nourish

Food Knowledge Graph Analysis

Abstract

The process of operating a food-related business is complex and requires in-depth knowledge of many factors, including local policies/regulations, supply chains, sources of funding, and more. These complicated factors have made the business of food a very difficult field to work in, for both new and existing professionals. To help with this effort, a comprehensive food business knowledge graph and chat based user interface, Nourish, was created in an attempt to reduce the barrier of finding the right information to successfully operate in the food industry. The knowledge graph was developed by integrating multiple data sources to address each component of the food industry. These data sources included geographic, document, ontological and relational data. For geographic data, the knowledge graph utilized Arcgis and Esri to pinpoint optimal business locations (based on a linear programming scoring algorithm) which could be suggested to the end user. For the document data, document stores were created to provide users with funding information from various government institutions, such as the Small Business Administration (SBA) and the United States Department of Agriculture (USDA). For the ontological data, various foundational ontologies including FOODON and BFO were integrated into a graph database. For relational data, the USDA Food Data Central database was also incorporated to better understand nutritional alternatives and increase accessibility to healthy options. To enable widespread access to the knowledge graph, Nourish was connected to Open AI large language model (LLM) GPT-3.5, which provided a user-friendly way to query the knowledge graph. To transform user queries with GPT-3.5 into responses, the ReAct conversational agent chain was implemented using the Langchain framework as an interface. The agent was composed of several tools to address user inputs. For example, the location tool was used to suggest optimal locations for a business. Another tool queried the document indexes created on the document stores to address loan eligibility and general loan inquiries. The Nourish chatbot was displayed through a Dash app to facilitate conversation with the user. Overall, the Nourish chatbot effectively queried the integrated knowledge graph to provide users with personalized recommendations on vital information such as where to open their business, what loans they could apply for, and where they could find additional support.

Introduction and Question Formulation

Navigating the food industry poses significant challenges for small business entrepreneurs. They must be able to correctly identify potential markets, find efficient supply chains, abide by local laws and regulations, and locate the best sources of funding, among other tasks. The required information is scattered across numerous disconnected data systems, making it difficult to access the necessary details. Furthermore, professionals often lack the means to search across these data systems for the specific information they need. Recognizing this problem, the aim was to develop a platform capable of offering diverse recommendations derived from a single knowledge base through conversation.

The product, Nourish, will need to go beyond traditional single-topic recommendations. It must provide accurate and concise business recommendations encompassing a wide range of food-related topics. To achieve this, a robust knowledge base incorporating domain-specific knowledge, contextual knowledge, and language proficiency is essential. The research efforts led to the formulation of the following key data science questions:

1. Is it possible to provide enough domain-specific knowledge tailored to the application to mitigate the inaccuracies known as "hallucinations" that often occur with Large Language Models?
2. Is it possible to create a recommendation system that utilizes this knowledge base to deliver meaningful insights to users through conversation?

To scope this project, three areas were prioritized from the perspective of an entrepreneur:

1. Funding recommendations
2. Location-based recommendations
3. Healthier food alternatives

By addressing these questions and scope, the objective was to provide a comprehensive and effective solution for leveraging data science in the food industry.

