Multi-Agent System for Visual of Citation Chronology from Scientific Paper

Submitted by:

Shivesh Kumar
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Under the Supervision of

Dr. Suman Kundu
Department of Computer Science and Engineering
Indian Institute of Technology, Jodhpur

June – July 2025

Abstract

In academic research, understanding citation relationships is crucial for literature analysis, topic discovery, and research mapping. This project presents a modular, multi-agent system that automates the extraction, summarization, clustering, and visualization of citations from academic PDF documents. Leveraging `Large Language Models (LLMs)` [3], the system identifies citation numbers, authors, and publication years, and generates concise two-line summaries for each citation. It utilizes modern embedding models (such as `nomic-embed-text` [8]) to vectorize these summaries and applies the elbow method with `K-Means` [7] clustering to determine the optimal number of citation groups. Each cluster is further refined through LLM-based topic inference, producing fine-grained thematic subgroups. The final output is a hierarchical `JSON` structure compatible with `D3.js` [2], enabling intuitive and interactive visualization of citation structures. This framework enhances citation comprehension, supports meta-analysis, and facilitates efficient, automated literature reviews.

By reducing the manual workload in literature analysis, this system empowers researchers to focus more on insights rather than data handling. Its scalable design opens new avenues for integrating intelligent citation analytics into academic search engines and review tools.

Introduction

Problem Statement

. Manually extracting and organizing citations from research papers is time-consuming, prone to errors, and inefficient, particularly with large documents. Existing tools often fail to understand the temporal context of citations or present them in a clear, chronologically meaningful way. This project addresses these limitations by proposing a Multi-Agent System that leverages `Large Language Models (LLMs)` [3] to automate citation extraction, summarization, and clustering, culminating in the generation of a visual citation chronology. It ensures speed, accuracy, and scalability of academic literature review.

Motivation

With the rapid growth of academic publications, manually analyzing citations has become inefficient and impractical. There is a need for an intelligent system that can automate citation extraction, summarization, and grouping to support faster literature reviews and research insights. Leveraging `LLMs` [3] enables deeper semantic understanding, making citation analysis more accurate, scalable, and insightful than traditional manual or rule-based methods.

Objective

The aim of this project is to design a `multi-agent pipeline` that automates the process of citations analysis from academic research paper. The manual way to extract the citations from the research paper is a laborious and unreliable task. Researchers often have to read through long papers and understand the context in which citations are being used.This process is not only time-consuming, but also inconsistent due to variation in interpretation.

The proposed system facilitates automatic extraction, understanding, and visualization of citations, supporting use cases such as literature reviews, citation mapping, and thematic analysis in a scalable and efficient manner.

Background

`Citations` play a crucial role in academic writing, as they provide credibility, track the development of knowledge, and help identify influential work within a research domain. Analyzing citations allows researchers to conduct literature reviews, understand thematic trends, and discover gaps in the existing body of knowledge.

With the advancements in `NLP`, particularly the emergence of `Large Language Models (LLMs)` [3] such as GPT, Mistral [1], and `LLaMA`, have significantly improved the ability to process and interpret unstructured text. Trained on extensive textual data, these models can perform complex tasks like information extraction, summarization, and semantic understanding without task-specific programming. Their contextual awareness makes them well-suited for academic applications, including citation extraction, metadata identification, and concise summarization of referenced works.

Organization of the Report

This report is organized into the following sections:

- **Introduction:** Provides an overview of the project, including the problem statement, motivation, and objectives.
- **Background:** Discusses the importance of citation analysis and recent advancements in `NLP`, especially the role of Large Language Models (LLMs) [3].
- **System Architecture:** Describes the overall design of the multi-agent pipeline, including individual agent responsibilities and data flow.
- **Methodology:** Explains the step-by-step process of citation extraction, metadata analysis, summarization, clustering, and visualization.
- **Results and Visualization:** Presents the output of the system, including citation timelines and D3 [2]-based hierarchical visualizations.
- **Conclusion and Future Work:** Summarizes key findings and suggests possible extensions and improvements for the system.
- **References:** Lists all the sources and tools referenced throughout the project.

