(s_i, t_j, X_{ij} | s_i \in S, t_j \in T, X_{ij} \in \mathbb{R}^d)\}$, where S is the spatial domain, T is the temporal domain, and X_{ij} signifies the observed attributes at location s_i and time t_j.

2.2 Language Models (LLMs)

Large Language Models (LLMs) are built upon the transformer architecture, originally introduced by [**?**]. This architecture revolutionized natural language processing (NLP) by replacing recurrent neural networks (RNNs) with self-attention mechanisms, enabling models to process entire sequences in parallel rather than sequentially.

The core components of the transformer architecture include:

- **Self-attention mechanisms**: Allow the model to assess the importance of different words within a given context. By computing weighted sums of all

positions in a sequence, with weights determined by query-key interactions, the model can focus on relevant information while ignoring irrelevant parts.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q is the query matrix, K is the key matrix, V is the value matrix, and d_k is the dimension of the keys.

- **Multi-head attention**: Enables the model to attend to information from multiple representation subspaces simultaneously.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and W_i^Q, W_i^K, W_i^V are learned projection matrices.

- **Feed-forward neural networks**: Apply non-linear transformations to the attention outputs.

- **Layer normalization**: Helps stabilize and accelerate the training process.

$$LayerNorm(x) = \omega \odot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

where μ and σ^2 are the mean and variance of the input, ϵ is a small constant for numerical stability, and ω and β are learnable parameters.

- **Positional encoding**: Introduces information about the order of tokens in a sequence. It is tipically encoded using sine and cosine functions of different frequencies.

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

where pos is the position and i is the dimension index.

Notable examples of LLMs include OpenAI's GPT-3 [**?**], Google's BERT [**?**] and T5 [**?**], and Meta's RoBERTa [**?**]. These models have set new benchmarks across a variety of NLP tasks such as text classification, machine translation, and summarization. While all are built on the transformer architecture, they differ in their design choices and training objectives. For instance, BERT [**?**], an encoder-only transformer, uses a masked language modeling objective, where random tokens in a sentence are masked and the model learns to predict them using surrounding context. In contrast, GPT-3 [**?**], a decoder-only transformer, follows an autoregressive training strategy, generating one token at a time based on previously generated tokens.

The theoretical foundation of LLMs lies in probabilistic language modeling. Given a sequence of tokens $x = (x_1, x_2, \ldots, x_n)$, a language model aims to estimate either the joint probability $P(x)$ or the conditional probability of the next token $P(x_n \mid x_1, \ldots, x_{n-1})$. The model is trained to assign higher probabilities to sequences that are more likely to appear in natural language, based on patterns learned from large-scale datasets. This is made possible by the transformer's use of self-attention and feed-forward layers, which together capture complex dependencies across tokens. In particular, positional encodings and multi-head attention enable LLMs to model long-range relationships—something that earlier architectures like RNNs and LSTMs struggled to achieve effectively.

2.2.1 Supervised Fine-Tuning

Supervised fine-tuning (SFT) of large language models (LLMs) is a technique used to adapt pre-trained models on large corpora to specific tasks, improving their performance and alignment with human preferences or task requirements [**?**]. This process involves training the model on a labeled dataset, where each input is paired with a corresponding output.

A common variant of SFT is instruction tuning, which trains the model on supervised datasets containing (instruction, output) pairs [**?**]. This method improves the model's zero-shot performance by shifting its behavior from next-token prediction to better understanding and following human directions [**?**].

2.2.2 LoRA (Low-Rank Adaptation)

LoRA [**?**] is a parameter-efficient fine-tuning technique that reduces the computational cost of adapting large language models. Instead of updating all model parameters, LoRA introduces low-rank matrices to the model's architecture, enabling task-specific adaptation with minimal resource usage. This makes it particularly useful for deploying LLMs in resource-constrained environments.

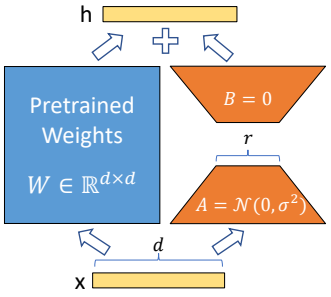

FIGURE 2.1: LoRA

2.2.3 Tool Integrated Reasoning

Several works combine natural language reasoning with Python code execution to improving accuracy on math problem resolution tasks [**?**], [**?**], [**?**], [**?**].