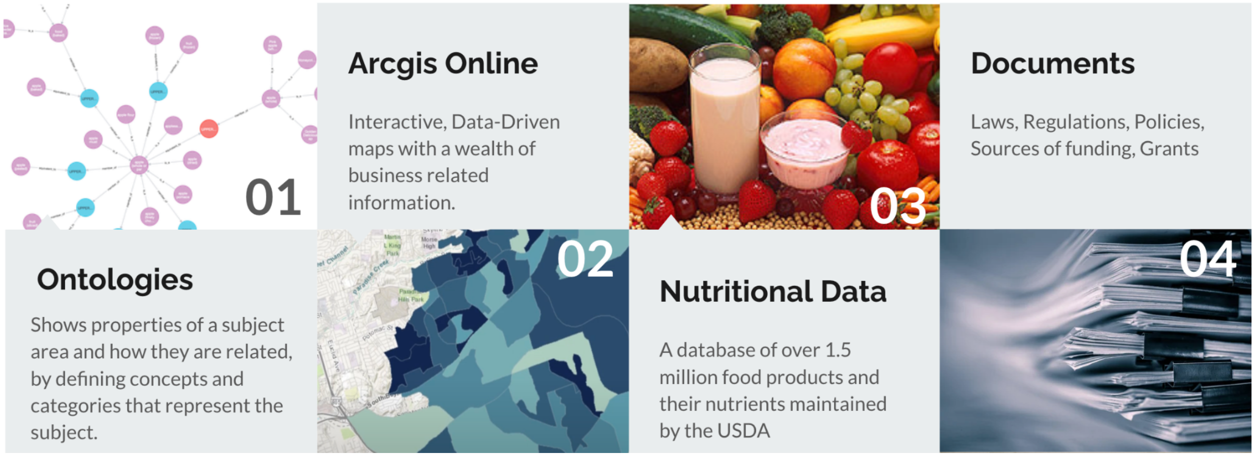
Team Roles and Responsibilities

Dr. Amarnath Gupta *Project Advisor*
Dr. Subhasis Dasgupta *Project Advisor*
Amol Kale *Data Engineer & Software Engineer*
Garrett Michael *ML Engineer & Product Manager*
Mona Henry *Data Engineer & UI Developer*
Matthew Stickle *Data Engineer & Software Engineer*
Jessica Allen *ML Engineer & AI Developer*

Data Acquisition

To establish the knowledge base, Nourish required a substantial amount of data encompassing a wide range of topics related to the food industry. Data sources were carefully selected that adequately covered the primary subjects addressed by in the recommendation system: location, laws, and nutrition. Four primary sources were chosen as the foundational components of the domain knowledge for Nourish.

1. Ontologies
2. ArcGis Online
3. Nutritional Data
4. Documents



Ontologies

Shows properties of a subject area and how they are related, by defining concepts and categories that represent the subject.

Arcgis Online

Interactive, Data-Driven maps with a wealth of business related information.

Nutritional Data

A database of over 1.5 million food products and their nutrients maintained by the USDA

Documents

Laws, Regulations, Policies, Sources of funding, Grants

Ontologies

Multiple ontologies were used to manage the complexity and organize the data into information and knowledge. These ontologies offered semantic richness, enabling the representation of intricate relationships and attributes. They formed the semantic framework that facilitated expressive modeling, effectively capturing the meaning and semantics within the food business domain. Furthermore, ontologies facilitated data integration and interoperability across diverse systems. Adopting standardized ontology languages such as OWL ensured seamless data sharing and integration. This aspect became crucial in the development of the final knowledge graph used by Nourish.

The primary ontologies employed included The Food Ontology (FoodOn) and The Financial Industry Business Ontology (FIBO). FoodOn provided comprehensive coverage of common food components and the associated manufacturing processes, while FIBO defined the relevant entities for financial business applications. These ontologies were complemented by semantic ontologies like The Basic Formal Ontology (BFO), Information Artifact Ontology

(IAO), and the Descriptive Ontology for Linguistic and Cognitive Engineering Ontology (DUL2). To best align with our project objectives, these ontologies were further enhanced by Dr. Amarnath Gupta.

Dataset Name	Source	Data Size
FoodOn	Downloaded	32,000 classes, 65 object properties, and 400 instances
FIBO	Downloaded	Thousands of classes across various models focused on specific domain of financial industry
Semantic	Internally generated from BFO, DUL2, and IAO	330 classes

ArcGis Online

ArcGIS Online is a cloud-based mapping and analysis platform developed by Esri, a leading provider of geographic information system software. ArcGIS enables users to combine spatial data and construct detailed, interactive spatial feature layers, which can be integrated into applications like Nourish. In order to create maps, data was extracted and combined to generate "Feature Layers" that support various scenarios within the user's conversation. These feature layers were created using business and demographic-related data found in ArcGis and stored in a PostgreSQL database and CSV files containing Census block group boundaries. The following serve as the primary sources of data.

Dataset Name	Source	Data Size	Description
California Businesses	Table stored in Postgres	515962 rows, 6 columns	A table of all businesses in California and their location
California Farms	Table stored in Postgres	638 rows, 5 columns	A table of all farms in California and their location

ESRI Variables	CSV files	2057 rows, 4241 columns	Various demographic, business and census data about block groups in San Diego
Farmers Market	Table stored in Postgres	1058 rows, 4 columns	A table of all farmers markets in California and their location

Nutritional Data

The USDA Food Data Central Database is a table-based data source that contains food label information for over 3 million food products, including the nutritional composition of each product. Having knowledge of the nutritional value of common food items enables the suggestion of healthier options and the grouping of similar foods based on nutritional content.