Multi-Agent Framework Architecture

This system is proposed to build a multi-agent pipeline that contains five basic agents to perform the task effectively.
Agent I: Responsible for loading the research paper in `PDF` format and splitting it into appropriately sized text chunks. It also identifies and counts all valid citation numbers present in the document using an `LLM [3]-based extraction`.
Agent II: Responsible for extracting the author names associated with each identified citation using `LLM [3]-based semantic analysis` of the text.
Agent III: Responsible for extracting the publication year associated with each citation using contextual analysis. It also generates a basic timeline plot to visually represent citation distribution over the years.
Agent IV: Responsible for generating a unique two-line summary for each citation using an `LLM` [3]. It then applies the `K-means` [7] `clustering` algorithm to group similar citations based on their semantic content and classifies them into appropriate clusters.
Agent V: Responsible for presenting the final clustered citation output through interactive and visual representations. It utilizes a mind-map style hierarchical diagram using `D3.js` [2] and a citation clusters (fishbone plot) generated with `Matplotlib` [6] to effectively visualize citation groupings and their chronological distribution.

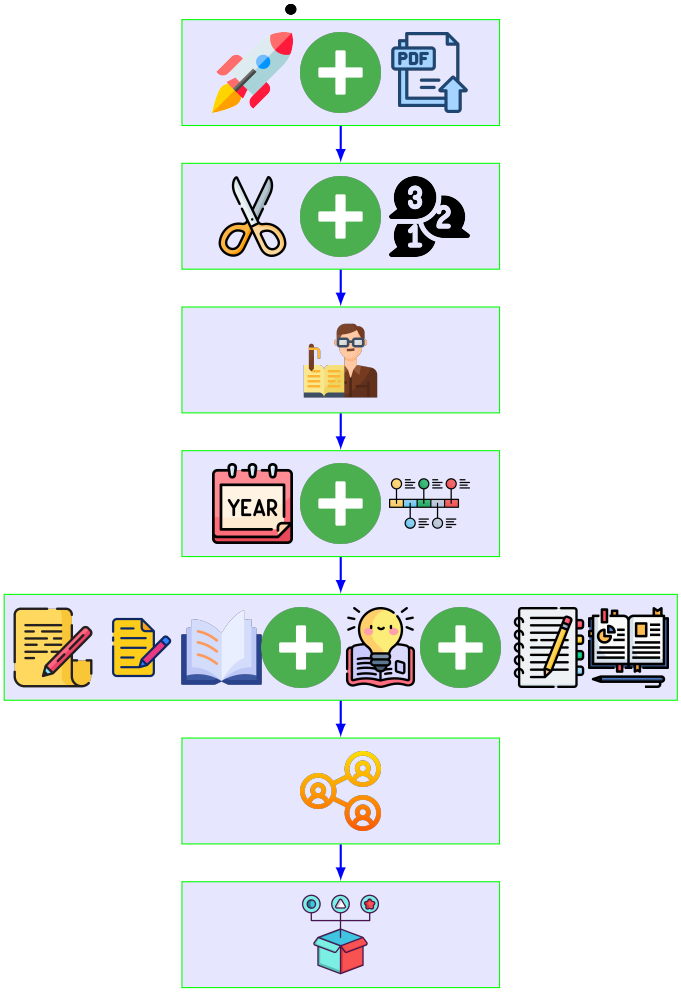

fig 1. Architecture flow diagram

Methodology

This project follows a modular, multi-agent process for citations extraction, analysis and visualization from academic research PDFs. Below is a detailed overview of each step:

Step 1: The system begins with accepting a research paper in pdf format. Using the PyPDFLoader from LangChain [5], the PDF is loaded and converted into raw text. To ensure the effectively processing the text is converted into manageable chunks using the RecursiveCharacterTextSplitter. An LLM [3] is then used to parse these chunks and extract the count of all unique citation numbers(e.g., [1], [2], (3), (Smith et al., 2021), [Touvron et al., 2023], (Research, 2022), [2024], (2011), etc.) by recognizing common citation format.
Step 2: Each citation number identified in last step is semantically matched with its surrounding text in the paper. An LLM [3] is prompted to extract the list of authors name associated with each citation. The result is being stored in structured format as JSON, mapping each citation with their respective authors.
Step 3: This step focuses on detecting the publication year for each citation. It uses both regular expressions and contextual LLM [3] prompting to extract years from text segments near citations. A basic timeline plot is generated using Matplotlib [6], visualizing the chronological distribution of the cited works. This helps identify trends such as the recency or historical spread of references used in the paper. The mapped citation with the year of publication is being stored in the JSON format.
Step 4: This agent uses the LLM [3] to generate concise two-line unique summaries for each citation, capturing the core concept of the cited work. The generated two-lined summaries are embedded into numerical vectors using the nomic-embed-text [8] model. Then, K-Means [7] clustering algorithm is used for grouping semantically similar citations together. The Elbow Curve Method is applied to determine the optimal number of clusters. Sub-clusters or topics within each cluster are inferred by prompting the LLM [3] to describe themes based on grouped summaries.
Step 5: The final step involves generating meaningful visualizations to represent citation relationships. An Interactive D3.js [2] Diagram representing the hierarchical structure of citation clusters and subtopics. A Fishbone Plot (using Matplotlib [6]) to show citation clusters with years.These visualizations aid in understanding how the cited literature is organized thematically.