2.3 Prompting

Basically prompting is a technique used to guide the behavior of LLMs by providing them with specific instructions or context. This can be done through various methods, such as zero-shot prompting, few-shot prompting, and chain-of-thought prompting. Each method has its own advantages and disadvantages, depending on the task at hand and the desired outcome.

2.3.1 Prompt Engineering

Prompt engineering is an emerging field focused on crafting and refining prompts to make the most effective use of language models (LMs) across diverse applications and research areas. It plays a crucial role in enhancing our understanding of the strengths and limitations of LLMs. Beyond simply writing prompts, prompt engineering involves a broad set of techniques essential for building with, interacting with, and expanding the capabilities of LLMs. It also contributes to improving model safety and enables the integration of specialized knowledge and functionalities into LLM-based systems

The following methods are some of the most common techniques and used in our work:

- **Zero-shot prompting**: Involves providing the model with a task description or question without any examples. The model is expected to generate a

response based solely on its pre-existing knowledge and understanding of the task.

- **Few-shot prompting**: As mentioned by [**?**], this technique involves providing the model with a few examples of the desired output format or task. This helps the model understand the context and generate more accurate responses.
- **Chain-of-thought prompting**: Firstly introduced by [**?**], encourages the model to generate intermediate reasoning steps before arriving at a final answer. This approach has been shown to improve performance on complex tasks, such as arithmetic reasoning and logical inference.

2.4 Retrieval Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG), first introduced by [**?**], is a framework that combines the strengths of retrieval-based and generative approaches for question answering. Its core idea is to enhance the generative capabilities of language models by incorporating relevant information retrieved from external knowledge sources, rather than relying solely on parametric knowledge (i.e., information encoded in the model's weights).

The RAG architecture consists of two primary components:

- **Retriever**: This component retrieves relevant documents or sentences from a large corpus based on the input query. It typically employs models such as BM25 [**?**] or dense retrieval methods to identify the most relevant information.
- **Generator**: Once the retriever has collected the relevant documents, the generator—usually a transformer-based language model—takes them as input to produce a coherent and contextually appropriate response. The generator can be fine-tuned for specific tasks to further improve its performance.

In recent years, RAG has gained considerable attention in the NLP community for its ability to generate high-quality responses. This has spurred the development of new techniques to improve both retrieval precision and generation quality. For instance, [**?**] introduces a modular RAG framework that enables the integration of diverse components across different stages of the pipeline. These include techniques such as query expansion, reformulation, and transformation in the pre-retrieval phase, as well as the use of LLMs as judges in the post-retrieval phase to evaluate if the response is enoughly complete. This modular design encourages flexibility and supports experimentation with novel combinations and architectural variations.

2.5 Metrics

2.5.1 Pass@K

[**?**]

2.5.2 Majority@K

[**?**]

2.5.3 Code-BLEU

[**?**]

CHAPTER III

STATE OF THE ART

3.1 Large-Language Models for Urban Scenarios

Recent advances show that LLMs are increasingly being adapted to address urban computing tasks. Three recent approaches illustrate how these models can be used for both forecasting urban phenomena and orchestrating multiple models to tackle complex urban tasks.

[**?**] proposes UrbanGPT, a spatio-temporal LLM for forecasting urban dynamics such as traffic flows and crime rates. The model receives spatial and time series information through the prompt, then employs a spatio-temporal dependency encoder and a lightweight alignment module to project these representations into the LLM's latent space, achieving performance on par with or surpassing state-of-the-art models in multiple datasets.

Complementing this, [**?**] leverage the agentic capabilities of LLMs to decompose urban-related queries into structured sub-tasks (e.g., forecasting, anomaly detection, POI recommendation, etc). This approach, termed UrbanLLM, assigns each sub-task to a specialised model from a curated model zoo, and integrates the results into a unified response.

Recent research from Google has explored the use of foundation models for geospatial reasoning [**?**]. It introduces an agentic workflow powered by Gemini to assist users in tasks such as visualizing pre and post disaster scenarios or conducting damage assessments. Their approach integrates diverse modalities: maps, weather data, and satellite imagery, and highlights the need for foundation models

capable of aligning heterogeneous spatial information . Together with UrbanGPT's forecasting focus and UrbanLLM's model orchestration, this work reflects a growing trend toward multimodal LLM-driven systems for urban scale analysis and decision making.