Dataset Name	Source	Data Size	Description
USDA Branded Food Database	Table stored in Postgres	3.2 Million Records	A table of common grocery items and their nutrition composition

Documents

The last type of data acquired were documents which covered topics from laws, regulations, policies, and funding. These are text based data sources and are stored on a shared Google Drive that Nourish can access.

Data Preparation

Nourish requires three forms of data for its recommendation system: domain knowledge, contextual knowledge, and language knowledge to effectively interact with the user. These three components form the foundation of Nourish's knowledge base. Building a robust knowledge base is crucial to reducing inaccuracies as reported by the LLM and enhancing detailed and helpful recommendations.

Domain Knowledge

The four data sources mentioned above underwent cleaning, feature extraction, and transformation processes to effectively support the recommendation system, ultimately forming the domain knowledge for Nourish.

To establish a comprehensive source of domain knowledge, a knowledge graph was constructed between the ontologies and the USDA data. The Food Ontology and USDA food products were linked using an open-source GitHub repository called LexMapr. The various ontologies obtained from Ontobee were merged into a single OWL file using Protege, an open-source ontology editor. Subsequently, a custom script utilizing OwlReady2 parsed the class information and relationships from the OWL file. The output of this script was a valid Neo4j node and relationship file.

The script traverses recursively over entities and properties in the merged ontology OWL file, deconstructing the ontology relationships into directional edges. To preserve the verbosity of relationships, placeholder conjunction nodes such as "AND," "OR," "BLANK," and "NOT" were introduced to represent complex ontological logic. The final node file contained approximately 45,000 entities, and the final edges file contained 60,000 edges. Both the node and relationship files were imported into Neo4j, a graph database platform that could be used for querying by the LLM.

Representation of ontology item	Neo4j translation using conjugations

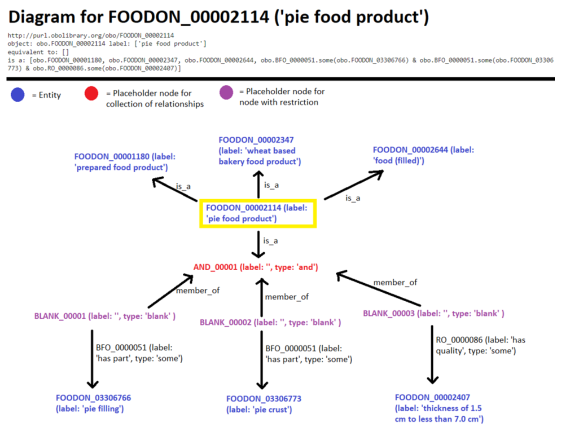
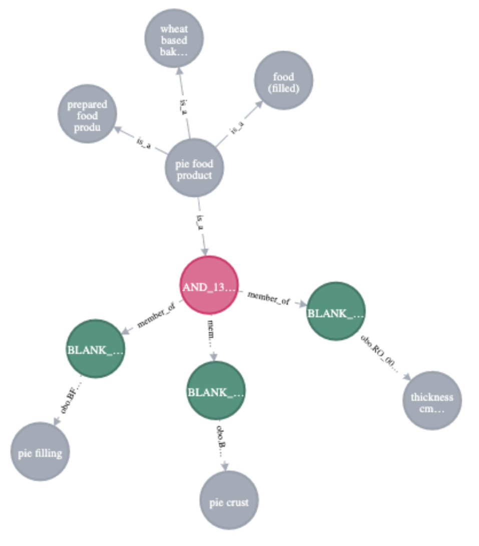
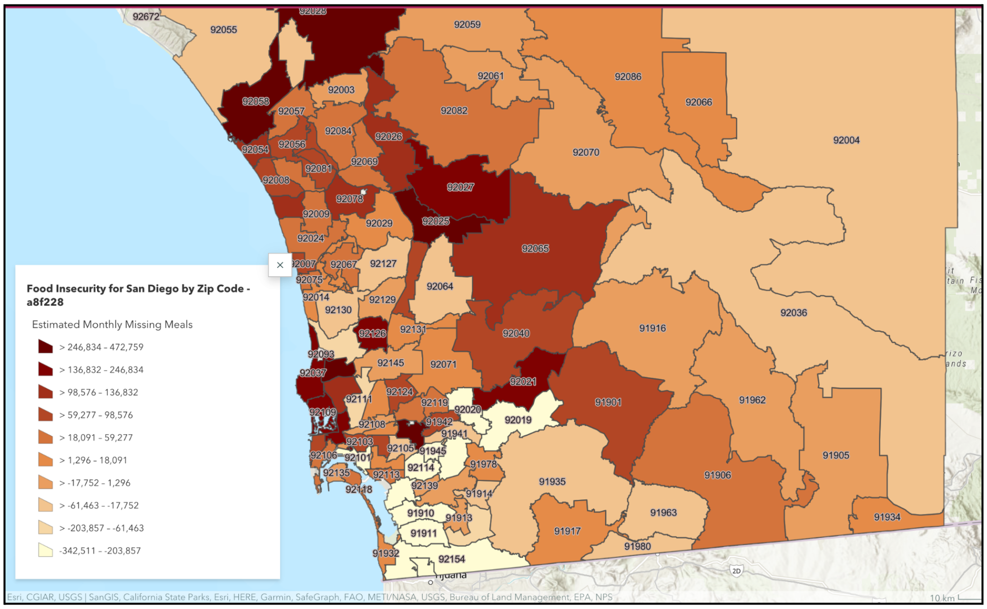


