LEVERAGING LARGE LANGUAGE MODELS FOR HOMELESS RESOURCE ACCESSIBILITY

By

Chenzi Xiu

A capstone project submitted for Graduation with University Honors

June 4, 2026

University Honors

University of California, Riverside

APPROVED

Dr. Yuzhou Chen

Department of Statistics

Dr. Begoña Echeverria, Howard H Hays, Jr. Chair

University Honors

ABSTRACT

Homelessness continues to be a major social issue in the United States, and one practical challenge is that people who need help often have difficulty finding clear and reliable information about available services. Food assistance, shelter programs, mental health support, and other community resources are often listed across different online directories, but these directories can be difficult to search quickly.

This capstone project developed a retrieval-based artificial intelligence system for organizing and searching homeless-support and community resource information. The system was built in Python using structured service data stored in CSV format. Service records were represented as subject–relationship–object triples, allowing each provider's name, service category, address, phone number, website, ZIP code, and hours of operation to be stored in a consistent format. The system used sentence-transformer embeddings and FAISS similarity search to match user questions with relevant service records. It also used ZIP code and city-based location cues to prioritize services relevant to the user's requested area.

Testing showed that the system could return multiple structured service recommendations for queries involving food assistance, location-based searches, and day-of-week availability. Instead of giving a single answer, the system presented several matching services with practical details such as address, phone number, website, and operating hours. Although the system depends on the accuracy and completeness of the underlying dataset, the project demonstrates how retrieval-based AI methods can make community resource information more organized, searchable, and accessible.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my faculty mentor, Dr. Yuzhou Chen, for his guidance, technical support, and encouragement throughout this project. His feedback helped me refine the direction of the project and improve the system's organization, implementation, and final presentation.

I would also like to acknowledge Leo, a fellow University Honors student, who worked on related components within the broader project environment. His work focused on knowledge graph development and Neo4j integration, while my contribution focused on the retrieval-based service recommendation system, including structured triple organization, semantic search, location-based prioritization, and response formatting.

In addition, I would like to thank the University Honors Program at the University of California, Riverside, for providing academic support and resources throughout the capstone process.

TABLE OF CONTENTS

Introduction

Homelessness remains a major social problem in the United States. People experiencing homelessness often face barriers when trying to access basic services such as food assistance, temporary shelter, mental health support, transportation help, and case management. Although many service organizations publish information online, this information is often spread across different websites and directories. For users who need help quickly, it can be difficult to identify which services are relevant, where they are located, whether they are available, and how to contact them.

Existing service directories can be useful, but they often require users to search through long lists of results and interpret the information on their own. This process can be difficult for people who are unfamiliar with the service system, have limited internet access, or are under immediate stress. A person looking for help may not know the official category name of a service or may describe a need in everyday language. For example, a user may ask for “food help” rather than “food pantry,” or may ask for services “near Los Angeles” instead of searching by a specific provider name. In these situations, the problem is not only whether service information exists, but whether it can be searched and understood in a practical way.

Recent developments in natural language processing and retrieval-based artificial intelligence provide new ways to organize and search public service information. Instead of requiring users to browse many pages manually, a retrieval-based system can accept a plain-language question and return structured results from stored service records. This approach is especially useful when the system is grounded in an external dataset rather than relying only on a language model’s general knowledge. For public service navigation, grounded retrieval is important because users need concrete details such as addresses, phone numbers, websites, service categories, ZIP codes, and hours of operation. A fluent response is not enough if the information is incomplete, outdated, or unsupported by the dataset.

This project was developed as an alpha-stage prototype for homeless-support and community resource navigation. The goal was not to create a final production system, but to test a smaller and more stable retrieval pipeline that could organize service data and return readable recommendations. The system was built in Python using service records stored in CSV format and organized as subject–relationship–object triples. It used sentence-transformer embeddings and FAISS similarity search to match user questions with relevant service records. It also used ZIP code and city-based cues to prioritize results connected to the user's requested area.

The main objective of this project was to examine whether structured service data and semantic retrieval could make community resource information easier to search and compare. By converting service records into a searchable triple-based format, the system could return multiple results with practical fields such as service name, address, phone number, website, availability status, and hours. This design supports the broader goal of improving access to community resources for people who may need fast, clear, and usable service information.