Another study from Google introduces Visual Chronicles [**?**], a multimodal LLM-based system designed to identify and describe frequently occurring visual changes across urban environments using a dataset provided by Google Street View imagery. It leverages a vast collection of geolocated, timestamped images to identify trends without requiring labeled training data. To overcome the limitations of MLLMs in processing such massive datasets, the authors design a scalable pipeline that enables efficient retrieval, comparison, and semantic analysis of visual patterns across both space and time.

3.2 RAG Techniques for Complex Data

Retrieval-Augmented Generation (RAG) frameworks aim to enhance LLMs by integrating external sources of knowledge, such as structured databases, time series, or knowledge graphs. Recent research has extended RAG beyond textual documents to support spatial, temporal, and graph-based data retrieval.

[**?**] extends RAG to spatial tasks by integrating sparse spatial retrieval (SQL-based queries) with dense semantic retrieval (LLM-based similarity). Their method introduces three preprocessing steps to help the LLM generate complete and executable SQL queries, addressing its limitations in query formulation.

In the temporal domain, [**?**] apply RAG to the context of time series forecasting using Dynamic-Time Warping (DTW) as a distance metric to retrieve similar waveforms and trends, given a time serie as a query. The retrieved information is then utilized to improve the LLM forecasting accuracy.

Other works, combine RAG techniques and hybrid approaches to address question-answering over textual knowledge graphs. One such example is [**?**], who introduce G-Retriever, a flexible QA framework for knowledge graphs that incorporates a RAG into its pipeline. The framework separates node entities and edge information into two distinct embedding spaces. Using cosine similarity, it retrieves the most relevant nodes and edges for the query and reconstructs the subgraph using the Prize-Collecting Steiner Tree (PCST) algorithm. The final answer is generated by a hybrid GNN-LLM, which processes the retrieved subgraph both as text in the query prompt and through a graph encoder aligned with the LLM's token space.

Building on this idea of graph-based retrieval, [**?**] map textual subgraphs directly to an embedding space, allowing for the retrieval of relevant subgraphs based on their semantic similarity to the query. Then applied techniques to merging and pruning the retrieved subgraphs to improve the quality of the final answer.

Recent work from [**?**] propose LightRAG, a fully prompt-driven framework that extracts knowledge graphs, generates keywords at multiple granularities, retrieves from both vector and graph indexes, and supports fast incremental updates.

3.3 Open Crime Datasets

Among the most popular datasets for crime analysis are the Chicago Crime dataset [**?**] and the New York City Crime dataset [**?**]. These datasets contain detailed records of reported crimes, including information on the type of crime, location, time, and other relevant attributes. They have been widely used in various research studies and applications related to crime prediction, analysis, and visualization.

Beyond the United States, similar efforts have been made in Latin America. In Brazil, for example, crime datasets are made publicly available at the state level through open data platforms. These resources have supported a range of research

projects, including [**?**] and [**?**], which process and analyze regional crime patterns or develop predictive models.

More recently, large-scale initiatives have emerged in Asia. [**?**] introduces a large-scale crime dataset from China, comprising approximately 1 million records. The dataset spans 31 provincial-level administrative regions, 222 city-level divisions, and 548 county (district)-level jurisdictions across mainland China. Unlike the structured records in the aforementioned datasets, this resource was constructed by extracting crime information from unstructured judicial documents using LLMs, enabling broader geographic and semantic coverage. Additionally, it includes detailed fields such as case descriptions, victim and defendant information, and final judgments, offering more possibilities for anlysis and research.

3.4 Crime-Data Visualization Tools

In the context of crime data analysis, several visualization systems have been proposed to support pattern recognition, hotspot identification, and urban context interpretation. These tools typically integrate geospatial data with interactive visual analytics techniques to assist expert users in understanding complex crime patterns.

Early tools like CrimeVis [**?**], focused on interactive exploration accross police districts (DPs) brushing-and-linking techniques. Later systems extended this groundwork by incorporating advanced feature engineering and machine learning techniques. CrimAnalyzer [**?**] proposed a Non-negative Matrix Factorization (NMF) based technique to identify hostpots. Furthermore, CriPAV [**?**] incorporates autoencoders to embed and cluster hotspots, facilitating the analysis of the relationship between crime and urban features.

3.5 NLP in Data Visualization

Eviza [**?**] convert natural language input to filters applied to visualizations.