Diagram for FOODON_00002114 ('pie food product')
`http://purl.obolibrary.org/obo/FOODON_00002114`
`object: obo.FOODON_00002114 label: ['pie food product']`
`equivalent to: []`
`is a: [obo.FOODON_00001180, obo.FOODON_00002347, obo.FOODON_00002644, obo.BFO_0000051.some(obo.FOODON_03306766) & obo.BFO_0000051.some(obo.FOODON_03306773) & obo.RO_0000086.some(obo.FOODON_00002407)]`
 ● = Entity
● = Placeholder node for collection of relationships
● = Placeholder node for node with restriction
FOODON_00001180 (label: 'prepared food product')
FOODON_00002347 (label: 'wheat based bakery food product')
FOODON_00002644 (label: 'food (filled)')
is_a
is_a
is_a
FOODON_00002114 (label: 'pie food product')
is_a
AND_00001 (label: '', type: 'and')
member_of
member_of
member_of
BLANK_00001 (label: '', type: 'blank')
BLANK_00002 (label: '', type: 'blank')
BLANK_00003 (label: '', type: 'blank')
BFO_0000051 (label: 'has part', type: 'some')
BFO_0000051 (label: 'has part', type: 'some')
RO_0000086 (label: 'has quality', type: 'some')
FOODON_03306766 (label: 'pie filling')
FOODON_03306773 (label: 'pie crust')
FOODON_00002407 (label: 'thickness of 1.5 cm to less than 7.0 cm')
	 wheat based bak...
	food (filled)
	prepared food produ
	is_a
	is_a
	is_a
	pie food product
	is_a
	AND_13...
	member_of
	member_of
	mem...
	BLANK_...
	BLANK_...
	BLANK_...
	obo.BF...
	obo.B...
	obo.RO_00...
	pie filling
	pie crust
	thickness cm...

To further enhance Nourish's domain knowledge, the ArcGis data was cleaned and organized. There were well over five thousand demographic and business variables which made feature selection inevitable. Through manual selection and suggestions from Prof. Tera Fazzino (Univ. of Kansas), the most relevant features were extracted from the data, grouped together, and stored in feature layers. The values for the selected variables are aggregations by block-groups, which are regions smaller than zip codes that allow for detailed summaries of small clusters of city-blocks. However, because these ESRI variables were stored via block-group there was no clear way to associate them with human readable locations. This posed a problem as generally a human would not refer to a region as a block-group number but rather the name of a neighborhood or region in a conversation. To overcome this, the geocoding module from the ArcGIS API is used to determine the latitude and longitude coordinates of a neighborhood. Using these coordinates, a spatial intersection is performed with block group boundaries to dynamically establish mappings while the application queries spatial data related to block groups and neighborhoods. This approach allows for efficient correlation and analysis of spatial information, facilitating seamless integration between the two datasets on-the-fly.