Step	Prompt as input	Output format
Step I	Extract ONLY citation numbers from the academic text. Citations look like [1], [2], (3), (Smith et al., 2021), (2011), etc.	Return a plain JSON list of integers like: [1, 2, 3]. Do not return extra text or markdown.
Step II	You are given a section of an academic paper. Extract a list of properly formatted citation entries. Each entry must have: - The full author list	Return as JSON list. Format each citation like this: [Citation No.1, Author: Kumar R., Sharma T.] Only include entries with author. Do not return anything else.
Step III	Your task is to find the publication year for each in-text citation. Only return years if they are clearly associated with a citation. Skip missing or unclear entries.	Return the result in this JSON format: [Citation No.1, Year: 2023]. Do not include citations without a year. Skip missing or unclear entries.
Step IV	Given this chunk of a research paper, generate a unique 2-line summary for each citation. Do not repeat the same summary.	Format as a JSON list:[Citation No.1, summary: Researchers proposed a method that combined SVM and feature selection...]
Step V	You are a helpful assistant. You are given a group of research summaries, each prefixed by a citation number like [22]. Your task is to analyze this group and organize it into labeled subgroups based on theme, technique, or topic.	Return the result as a JSON object with structure like: [Main Theme of This Cluster: Subgroup A: [22, 23], Subgroup B: [24, 25]. Only use citation numbers in the output. Do not include summaries or explanations.

Table 1: Sample prompts used by each step in the multi-agent system

Tool / Library	Purpose / Functionality
LangChain [5]	Framework for building modular LLM [3] pipelines. Used for chaining PDF loaders, LLM [3] prompts, and output parsers.
Ollama [4] + Mistral [1]	Local LLM backend used for citation number extraction, metadata identification, and summarization.
nomic-embed-text [8]	Embedding model used to convert text summaries into numerical vectors for clustering.
scikit-learn	Library used for K-Means [7] clustering and Elbow method implementation.
Matplotlib [6]	Used to generate static citation timeline and fishbone visualizations.
D3.js [2]	JavaScript library for creating interactive hierarchical visualizations (e.g., citation tree maps).
Python	Programming language used to implement all backend logic and agents.
LaTeX + TikZ	Used for professional-quality documentation and creating custom flow diagrams.

Table 2: Overview of tools and libraries used in the citation analysis pipeline

Implementation

Clustering: K-Means Clustering Algorithm

- **Initialization:** Select the number of clusters k and randomly initialize k centroids.
- **Assignment Step:** Assign each data point to the nearest centroid based on the Euclidean distance.
- **Update Step:** Recalculate the centroids as the mean of all data points assigned to each cluster.
- **Repeat:** Repeat the assignment and update steps until the centroids no longer change significantly or a maximum number of iterations is reached.

Elbow Method Algorithm for Determining Optimal Clusters

- **Step 1:** Define a range for the number of clusters k (e.g., from 1 to 10).
- **Step 2:** For each value of k, apply the K-Means [7] clustering algorithm to the dataset.
- **Step 3:** Compute the Within-Cluster Sum of Squares (WCSS) - the total squared distance between each point and its assigned centroid.
- **Step 4:** Plot the WCSS against the corresponding number of clusters k.
- **Step 5:** Identify the "elbow point" in the plot - the point after which WCSS reduction becomes marginal and choose this k as the optimal number of clusters.

Result and Evaluation

Figure 4 shows a `2D t-SNE visualization` of citation clusters generated using `K-Means` [7] (with k=6). Each citation is represented by its author and year, and is color-coded by cluster ID. The plot reveals how semantically similar citations are grouped closely together in the embedded space. This is the representation of the diagram after storing the summaries of the citations and before creating the clusters.