Literature Review

Artificial intelligence has increasingly been used to improve information search, recommendation systems, and user interaction. For public service access, these technologies are useful because users are often not looking for broad explanations. They usually need specific, practical information, such as where a service is located, what type of help it provides, whether it is open, and how to contact the provider. Individuals experiencing homelessness or housing insecurity may need to locate food assistance, shelter, mental health support, transportation help, or case management quickly. In this context, the main challenge is not only storing service information, but making that information searchable and understandable for people who may be under stress or unfamiliar with service directories.

Traditional keyword-based search can be limited in this type of setting. Service records may use formal category labels, while users may describe their needs in informal language. A database may use the phrase "food pantry," but a user may search for "food help," "free groceries," or "something to eat." Similarly, a record may list "behavioral health" or "counseling," while a user may ask for "mental health support." These differences in wording can make exact keyword search less effective. This project therefore builds on research in artificial intelligence, natural language processing, retrieval-based systems, and human-computer interaction to develop a system for homeless-support and community resource navigation.

One important issue in AI-supported service systems is reliability. Bishara et al. (2022) discuss AI Quality Improvement and emphasize that AI systems require monitoring and updating when external conditions change. Although their work focuses on clinical artificial intelligence, the same concern applies to public service navigation. Service information such as phone numbers, addresses, hours, eligibility rules, and availability can change frequently. If a system provides outdated information, the output may be misleading even if the search method itself works correctly. This supports the design choice in the present project to ground responses in a structured external dataset rather than relying only on a language model's general knowledge.

Related work also shows the importance of data systems that can adapt to updated information. Alsulami et al. (2022) examine data generation and updating for networking AI algorithm development. While their research is in a different domain, it highlights a broader principle: AI systems are more useful when their data sources can be updated and reused without rebuilding the whole system. This is relevant to community resource navigation because service directories are not static. New services may be added, programs may close, and hours of operation may change. A retrieval-based system can better support this kind of changing information because the external dataset can be revised while the overall search pipeline remains the same.

Conversational AI research provides another foundation for this project. Inkster et al. (2018) study Wysa, a conversational AI agent designed to support mental well-being. Their work shows that natural language interfaces can lower barriers for users who may not know exactly how to search for help. This is important for homeless-support systems because users may describe their needs in everyday language rather than formal service categories. A system that accepts plain-language questions can make service information easier to access, especially when users do not know the official name of a program or provider.

Human-computer interaction research also shows that usability affects whether technical systems are actually helpful. Alkatheiri (2022) discusses how AI-assisted human-computer interaction can improve communication between users and computer systems. This is relevant for users who may have limited time, limited internet access, or limited familiarity with service directories. In this project, usability is addressed through structured output formatting. Instead of returning raw database entries or long unorganized text, the system presents service name, address, ZIP code, phone number, website, availability, and hours in a consistent format. This design choice reflects the idea that useful information should be easy to compare, not just technically available.

The technical foundation of modern language systems begins with transformer-based models. Vaswani et al. (2017) introduced the Transformer architecture, which uses attention mechanisms to process text more efficiently than earlier recurrent models. This architecture became a foundation for many modern language and embedding models. Although the present project does not rely only on a large language model, transformer-based research is still relevant because it supports the semantic representation methods used in retrieval systems. These representation methods allow text to be compared by meaning rather than by exact word overlap.

Sentence embeddings are especially important for retrieval-based applications. Reimers and Gurevych (2019) introduced Sentence-BERT, which creates meaningful sentence-level embeddings that can be compared using similarity measures. This is useful when a user's wording does not exactly match the wording in the dataset. For example, a query about "mental health help" may need to match records labeled as "counseling," "behavioral health," or "crisis support." Semantic similarity search makes this matching more flexible than exact keyword search. For a service navigation system, this flexibility is important because users may not know the exact language used by service providers.

Retrieval-Augmented Generation also provides an important framework for this project. Lewis et al. (2020) introduced RAG as a method that combines language generation with external retrieval. Rather than generating responses only from model parameters, a RAG system retrieves relevant documents and uses them as the basis for an answer. This can reduce unsupported or hallucinated responses. In public service applications, grounding responses in data is especially important because users need accurate details such as addresses, phone numbers, eligibility rules, and operating hours. A response that sounds fluent but is not supported by the dataset could be harmful in a real service-access context.

Prior work on AI and system performance also helps explain the need for efficient retrieval. Kunduru (2023) discusses the use of artificial intelligence in improving cloud application performance. Although this source is not directly about homeless service navigation, it relates to the broader challenge of building systems that remain usable as data size increases. In this project, embedding-based retrieval and FAISS similarity search allow the system to search structured service records efficiently. This matters because community resource datasets can contain many records, and a user-facing system needs to return results quickly enough to be practical.