Esri, CGIAR, USGS | SanGIS, California State Parks, Esri, HERE, Garmin, SafeGraph, FAO, METI/NASA, USGS, Bureau of Land Management, EPA, NPS

Since one of the primary goals of Nourish is to recommend where a user should start or expand a business, a method for suggesting locations with the highest opportunity was needed. In order to do so, an opportunity score was created for each census block group as a high overview summary. The opportunity score was based on eight values:

Name	Description	ESRI Variables	Calculation
Demographics	Understanding the target base and selecting most relevant demographic profile	● Divindx_cy: Diversity Index	DIVINDX_CY
Population Density	Selecting a location with a sufficient population density to support the customer base	● Popdens_cy: Population per square mile	POPDENS_CY
Foot Traffic	Opting for a busy area with high pedestrian traffic to attract potential customers	● S09_bus: Retail Trade Businesses ● S26_bus: Movie/Amusement Businesses ● S29_bus: Education/Library Businesses	(S09_BUS + S26_BUS + S29_BUS)/sqm
Competition	Assessing the level of competition in the vicinity and choosing a location with a balanced market	● S12_bus: Food store businesses ● S16_bus: Eating & drinking businesses ● N13_bus: Food & beverage stores ● N35_bus: Food services business ● S01_bus: Total businesses	S12_BUS + S16_BUS + N13_BUS +N35_BUS/S01_BUS
Opportunity	A favorable circumstance for progress and success, arising from changes, trends, or market demand	● X1130_a: 2022 Food Away from Home ● X1130fy_a:2027 Food Away from Home ● X1130FY_a:2027 Food Away from Home ● X1002fy_a:2027 Food	(X1130_A / X1002_A) * (X1130FY_A / X1002FY_A)
Affordability	Being close to office complexes can generate regular business from nearby customers	● Hai_cy: Housing affordability index	HAI_CY
Crime	Prioritizing the safety of customers and staff by selecting a location in a secure neighborhood.	● Crmcytotc: Crime index	CRMCYTOTC
Growth Potential	Assessing the growth potential of the surrounding area in terms of	● Totpop_cy: Population	TOTPOP_CY/TOTPOP_FY

	population, development plans, and economic indicators	- Totpop_fy: Expected population in five years	

Each of these values were created via feature extraction using ESRI variables. Each variable is normalized based on other block-group values and centered around 0. Then a linear optimization function is made as below:

$$a * Demo + b * PopDen + c * FootTraf - d * Comp + e * Opp + f * Aff - g * Crime + h * GrowthPot$$

Where the coefficients a,b,c, etc are easily changeable to best fit the type of business. Then, using a python package, PuLP, the optimization function is solved. Each block group's value is calculated and the distance from the optimal score is found. This distance is the block group's opportunity score. The reason PuLP was implemented is to allow future development on this score. PuLP is used for mathematical optimization modeling and provides a high-level interface for creating and solving linear programming problems. It easily allows for different constraints and objectives to be added on in the future. This final opportunity score is then shown on the map as a recommender for where a business should be started.

Since Nourish also aims to promote healthy food-related businesses, a method for suggesting the healthiest alternatives for various types of foods was necessary. Similar to the opportunity score, each USDA food product was assigned a nutrition score. However, determining what foods are considered healthy is often subjective. To address this, a method that focuses on identifying hyperpalatable foods was chosen. Hyperpalatable foods typically consist of processed foods or sweets with enticing combinations of fat, sugar, carbohydrates, and sodium. By indicating whether a food item is hyperpalatable, it can be strongly suggested whether the food is healthy or not.

To determine if a food is hyperpalatable, a technique described in the journal "Hyper‑Palatable Foods: Development of a Quantitative Definition and Application to the US Food System Database" was applied. This journal presents a set of formulas that establish criteria for categorizing foods into three different clusters of hyperpalatable foods: (1) FSOD - Fat and Sodium (> 25% kcal from fat and ≥ 0.30% sodium by weight), (2) FS - Fat and Simple Sugars (> 20% kcal from fat and > 20% kcal from sugar), and (3) CSOD - Carbohydrates and Sodium (> 40% kcal from carbohydrates and ≥ 0.20% sodium by weight). These equations were applied to the USDA Food Data Central database, which encompasses over 1.5 million food products available in grocery stores throughout the United States. This database was selected for its comprehensive nutritional composition information for each food item. To apply the equations, the nutritional values of fat, sugar, carbohydrates, and sodium were extracted and converted to the appropriate units for calculation. Subsequently, a table was created indicating the true/false value for the inclusion of each cluster for every food item.