[**?**] introduces Eviza, a natural language interface for visual data analysis, leveraging a probabilistic grammar-based approach with predefined syntactic rules. The system incorporates a template-based autocompletion feature to provide users with contextual suggestions, enabling an interactive conversation with existing visualizations. Tested on geographic datasets, such as earthquake data in the US, the interface enhances user interaction by implementing language pragmatics through a finite state machine. However, Eviza faced challenges in recognizing all parts of long and complex queries and struggled with certain grammatical constructs, highlighting the need for more robust natural language processing techniques in visualization systems.

[**?**]

Early research on natural language interfaces for data visualization has explored how users can interact with visual content through conversational input.

[**?**] introduce a framework for controlling data visualizations through natural language. Their approach centers on two key components: a natural language-to-task translator and a visualization manipulation parser. The translator, based on a fine-tuned T5 model, maps user queries into a hierarchical structure of tasks, which are then interpreted to apply manipulation operations over existing visualizations.

[**?**]

[**?**]

[**?**]

CHAPTER IV

METHODOLOGY

Recent advancements in Retrieval-Augmented Generation (RAG) have shown great promise across various domains [? ? ? ? ?]; however, none of these architectures are specifically tailored for spatio-temporal information retrieval and reasoning over crime data. Existing approaches typically focus on textual or knowledge graph-based sources, leaving a key research gap for systems capable of handling dynamic urban crime contexts across space and time.

In parallel, numerous visualization tools have been developed to support crime data analysis [? ? ? ? ?], and recent works attempt to bridge natural language interfaces with visual analytics [?]. Yet, none of these efforts fully integrate geographic crime data querying through natural language while also offering intuitive, interactive visualizations. This highlights a missed opportunity to democratize access to urban crime insights.

Moreover, while large language models (LLMs) have been used for spatio-temporal question answering, current pipelines still face limitations. For instance, [?] embed domain knowledge directly into the model via fine-tuning, which reduces transparency and flexibility. Google's recent work [?] proposes LLM-based reasoning over geospatial data, but without a focus on urban safety or crime-specific tasks. Similarly, [?] and [?] apply LLMs to urban computing, but rely on pre-embedded data in prompts, bypassing interactive user-driven retrieval.

This section presents our proposed architecture, designed to overcome these limitations by enabling spatio-temporal crime data analysis and visualization through natural language interaction.

4.1 Dataset Creation

[**?**], [**?**]

4.2 Model Training

[**?**]

4.3 Methodological Proposal

The proposal is divided into two phases. The first phase focuses on the development of a prototype that integrates a hybrid retrieval mechanism with an LLM-based chat interface by using a criminalistic dataset from China as a knowledge base. The evaluation method will be done by user studies, in order to determine its effectiveness in answering spatio-temporal crime questions. The second phase aims to enhance the system by generating synthetic datasets and fine-tuning the selected LLMs based on the generated data. This approach will allow us to improve the model's performance and adapt it to specific crime-related tasks.

[**?**], [**?**], [**?**], [**?**], [**?**]

4.4 Datasets

We base our prototype on the dataset introduced by [**?**], which provides a large-scale, open-access repository of nearly one million criminal court records across China. From this dataset, we construct a benchmark of spatio-temporal statistical questions designed to evaluate the performance of our LLM-based system in crime data exploration. Inspired by prior work on question classification for temporal

knowledge graphs [**?**], as well as tourism and spatial reasoning benchmarks [**?** **?**], we define a taxonomy of question types that reflects the analytical goals of urban crime investigation.

Table **??** summarizes the types of questions we support, along with representative templates and instantiated examples using the Chinese crime dataset.

[**?**], [**?**], [**?**], [**?**]

TABLE 4.1: Question type examples supported over the spatio-temporal crime dataset

Category	**Question Template Example**
Simple time reasoning	How many crimes occurred on <Time Entity>?
Spatial aggregation	How many incidents occurred in <Spatial Entity>?
Spatio-temporal filtering	How many crimes happened in <Spatial Entity>during <Time Entity>?
Before/After comparison	Did crime increase in <Spatial Entity>after <Time Point>?
First/Last occurrence	When was the last crime reported in <Spatial Entity>?
Most affected area	What is the most crime-prone <Spatial Level>during <Time Period>?
Location-based correlation	How does crime frequency vary between <Entity 1>and <Entity 2>?
Intersection or routing	What streets intersect with <Street Name>?