Another important design issue is how service information is represented. Unstructured text can contain useful details, but it is harder to search, filter, and format consistently. Structured records make it easier to separate service name, category, location, contact information, availability, and hours. In this project, service records were represented as subject–relationship–object triples. This format helped the system group related facts about the same provider and return results in a readable structure. The use of triples also made the retrieval process easier to explain and test because each recommendation was connected to stored fields in the dataset.

Together, these studies show that an effective AI system for public service navigation should combine reliable data, flexible search, and clear presentation. Transformer-based models and sentence embeddings support semantic matching, while retrieval-augmented methods show how external data can ground responses. Research on conversational AI and human-computer interaction shows that the system must also be understandable to nontechnical users. For homeless-support and community resource navigation, these factors are closely connected: a technically accurate system is still limited if the output is difficult to read, and a readable system is not useful if the information is not grounded in reliable records.

However, much existing research focuses on healthcare, mental health chatbots, cloud systems, or general natural language processing tasks. Less attention has been given to applying retrieval-based AI methods to homeless-support and community resource navigation. This project addresses that gap by organizing service records as subject–relationship–object triples and using semantic retrieval to return structured, location-relevant results. Rather than treating a language model as the only source of information, the system uses stored service data as the foundation for its responses. This approach fits the needs of public service navigation, where accuracy, clarity, and practical usability are more important than generating long conversational answers.

Methodology

This project developed a retrieval-based system for searching homeless-support and community resource information. The system was designed to help users find practical service records by entering plain-language questions about needs such as food assistance, shelter, mental health support, or location-based services. Rather than relying only on a language model's general knowledge, the system used structured service data as the main source of information. This design was selected because public service recommendations require grounded details such as addresses, phone numbers, websites, ZIP codes, service categories, and hours of operation.

The project was developed as an alpha-stage prototype within a broader homeless resource navigation project. Since the larger system was still under development, this capstone focused on building a smaller and more stable retrieval pipeline that could be tested and demonstrated reliably. The goal of the prototype was not to replace a full production system, but to evaluate whether structured service records could be converted into readable answers through semantic retrieval and organized output formatting. This narrower scope made it possible to test the system with fewer moving parts and fewer API-related failures.

The system was implemented in Python and used several supporting libraries for data processing and retrieval. The service data was stored in a CSV file named Triples.csv. Each row in the file contained three main columns: Subject, Relationship, and Object. The subject usually represented the name of a service provider or program. The relationship described the type of information being recorded, such as main_services_are, phone_number_is, website_is, is_located_at, or zipcode_is. The object contained the actual value associated with that relationship. This structure allowed service information to be stored as subject–relationship–object triples.

The triple-based format was useful because many service providers have multiple pieces of information that need to be kept together. For example, one service record may include a service category, address, ZIP code, phone number, website, eligibility description, availability status, and hours of operation. If these details are treated as separate unrelated rows, the system may retrieve incomplete information. By using triples, the system could preserve different facts about the same provider while still keeping the data structured. This helped the system produce results that were easier to read and compare.

Before retrieval, the system grouped triples by subject. This means that all available facts about the same service provider were combined into a subject-level document. Each document began with the service name and then listed related fields from the dataset. This step was important because users are usually looking for complete service recommendations, not individual database rows. By grouping triples into service-level records, the system could return a recommendation that included several useful details about the same provider instead of returning only one isolated fact.

The data preparation process also helped make the retrieval process more consistent. Empty values were handled so that missing fields would not break the system. Duplicate or repeated relationship-object pairs were avoided when building subject-level documents. The system also limited the number of facts included for each subject so that very large records would not create overly long context blocks. These choices were made to keep the retrieval process stable and to make the final output easier to format.

Semantic retrieval was implemented using the sentence-transformers library. The embedding model converted each subject-level service document into a numerical vector representation. User questions were also converted into vector representations. The system then compared the question embedding with the stored service embeddings to identify records with similar meanings. This allowed the system to match flexible user wording with relevant service records. For example, a user asking for "food help" could still be matched with records labeled as "food pantry" or other food assistance services.