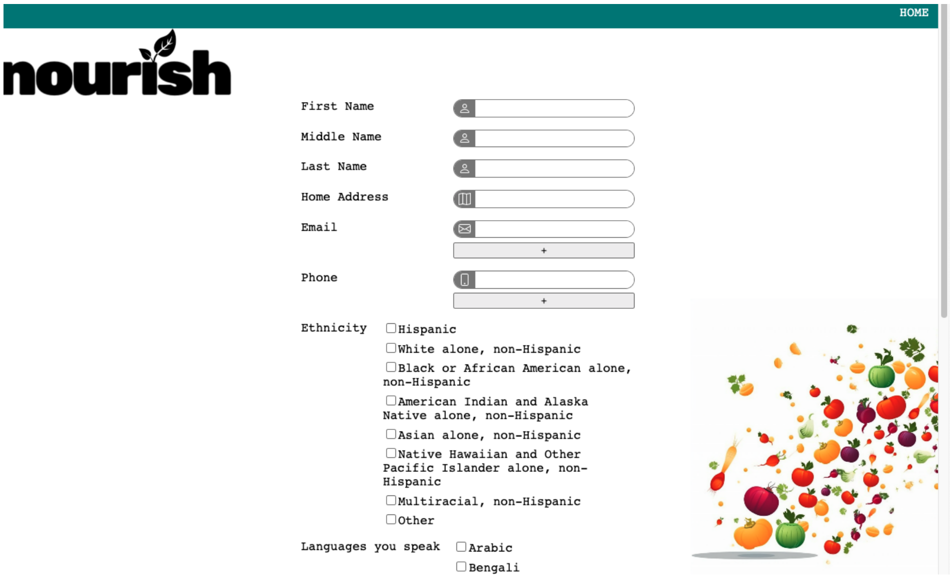
Category	%kcal from fat	%kcal from carbohydrates	%kcal from simple sugars	%sodium by weight
FSOD: Fat and Sodium	>25%	-	-	≥0.30%
FS: Fat and Simple Sugars	>20%	-	>20%	-
CSOD: Carbohydrates and Sodium	-	>40%	-	≥0.20%

The clustering boundaries will not change as they were defined beforehand, thus, no validation of the clusters calculations were performed. However, it was observed that there are instances where nutritional information in the Food Data Central database was incorrect and impossible for certain food products. For example, a record claimed that a turkey had over 500 grams of protein, per 100 grams of weight, which is impossible. Realizing this, a Food Data Central representative was contacted. The representative confirmed that the data was incorrect. Thus, we advise that for the continuation of this project, data validation should be added to ensure that the nutritional composition of foods are correct before applying the equation.

With the knowledge graph, new scores, multiple feature layers, and documents, Nourish's domain knowledge encompasses the data needed to give meaningful insights.

Contextual Knowledge

For the recommendation system to function effectively, it must be capable of extracting context from simple requests and delivering relevant insights to the specific user. For instance, when a user asks, "What types of loans does my business qualify for?", Nourish needs to possess the appropriate context to provide an accurate response. To establish robust and consistent contextual knowledge within the system, a front end was developed, requiring users to sign in or sign up to access the chatbot. By implementing this login feature, the LLM model consistently possesses background knowledge about the user's identity and intentions.