4.4.1 Phase 1: Expected PFC3

The following components structure this phase:

- **Dataset Question Generation**: Utilizing question templates such as those proposed in [**?**] and [**?**], we will generate spatio-temporal statistical questions derived from the dataset presented in [**?**].

- **Hybrid Retrieval Mechanism**: Drawing on methodologies outlined in works like [**?**], a hybrid RAG mechanism will be implemented, integrating vector-based and query-based retrieval approaches to enhance accuracy.

- **Model Selection and Prompting**: Open-source LLMs, like Llama3 [**?**] and Qwen2.5 [**?**] series, will be evaluated for their effectiveness in generating responses to the formulated questions.

- **Chat Implementation and Feedback Loop**: A chat-based interface will be developed to preprocess user queries and generate responses. This interface will incorporate geospatial visualization tools to provide users with visual feedback. Additionally, user studies will be conducted to evaluate the usability and effectiveness of the interface.

4.4.2 Phase 1: Pipeline

The proposed pipeline, illustrated in Figure **??**, describes the architecture of the first prototype. The system is designed to process natural language queries about crime occurrences across space and time, combining language models with a hybrid retrieval strategy and visual feedback mechanisms.

The architecture begins with an **LLM-powered chat interface**, where users can ask statistical-spatio temporal queries. This initial query is processed through a **query decomposition module**, which splits the question into street-level sub-queries that the selected model can handle.

These subqueries are routed through a **hybrid retrieval mechanism**, inspired by [**?**], which is divided in two **LLM extractors**. The first, the **Entity Extractor** extract named entities (e.g., street names) using a dense retriever based on multilingual sentence embeddings and a sparse retriever using BM25. While the **Time Extractor** extracts temporal references (e.g., "Q1 2024", "last month"). With this information we retrieve the spatio-temporal data from the dataset, which is a collection of crime records. These records are then passed to a **data parsing module**, which processes the data and pass to the prompt as a contextual information.

Once the context has been aligned, each subquery is processed by an LLM, which generates a set of answers. These answers are then passed to a **summarization module** that produces the final response. This response is further enhanced with **visual elements**, such as a map that highlights the queried streets.

The **query decomposition module** and **summarization module** was inspired by the bottom-up approach presented by [**?**].

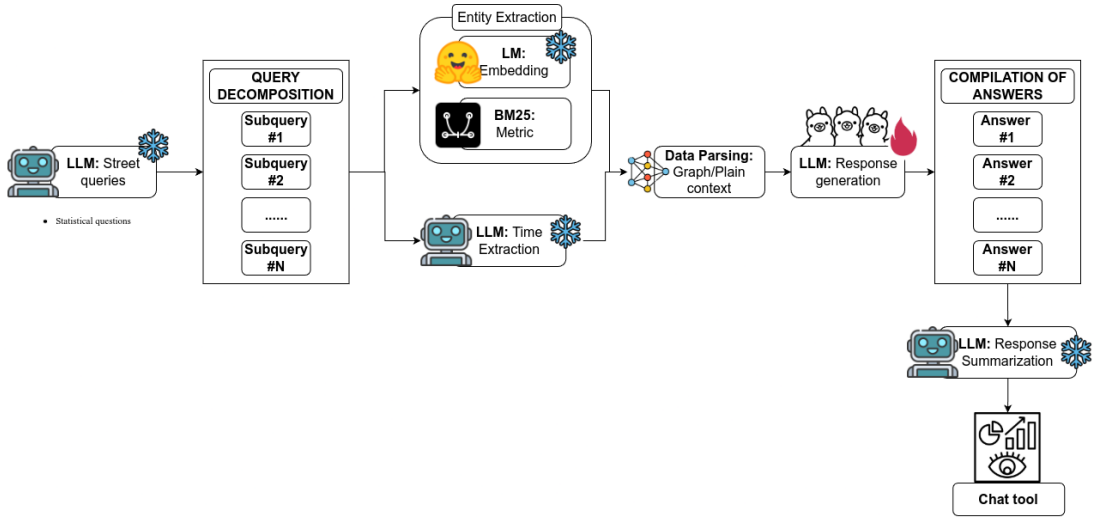

FIGURE 4.1: Proposed Pipeline - Phase 1

4.4.3 Phase 2: Expected Next Semester

- **Synthetic Dataset Generation**: The capabilities of proprietary LLMs will be leveraged to generate synthetic answers ([**?**], [**?**]) based on the formulated questions and the retrieval mechanism. The generated chain-of-thought (CoT) reasoning and associated code will be stored for fine tune the selected LLMs.
- **Model Fine-tuning and Evaluation**: The selected LLMs will be fine-tuned using the expanded dataset, and their performance will be assessed using metrics such as BLEU [**?**], ROUGE [**?**], BertScore [**?**], and METEOR [**?**].