This semantic matching approach was selected because exact keyword search would be too limited for this type of system. Users may describe the same need in different ways, and service records may use formal labels that do not match everyday language. A person looking for mental health support may not use the same words as the database field. Similarly, a person looking for food assistance may not know whether the service is listed as a pantry, meal program, or community resource. Embedding-based search helps reduce this wording gap by comparing records based on meaning rather than exact word overlap.

To make retrieval efficient, the system used FAISS similarity search. After the service documents were embedded, the vectors were stored in a FAISS index. When a user entered a question, the system searched the index and returned the top matching service records. The system used a top-k retrieval approach, returning multiple matches rather than a single result. Returning several options is useful in a public service context because users may need to compare location, hours, contact information, and availability before choosing a service.

The system also included location-based prioritization. When a user entered a ZIP code or city name, the system extracted those location cues from the question and checked whether the retrieved service records contained matching location information. Records that matched the requested ZIP code or city cue were prioritized ahead of less location-relevant records. This method was not a full geographic distance calculation. It did not compute travel distance or use latitude and longitude to rank nearby services. However, it improved the usefulness of search results by giving preference to services connected to the user's requested area.

The stable version of the system used local retrieval and structured output formatting. After relevant service records were retrieved, the system formatted the results into a readable list. Each result could include the service name, address, ZIP code, service type, phone number, availability status, hours of operation, and website when those fields were available in the dataset. This local formatting approach allowed the system to operate without depending on an external API during the presentation version. This was important because external generation tools can be unreliable during high-traffic periods or when API access is limited.

An optional Gemini generation component was included in the code, but it was not the main source of the system's answers. The main design remained grounded in local retrieval over the structured CSV dataset. When local-only mode was used, the system still returned readable recommendations by extracting and formatting fields from the retrieved records. This design reduced dependence on external model availability and made the demonstration version more stable. It also helped keep the system's answers closer to the stored data instead of relying on unsupported generation.

Several design choices were made to improve reliability. First, the system answered based on records retrieved from Triples.csv rather than generating unsupported service information. Second, the system returned multiple results so users were not limited to one recommendation. Third, the output was organized into practical fields instead of long paragraphs, making it easier for users to compare services. Fourth, the system exposed important details such as hours and websites so that users could verify information before visiting or contacting a service provider.

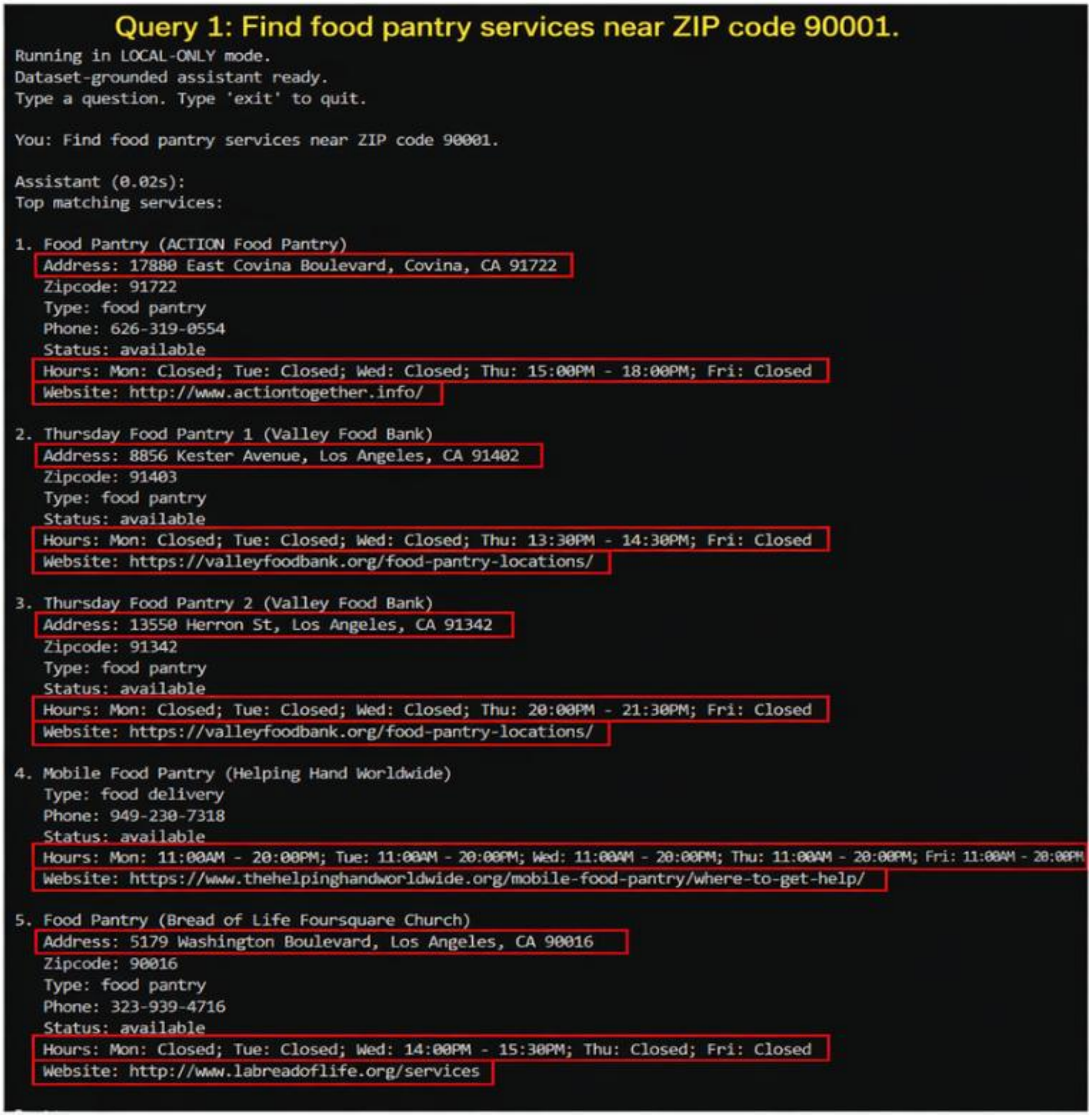
The methodology also reflected the limits of the alpha-stage prototype. The system was able to retrieve and format useful results, but it was not designed as a complete production-level recommendation engine. Location matching was based on ZIP code and city cues rather than full distance ranking. Day-of-week availability was displayed in the output, but strict filtering by operating hours required further improvement. These limitations were expected at the prototype stage and helped identify the next steps needed for a more complete version of the system.