`D3.js` [2] visualization (in Figure 2) illustrates the hierarchical structure of citation clusters formed from the academic paper. It represents the result of semantic clustering performed by Agent IV and post-processed topic-based subgrouping handled by Agent V using `LLM` [3] prompts. At the root, all citations are collectively represented. From there, the data is divided into three main clusters - Cluster 1, Cluster 2, and Cluster 3 - as determined by the `K-Means` [7] algorithm (with k=3 chosen based on the `elbow method`). Each cluster is further organized into subtopics or thematic branches, labeled based on the semantic content of the grouped citation summaries.

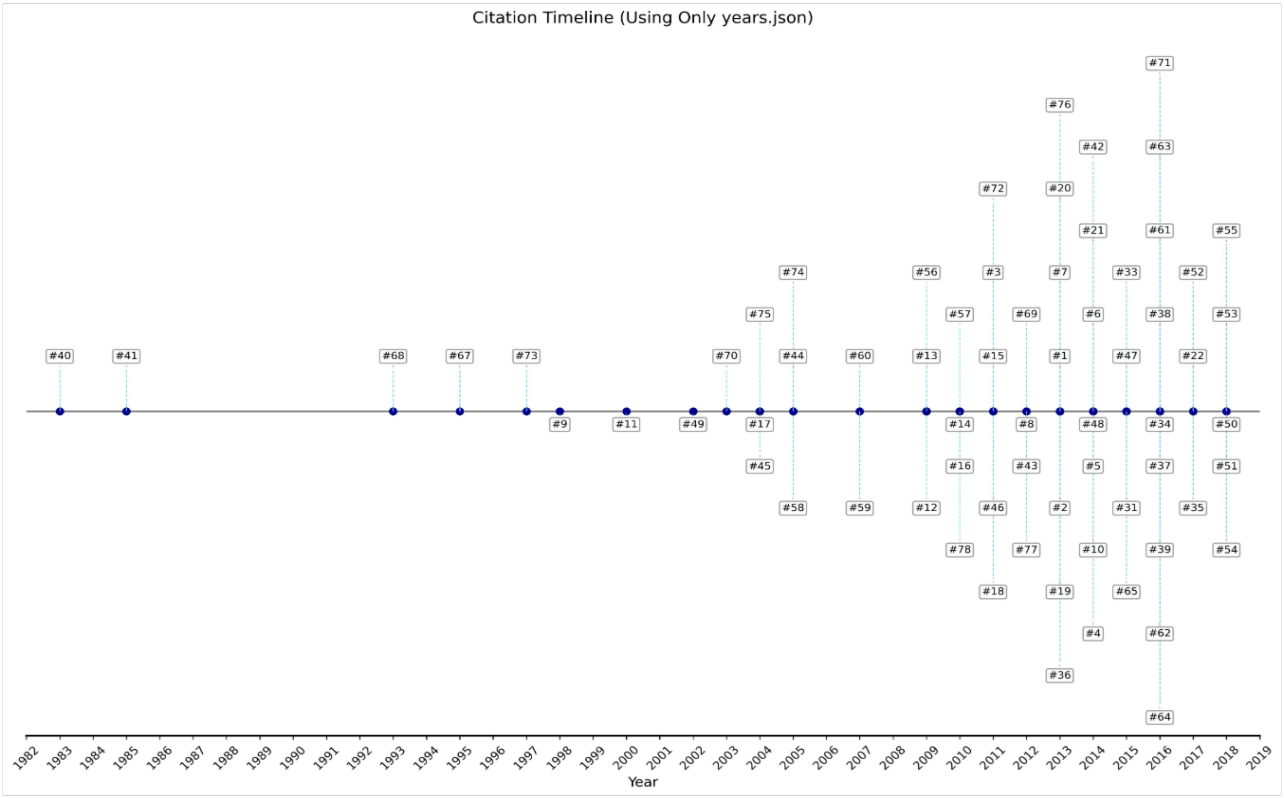

Figure 1: Citation distribution over years generated by Agent III

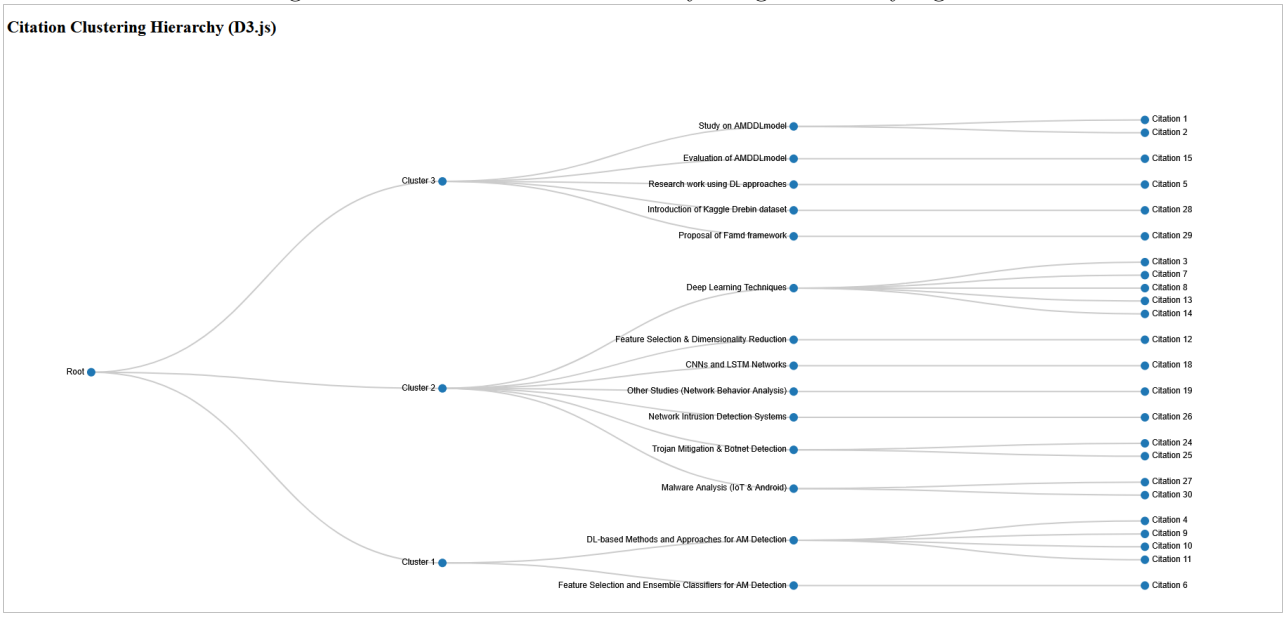

Figure 2: Citation Cluster Distribution Based on Semantic Similarity of Summaries

The timeline (in Figure 1) illustrates the chronological distribution of citations extracted from the academic research paper. Each blue dot represents a citation, plotted along the horizontal axis based on its publication year. The vertical dotted lines and labeled boxes indicate individual citation IDs. This visualization, generated by Agent III, helps identify the temporal spread of references, showing trends such as whether the paper relies more on recent literature or includes foundational work from earlier decades. As seen in the plot, the majority of citations cluster around the period from 2010 to 2018, suggesting that the research is grounded heavily in contemporary studies, while a few citations from the 1980s and 1990s highlight the inclusion of historically significant work.