[**?**]

CHAPTER V

EXPERIMENTACIÓN Y RESULTADOS PRELIMINARES

5.1 Experimentos

Table 5.1: Evaluation Metrics: Mean, Std, and Median per Model

Metric	GPT-4o			Deepseek-chat		
	Mean	Std	Median	Mean	Std	Median
pass@k	0.0	0.0	0.0	0.0	0.0	0.0
maj@k	0.0	0.0	0.0	0.0	0.0	0.0
passˆk	0.0	0.0	0.0	0.0	0.0	0.0
code_bleu@k	0.273	0.045	0.273	0.304	0.042	0.29
perc_error@k	1.0	0.0	1.0	1.0	0.0	1.0

Table 5.2: Evaluation Metrics of Qwen2.5-code-7b: Mean, Std, and Median per Model Checkpoint

Metric	Checkpoint 50			Checkpoint 300		
	Mean	Std	Median	Mean	Std	Median
pass@k	0.115	0.319	0.0	0.12	0.324	0.0
maj@k	0.046	0.209	0.0	0.054	0.227	0.0
passˆk	0.017	0.064	0.0	0.015	0.051	0.0
code_bleu@k	0.36	0.065	0.356	0.357	0.227	0.352
perc_error@k	0.402	0.213	0.375	0.419	0.219	0.375

5.2 Resultados y discusión

BIBLIOGRAPHY

[1] S. T. S. Tevdoradze, Z. M. Z. Mushkudiani, and N. T. N. Tevdoradze, "The effect of criminal activity on tourism," *The New Economist*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:273525446

[2] M. de Barros Tomé Machado, "Medo social e turismo no rio de janeiro," *Tourism & Management Studies*, pp. 48–54, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:155721043

[3] G. García-Zanabria, M. M. Raimundo, J. Poco, M. B. Nery, C. T. Silva, S. Adorno, and L. G. Nonato, "Cripav: Street-level crime patterns analysis and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4000–4015, Dec 2022.

[4] M. M. Salah and K. wen Xia, "Big crime data analytics and visualization," *Proceedings of the 2022 6th International Conference on Compute and Data Analysis*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248990234

[5] L. J. S. Silva, S. Fiol-González, C. F. P. Almeida, S. D. J. Barbosa, and H. C. V. Lopes, "Crimevis: An interactive visualization system for analyzing crime data in the state of rio de janeiro," in *International Conference on Enterprise Information Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:46819581

[6] G. G. Zanabria, E. G. Nieto, J. Silveira, J. Poco, M. B. Nery, S. Adorno, and L. G. Nonato, "Mirante: A visualization tool for analyzing urban crimes," *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 148–155, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227221852

[7] G. Garcia, J. Silveira, J. Poco, A. Paiva, M. B. Nery, C. T. Silva, S. Adorno, and L. G. Nonato, "Crimanalyzer: Understanding crime patterns in são paulo," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 4, pp. 2313–2328, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2019.2947515

[8] Y. Zhang, M.-P. Kwan, and L. Fang, "An llm driven dataset on the spatiotemporal distributions of street and neighborhood crime in china," *Scientific Data*, vol. 12, no. 1, p. 467, mar 2025. [Online]. Available: https://doi.org/10.1038/s41597-025-04757-8

[9] N. Y. C. P. Department, "Nypd complaint data historic," https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qb7u-rbmr, 2025.

[10] C. P. Department, "Crimes - 2001 to present," https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2, 2024.