Overall, the methodology focused on building a stable retrieval pipeline that connected structured service data with user-friendly output. The combination of triple-based data organization, sentence-transformer embeddings, FAISS similarity search, location-based prioritization, and local structured formatting allowed the system to turn service records into readable recommendations. This approach supported the main goal of the project: making community resource information easier to search, understand, and compare for users seeking homeless-support and related public services.

Results and Discussion

The first test evaluated the system's ability to retrieve food assistance resources from a ZIP code-related query. The user entered the query: "Find food pantry services near ZIP code 90001." The system processed the query by converting it into an embedding representation and comparing it with stored service-record embeddings. After retrieval, the system returned multiple food pantry-related records and formatted each result with available fields such as service name, address, ZIP code, service type, phone number, availability status, hours of operation, and website.

The results showed that the system was able to identify records related to the requested service type and present them in a readable format. However, the test also showed that the location component should be understood as location-based prioritization rather than strict distance filtering. Some returned services were located outside the exact ZIP code entered by the user. This means the system was effective at retrieving food pantry services, but it did not fully limit the results to ZIP code 90001 or calculate geographic distance. Figure 1 shows an example output generated by the system in response to this ZIP code-based query.



The second test evaluated the system's response to a day-of-week availability query. The user entered the query: "Find services open on Monday in Los Angeles." The system processed the query using semantic similarity matching and returned Los Angeles-area service records with structured fields such as address, ZIP code, service type, phone number, availability status, hours, and website. Figure 2 shows an example output generated by the system for this query.