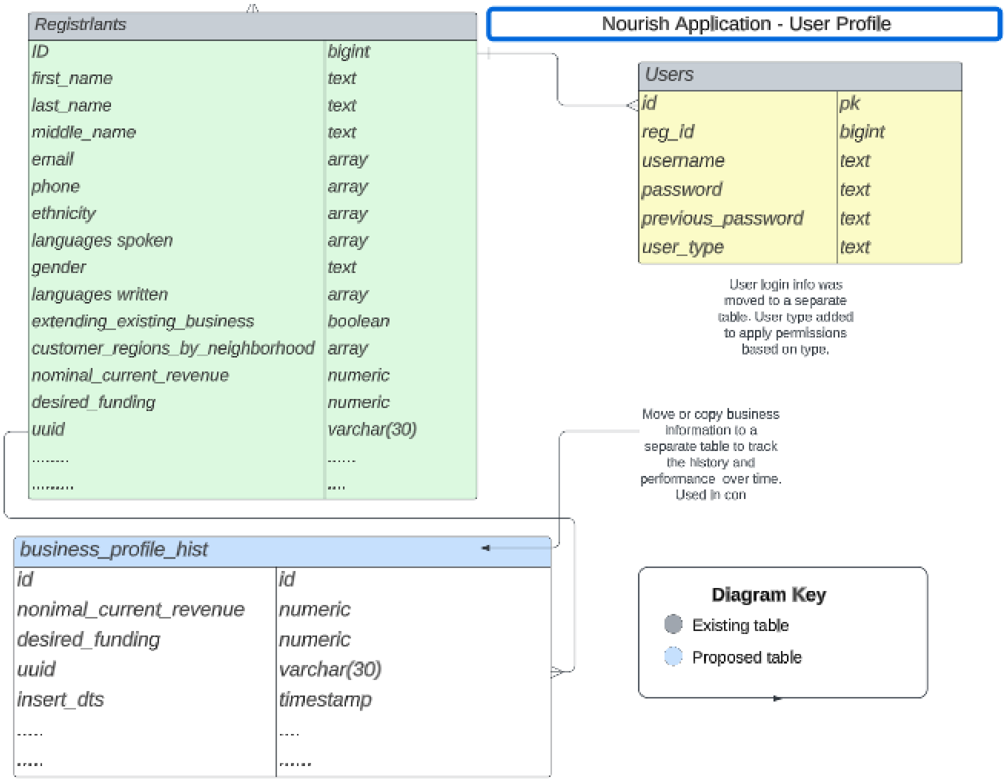


First Name
Middle Name
Last Name
Home Address
Email
Phone

Ethnicity
☐Hispanic
☐White alone, non-Hispanic
☐Black or African American alone, non-Hispanic
☐American Indian and Alaska Native alone, non-Hispanic
☐Asian alone, non-Hispanic
☐Native Hawaiian and Other Pacific Islander alone, non-Hispanic
☐Multiracial, non-Hispanic
☐Other

Languages you speak
☐Arabic
☐Bengali

Upon registration the user is asked various questions ranging from type of business, target audience, geographic location to funding needs. A total of thirty two variables are captured about the user and stored in a Postgres database. This table can and will be accessed by the recommendation system to provide user specific answers.



Registrlants	
ID	bigint
first_name	text
last_name	text
middle_name	text
email	array
phone	array
ethnicity	array
languages spoken	array
gender	text
languages written	array
extending_existing_business	boolean
customer_regions_by_neighborhood	array
nominal_current_revenue	numeric
desired_funding	numeric
uuid	varchar(30)
....	......
....	....

Users	
id	pk
reg_id	bigint
username	text
password	text
previous_password	text
user_type	text

User login info was moved to a separate table. User type added to apply permissions based on type.

Move or copy business information to a separate table to track the history and performance over time. Used in con

business_profile_hist	
id	id
nonimal_current_revenue	numeric
desired_funding	numeric
uuid	varchar(30)
insert_dts	timestamp
.....	....
.....	......

Language Knowledge - Utilizing LLMs

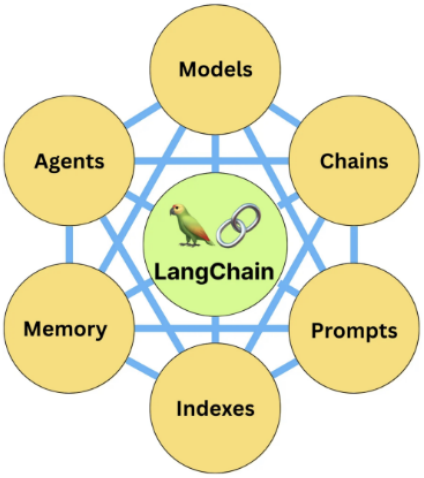


Image from "How to Build a Smart Chatbot in 10 Mins with LangChain"

The Conversational AI Agent is powered by OpenAI's advanced GPT-3.5 language model, which incorporates the LangChain module as an essential component. LangChain serves multiple important purposes within the architecture. Firstly, it enables the creation of a dynamic conversational agent that can respond to questions and inquiries in a flexible and adaptable manner. Unlike traditional chatbots that follow a pre-determined flow, LangChain allows our AI agent to generate responses based on the specific context and needs of each interaction. This ensures a more engaging and interactive conversational experience for users, as the agent can tailor its responses dynamically to address the unique queries posed by individuals.