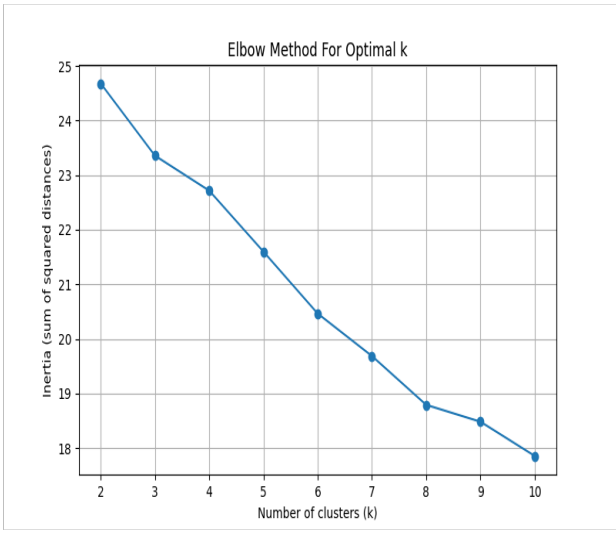

Figure 3: Finding the optimal number of clusters using elbow method

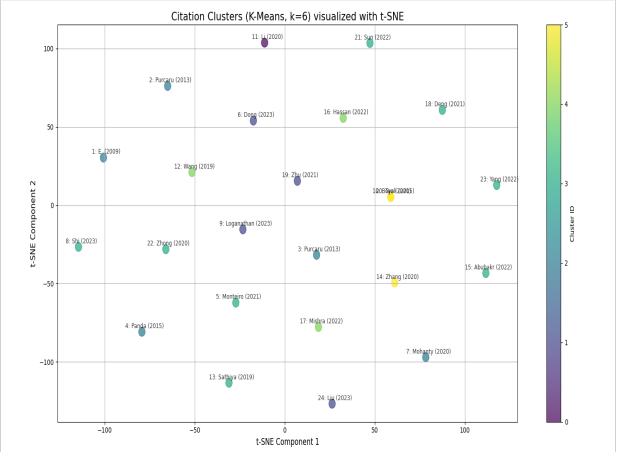

Figure 4: Diagrams of summaries before creating the clusters

Figure 3 shows the `Elbow Method` used to determine the optimal number of clusters for citation grouping. The curve suggests that the optimal value of k is around 3-4, where the reduction in inertia begins to level off.

Limitations and Challenges

- → The system currently supports research papers with up to 300 citations.
- → Only one PDF file can be processed at a time.
- → Processing is time-consuming, as most agents operate sequentially.
- → For better results, larger LLMs [3] can be used for text extraction and chunking.
- → Systems with high-performance `GPUs` yield faster and more optimal outcomes.
- → The system does not support citation extraction from documents with unstructured or poorly formatted references.
- → Large documents with extensive citations may cause memory issues on systems with limited RAM or VRAM.
- → Although the system uses well-crafted prompts, `LLMs [3] may still hallucinate` citation summaries or author names, especially when the context around a citation is vague, incomplete, or poorly formatted.
- → Only `English-language` papers are supported reliably; multilingual or non-English PDFs may lead to inconsistent output.
- → The generated `D3.js [2] visualizations` are not interactive within the LaTeX report and need to be viewed in a browser separately.
- → In different papers, the citations appear in various formats that is quite challenging to extract.
- → Some citations do not clearly mention the publication year or mention it far from the citation itself, leading to incomplete or inaccurate timeline visualization.
- → Verifying the correctness of extracted authors, summaries, and clusters manually is time-consuming and subjective.
- → Smaller local models sometimes lack the reasoning capacity for detailed extraction; larger ones perform better but are resource-heavy.
- → Large models provide better accuracy and deeper understanding, but are slower and require more computational resources.
- → Smaller models are faster and lightweight, but may sacrifice accuracy and context handling.

Future works

In the future, this system can be extended to support the analysis of `multiple PDFs` simultaneously, enabling broader research aggregation. Employing larger or `fine-tuned language models` could further enhance the accuracy of metadata extraction and summarization. Visualizations can be made more interactive and searchable for better user experience. Support for multilingual documents would also increase the applicability of the system globally. Performance improvements may be achieved by executing agents in parallel or asynchronously, reducing overall processing time. Lastly, incorporating mechanisms for user feedback could help refine and validate extracted citation data.

Conclusion

The proposed multi-agent system effectively automates citation extraction, analysis, and visualization from academic PDFs. It leverages LLMs [3] and clustering techniques to identify citation metadata and organize it thematically. While accurate and scalable, it faces limitations with unstructured formats and processing time. Overall, it offers a promising foundation for intelligent citation mapping and literature review support.

References

[1] Alexandre Sablayrolles. Albert Q. Jiang. "Mistral 7B". In: *Mistral.AI* (2023).

[2] Jeff Heer. ichael Bostock. and Vadim Adolp. "Data-Driven Documents". In: *D3.js is a JS library for creating interactive data visualizations using HTML, SVG, and CSS.* (2011). URL: https://d3js.org/.

[3] Subbiah M Brown T Ryder N and Kaplan J. "OpenAI". In: *Language Models are Few-Shot Learners* (2020). URL: https://python.langchain.com/docs/introduction/.

[4] Maximilian Weber. Johannes B. Gruber. "https://github.com/ollama/ollama". In: *rollama: An R package for using generative large language models through Ollama* (2024).

[5] Mehfuza Holia. Keivalya Pandya. "Automating Customer Service using LangChain". In: *Building custom open-source GPT Chatbot for organizations* (2023).

[6] *Matplotlib Documentation*. Matplotlib Development Team. 2024. URL: https://matplotlib.org/stable/contents.html.

[7] Guan yong Shi Na and Liu Xumin. *Research on k-means Clustering Algorithm*. 2010. DOI: DOI10.1109/IITSI.2010.74.

[8] Andriy Mulyar. Zach Nussbaum. John Xavier Morris and Brandon Duderstadt. "Nomic-Embed-Text". In: *Nomic Embed: Training a Reproducible Long Context Text Embedder* (2025).