The results showed that the system could display operating-hour information in a readable format, which allows users to compare whether services are open on a specific day. However, the test also revealed a limitation in the filtering logic. Some returned records displayed “Mon: Closed,” which means the system did not fully exclude services that failed the Monday availability condition. Instead, the system retrieved relevant service records and exposed the hours field for user review. This result suggests that the system’s formatting was useful, but future versions should include stricter day-of-week filtering before returning final recommendations.

```
Query 2: Find services open on Monday in Los Angeles.
Running in LOCAL-ONLY mode.
Dataset-grounded assistant ready.
Type a question. Type 'exit' to quit.

You: Find services open on Monday in Los Angeles.

Assistant (0.01s):
Top matching services:

1. Emergency Services (Los Angeles Mission, Inc)
   Address: 303 East 5th Street, Los Angeles, CA 90013
   Zipcode: 90013
   Type: temporary shelter
   Phone: 213-629-1227
   Status: available
   Hours: Mon: 11:00AM - 19:30PM; Tue: 11:00AM - 19:30PM; Wed: 11:00AM - 19:30PM; Thu: 11:00AM - 19:30PM; Fri: 11:00AM - 19:30PM
   Website: https://losangelesmission.org/emergency-services/

2. Wednesday Food Pantry 2 (Valley Food Bank)
   Address: 18120 Saticoy St, Los Angeles, CA 91335
   Zipcode: 91335
   Type: food pantry
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: 16:00PM - 18:00PM; Thu: Closed; Fri: Closed
   Website: https://valleyfoodbank.org/food-pantry-locations/

3. Wednesday Food Pantry 1 (Valley Food Bank)
   Address: 8767 Woodman Ave, Los Angeles, CA 91331
   Zipcode: 91331
   Type: food pantry
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: 14:00PM - 16:00PM; Thu: Closed; Fri: Closed
   Website: https://valleyfoodbank.org/food-pantry-locations/

4. Thursday Food Pantry 2 (Valley Food Bank)
   Address: 13550 Herron St, Los Angeles, CA 91342
   Zipcode: 91342
   Type: food pantry
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: Closed; Thu: 20:00PM - 21:30PM; Fri: Closed
   Website: https://valleyfoodbank.org/food-pantry-locations/

5. Food Pantry (Del Rey Church (DRC))
   Address: 8595 Saran Drive, Los Angeles, CA 90293
   Zipcode: 90293
   Type: food pantry
   Phone: 310-823-4275
   Status: available
   Hours: Mon: 12:00PM - 20:00PM; Tue: 12:00PM - 20:00PM; Wed: 12:00PM - 20:00PM; Thu: 12:00PM - 20:00PM; Fri: 12:00PM - 20:00PM
   Website: https://www.delreychurch.com/food-pantry
```

An important feature of the system is its ability to return multiple service recommendations instead of a single output. Returning several results gives users more flexibility, especially when one provider is closed, too far away, or not appropriate for the user's needs. The system ranked results based on semantic similarity between user queries and stored service records. This ranking method allowed relevant services to appear near the top while still giving users multiple options to compare.

Several strengths were observed during system testing. First, the system successfully handled plain-language queries and returned structured service information. This improves accessibility because users are not required to search using exact database terms. Second, the triple-based data structure helped organize service information into consistent fields, including address, ZIP code, phone number, website, service type, availability status, and hours of operation. Third, the local retrieval and formatting approach allowed the system to produce useful outputs without depending on an external API during the stable presentation version. Together, these strengths show that the system can make community resource information easier to search, read, and compare.

At the same time, testing revealed several limitations. The location component did not perform full geographic distance calculation, so results were not always limited to the exact ZIP code or closest service location. The day-of-week query also showed that the system did not fully exclude records that were closed on the requested day. Instead, it displayed the hours field so users could review availability themselves. Another limitation is that the system depends on the quality and completeness of the dataset. If service records are outdated, missing fields, or formatted inconsistently, the output may be incomplete or misleading.

Future improvements should include stricter ZIP code filtering, true distance-based ranking using geographic coordinates, and more precise day-of-week filtering before results are returned. The system could also be improved through regular dataset updates and more advanced ranking rules that consider service type, location, availability, and user need at the same time.

Conclusion

This project developed a retrieval-based recommendation system designed to help users search homeless-support and community resource information. The system combined structured triple-based service records with semantic retrieval and structured output formatting. By organizing service data into subject–relationship–object triples, the system was able to retrieve and display related service attributes such as address, ZIP code, service type, phone number, website, availability status, and hours of operation.

Testing showed that the system could return multiple relevant service recommendations for plain-language queries involving food assistance, Los Angeles-area services, ZIP code references, and operating-hour information. These results suggest that semantic retrieval can make community resource information easier to search than a traditional keyword-based or manually browsed directory. Returning multiple results also gives users more flexibility when comparing service options. This is important in a service-navigation setting because a single result may be closed, too far away, missing contact information, or not appropriate for the user's specific need.

The project also revealed several limitations. The location component worked as ZIP code and city-based prioritization rather than full geographic distance calculation, so results were not always limited to the closest service locations. The day-of-week query showed that the system could display hours clearly, but it did not always exclude services that were closed on the requested day. In addition, the system depends on the accuracy and completeness of the underlying dataset. If service records are outdated, missing, or inconsistently formatted, the quality of the recommendations may be affected.