[11] Z. Wang and H. Zhang, "Construction, detection, and interpretation of crime patterns over space and time," *ISPRS International Journal of Geo-Information*, vol. 9, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2220-9964/9/6/339

[12] D. Yang, S. T. Wu, and M. A. Hearst, "Human-ai interaction in the age of llms," *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies (Volume 5: Tutorial Abstracts), 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270514463

[13] S. R. Pappula and S. R. Allam, "Llms for conversational ai: Enhancing chatbots and virtual assistants," *International Journal of Research Publication and Reviews*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266461220

[14] C. Liu, J. Yu, Y. Guo, J. Zhuang, Y. Luo, and X. Yuan, "Breathing new life into existing visualizations: A natural language-driven manipulation framework," 2024. [Online]. Available: https://arxiv.org/abs/2404.06039

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[20] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, and Q. Liu, "Aligning large language models with human: A survey," 2023. [Online]. Available: https://arxiv.org/abs/2307.12966

[21] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," 2022. [Online]. Available: https://arxiv.org/abs/2109.01652

[22] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang, "Instruction tuning for large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2308.10792

[23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[24] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, M. Huang, N. Duan, and W. Chen, "Tora: A tool-integrated reasoning agent for mathematical problem solving," 2024. [Online]. Available: https://arxiv.org/abs/2309.17452

[25] I. Moshkov, D. Hanley, I. Sorokin, S. Toshniwal, C. Henkel, B. Schifferer, W. Du, and I. Gitman, "Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset," 2025. [Online]. Available: https://arxiv.org/abs/2504.16891

[26] S. Yin, W. You, Z. Ji, G. Zhong, and J. Bai, "Mumath-code: Combining tool-use large language models with multi-perspective data

augmentation for mathematical reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2405.07551

[27] J. Li, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. Huang, K. Rasul, L. Yu, A. Q. Jiang, Z. Shen, Z. Qin, B. Dong, L. Zhou, Y. Fleureau, G. Lample, and S. Polu, "Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions," Numina, Hugging Face, MIT, Mistral AI, Peking University, Answer AI, Tech. Rep., July 2024, winner of the 1st AIMO Progress Prize. [Online]. Available: https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf

[28] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[29] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[30] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[31] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, p. 333–389, Apr. 2009. [Online]. Available: https://doi.org/10.1561/1500000019

[32] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular rag: Transforming rag systems into lego-like reconfigurable frameworks," 2024. [Online]. Available: https://arxiv.org/abs/2407.21059

[33] N. Levi, "A simple model of inference scaling laws," 2024. [Online]. Available: https://arxiv.org/abs/2410.16377

[34] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[35] S. Ren, D. Guo, S. Lu, L. Zhou, S. Liu, D. Tang, N. Sundaresan, M. Zhou, A. Blanco, and S. Ma, "Codebleu: a method for automatic evaluation of code synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2009.10297

[36] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, "Urbangpt: Spatio-temporal large language models," 2024. [Online]. Available: https://arxiv.org/abs/2403.00813

[37] Y. Jiang, Q. Chao, Y. Chen, X. Li, S. Liu, and G. Cong, "Urbanllm: Autonomous urban activity planning and management with large language models," 2024. [Online]. Available: https://arxiv.org/abs/2406.12360

[38] D. Schottlander and T. Shekel, "Geospatial reasoning: Unlocking insights with generative ai and multiple foundation models," Google Research Blog, Apr. 2025, available at: https://research.google/blog/geospatial-reasoning-unlocking-insights-with-generative-ai-and-multiple-foundation-models/ (Accessed: 2025-04-18).

[39] B. Deng, S. Peng, K. Genova, G. Wetzstein, N. Snavely, L. Guibas, and T. Funkhouser, "Visual chronicles: Using multimodal llms to analyze massive collections of images," 2025. [Online]. Available: https://arxiv.org/abs/2504.08727

[40] D. Yu, R. Bao, G. Mai, and L. Zhao, "Spatial-rag: Spatial retrieval augmented generation for real-world spatial reasoning questions," 2025. [Online]. Available: https://arxiv.org/abs/2502.18470

[41] S. Yang, D. Wang, H. Zheng, and R. Jin, "Timerag: Boosting llm time series forecasting via retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/abs/2412.16643

[42] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," 2024. [Online]. Available: https://arxiv.org/abs/2402.07630

[43] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, "Grag: Graph retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/abs/2405.16506

[44] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "Lightrag: Simple and fast retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/abs/2410.05779

[45] W. Hassan, M. M. Cabral, T. R. Ramos, A. C. Filho, and L. G. Nonato, "Modeling and predicting crimes in the city of sao paulo using graph neural networks," in *Intelligent Systems*, A. Paes and F. A. N. Verri, Eds. Cham: Springer Nature Switzerland, 2025, pp. 372–386.

[46] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang, "Eviza: A natural language interface for visual analysis," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 365–377. [Online]. Available: https://doi.org/10.1145/2984511.2984588

[47] Y. Wu, Y. Wan, H. Zhang, Y. Sui, W. Wei, W. Zhao, G. Xu, and H. Jin, "Automated data visualization from natural language via large language models: An exploratory study," *Proc. ACM Manag. Data*, vol. 2, no. 3, May 2024. [Online]. Available: https://doi.org/10.1145/3654992

[48] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 217–226, 2022.

[49] A. Narechania, A. Srinivasan, and J. Stasko, "Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 369–379, Feb. 2021. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2020.3030378

[50] C. Liu, Y. Han, R. Jiang, and X. Yuan, "Advisor: Automatic visualization answer for natural-language question on tabular data," in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, 2021, pp. 11–20.

[51] Q. Wei, M. Yang, J. Wang, W. Mao, J. Xu, and H. Ning, "Tourllm: Enhancing llms with tourism knowledge," 2024. [Online]. Available: https://arxiv.org/abs/2407.12791

[52] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," 2020. [Online]. Available: https://arxiv.org/abs/1904.09751

[53] Qwen *et al.*, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

[54] Y. Fleureau, J. Li, E. Beeching, L. Tunstall, B. Lipkin, R. Soletskyi, S. Costa Huang, and K. Rasul, "How numinamath won the 1st aimo

progress prize," https://huggingface.co/blog/winning-aimo-progress-prize, jul 2024, blog post, consultado el 6 de mayo de 2025.

[55] A. Jain, A. Maleki, and N. Saade, "How to fine-tune: Focus on effective datasets," https://ai.meta.com/blog/how-to-fine-tune-llms-peft-dataset-curation/, aug 2024, blog post, consultado el 6 de mayo de 2025.

[56] W. U. Ahmad, S. Narenthiran, S. Majumdar, A. Ficek, S. Jain, J. Huang, V. Noroozi, and B. Ginsburg, "Opencodereasoning: Advancing data distillation for competitive coding," 2025. [Online]. Available: https://arxiv.org/abs/2504.01943

[57] A. Saxena, S. Chakrabarti, and P. Talukdar, "Question answering over temporal knowledge graphs," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 6663–6676. [Online]. Available: https://aclanthology.org/2021.acl-long.520/

[58] D. Contractor, K. Shah, A. Partap, Mausam, and P. Singla, "Large scale question answering using tourism data," 2020. [Online]. Available: https://arxiv.org/abs/1909.03527

[59] X. Dai, H. Li, and G. Qi, "Question answering over spatio-temporal knowledge graph," 2024. [Online]. Available: https://arxiv.org/abs/2402.11542

[60] Unsloth Team, "Datasets guide - unsloth documentation," 2024, accessed: 2025-04-25. [Online]. Available: https://docs.unsloth.ai/basics/datasets-guide#how-big-should-my-dataset-be

[61] ——, "What model should i use? - unsloth documentation," 2024, accessed: 2025-04-25. [Online]. Available: https://docs.unsloth.ai/get-started/beginner-start-here/what-model-should-i-use

[62] A. G. et al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[63] NVIDIA Blog Team, "How ai reasoning is being tested with international math olympiad problems," 2024, accessed: 2025-04-20. [Online]. Available: https://blogs.nvidia.com/blog/reasoning-ai-math-olympiad/

[64] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. USA: Association for Computational Linguistics, 2002, p. 311–318. [Online]. Available: https://doi.org/10.3115/1073083.1073135

[65] K. Ganesan, "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks," 2018. [Online]. Available: https://arxiv.org/abs/1803.01937

[66] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020. [Online]. Available: https://arxiv.org/abs/1904.09675

[67] A. Lavie and A. Agarwal, "Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation*, ser. StatMT '07. USA: Association for Computational Linguistics, 2007, p. 228–231.

[68] A. Pareja, N. S. Nayak, H. Wang, K. Killamsetty, S. Sudalairaj, W. Zhao, S. Han, A. Bhandwaldar, G. Xu, K. Xu, L. Han, L. Inglis, and A. Srivastava,

"Unveiling the secret recipe: A guide for supervised fine-tuning small llms," 2024. [Online]. Available: https://arxiv.org/abs/2412.13337