These limitations show that the system should be understood as an alpha-stage prototype rather than a final public-facing tool. The prototype was useful because it demonstrated that structured service records could be converted into readable recommendations through semantic retrieval. At the same time, the results show that any real-world version would need stronger validation before being used by people seeking urgent support. In public service contexts, incorrect hours, outdated phone numbers, or poorly ranked results could create real barriers for users.

Future work should focus on improving filtering and ranking logic. This includes stronger ZIP code filtering, true distance-based ranking using geographic coordinates, stricter day-of-week availability filtering, and regular dataset updates. Additional improvements could include user feedback collection, clearer error handling, and expanded coverage across more regions and service categories. The system could also include warning messages encouraging users to verify hours and availability directly with service providers before visiting.

Overall, this project demonstrates that structured service data and semantic retrieval can support more accessible community resource navigation. While the system is not a fully optimized recommendation engine, it provides a practical foundation for using retrieval-based AI methods to organize service information and present it in a clear, usable format for people seeking support.

References

Alkatheiri, M. S. (2022). Artificial intelligence assisted improved human-computer interactions for computer systems. *Computers and Electrical Engineering, 101*, 107950. https://www.sciencedirect.com/science/article/pii/S0045790622002282

Alsulami, K., Ye, F., & Zhang, J. (2022). Improvement on a traffic data generator for networking AI algorithm development. *IEEE Xplore*. https://ieeexplore.ieee.org/document/9685616

Bishara, A., Celi, L. A., Feng, J., Hubbard, A. E., Malencia, I., & Phillips, R. V. (2022, May 31). Clinical artificial intelligence quality improvement: Towards continual monitoring and updating of AI algorithms in healthcare. *Nature*. https://www.nature.com/articles/s41746-022-00611-y

Inkster, B., Sarda, S., & Subramanian, V. (2018). An empathy-driven, conversational artificial intelligence agent (Wysa) for digital mental well-being: Real-world data evaluation mixed-methods study. *JMIR mHealth and uHealth, 6*(11), e12106. https://mhealth.jmir.org/2018/11/e12106

Kunduru, A. R. (2023). Artificial intelligence usage in cloud application performance improvement. *Central Asian Journal of Mathematical Theory and Computer Science*. https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS/article/view/491

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3982–3992. https://arxiv.org/abs/1908.10084

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. https://papers.nips.cc/paper/7181-attention-is-all-you-need

Appendix

```
Query 1: Find food pantry services near ZIP code 90001.
Running in LOCAL-ONLY mode.
Dataset-grounded assistant ready.
Type a question. Type 'exit' to quit.

You: Find food pantry services near ZIP code 90001.

Assistant (0.02s):
Top matching services:

1. Food Pantry (ACTION Food Pantry)
   Address: 17880 East Covina Boulevard, Covina, CA 91722
   Zipcode: 91722
   Type: food pantry
   Phone: 626-319-0554
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: Closed; Thu: 15:00PM - 18:00PM; Fri: Closed
   Website: http://www.actiontogether.info/

2. Thursday Food Pantry 1 (Valley Food Bank)
   Address: 8856 Kester Avenue, Los Angeles, CA 91402
   Zipcode: 91403
   Type: food pantry
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: Closed; Thu: 13:30PM - 14:30PM; Fri: Closed
   Website: https://valleyfoodbank.org/food-pantry-locations/

3. Thursday Food Pantry 2 (Valley Food Bank)
   Address: 13550 Herron St, Los Angeles, CA 91342
   Zipcode: 91342
   Type: food pantry
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: Closed; Thu: 20:00PM - 21:30PM; Fri: Closed
   Website: https://valleyfoodbank.org/food-pantry-locations/

4. Mobile Food Pantry (Helping Hand Worldwide)
   Type: food delivery
   Phone: 949-230-7318
   Status: available
   Hours: Mon: 11:00AM - 20:00PM; Tue: 11:00AM - 20:00PM; Wed: 11:00AM - 20:00PM; Thu: 11:00AM - 20:00PM; Fri: 11:00AM - 20:00PM
   Website: https://www.thehelpinghandworldwide.org/mobile-food-pantry/where-to-get-help/

5. Food Pantry (Bread of Life Foursquare Church)
   Address: 5179 Washington Boulevard, Los Angeles, CA 90016
   Zipcode: 90016
   Type: food pantry
   Phone: 323-939-4716
   Status: available
   Hours: Mon: Closed; Tue: Closed; Wed: 14:00PM - 15:30PM; Thu: Closed; Fri: Closed
   Website: http://www.labreadoflife.org/services
```

Figure 1. Example System Output for Food Pantry Query

Example output showing multiple food pantry-related records retrieved from a ZIP code-based query.

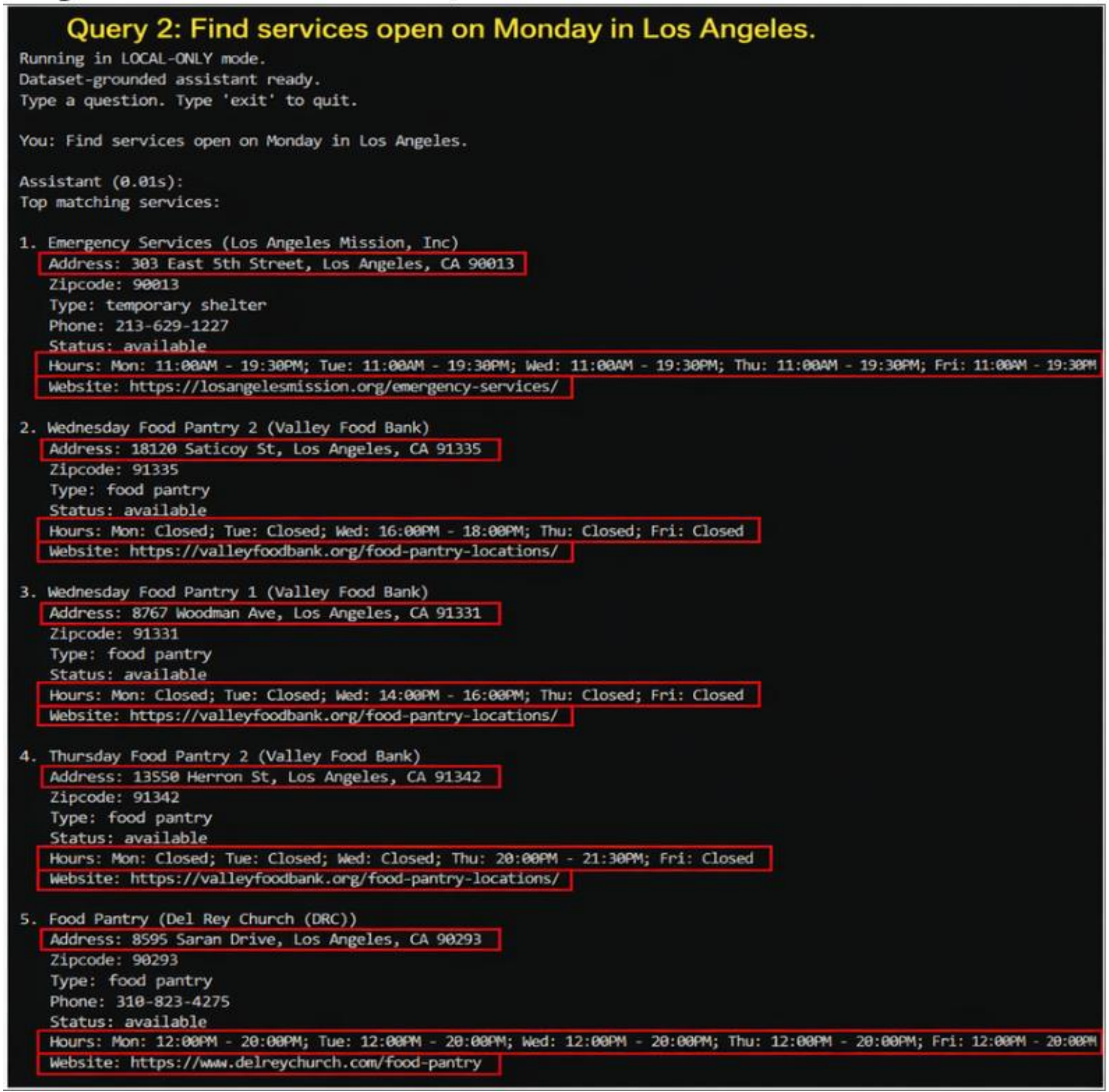


Figure 2. Example System Output for Time-Based Service Query

Example output showing service records with displayed Monday availability information in Los Angeles.