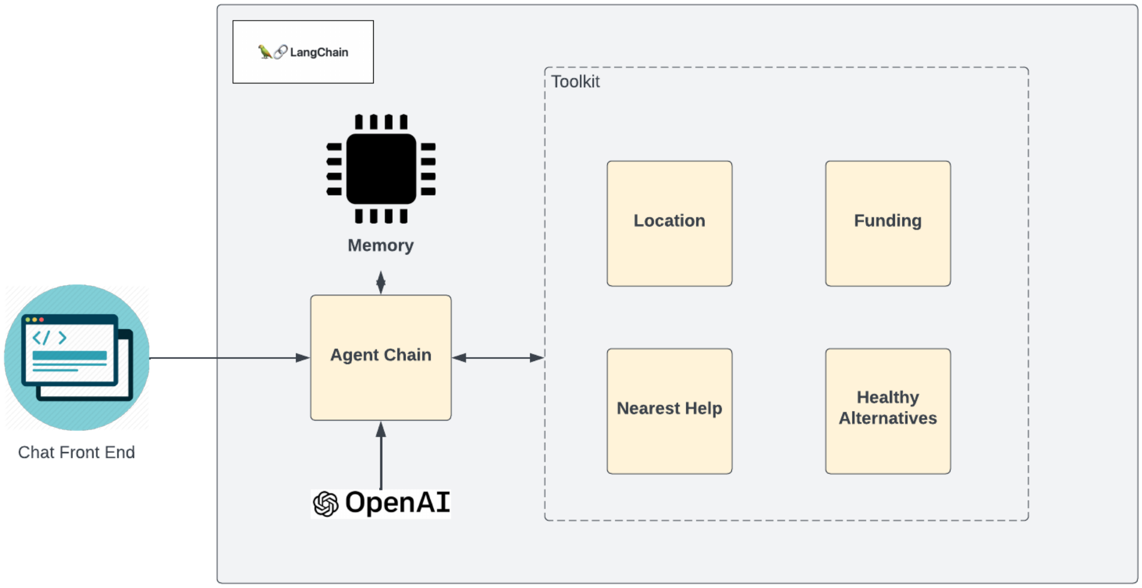
In addition to its conversational capabilities, LangChain also provides a valuable memory feature. This feature allows our AI agent to retain and update the entire conversation as it unfolds during a chat session. By maintaining a contextual memory of the ongoing dialogue, the agent can provide more coherent and coherent responses, taking into account the previous exchanges and ensuring a smoother conversation flow. This memory feature greatly enhances the agent's ability to maintain continuity and context throughout the conversation, leading to a more natural and human-like interaction. It allows the agent to remember important details and references made earlier in the conversation, avoiding repetitive or redundant responses and creating a more personalized experience for users.

Furthermore, LangChain empowers Nourish to leverage its capabilities to develop custom tools and toolkits. These tools enable the application to incorporate domain-specific data or other external data sources beyond what the language model itself provides. This flexibility allows Nourish to enhance the AI agent's knowledge and understanding in specific areas, making it more proficient in handling domain-specific queries or accessing relevant information from external sources. By leveraging LangChain's tool creation abilities, Nourish can augment the AI agent's capabilities and provide users with more accurate, comprehensive, and specialized responses to their queries.

Analysis Methods

Now that the knowledge base is established, Nourish requires the ability to interact with the user. Dash, Flask, and a web based front end was used to deliver a simple interface. The user is able to directly interact with Nourish and the LLM agent by submitting their queries on the interface, triggering a callback. On the callback, the response is given to the agent which forwards the request with prompt engineering to GPT's LLM. The LLM's response is then passed back to the user. If the response is text, the response is displayed. If it's an ArcGis map, the map is saved to a static folder and GPT returns the location of where it's saved. Flask is then used to route the html file to display on Dash. Chat history is maintained locally as a list of dash objects to show chat history.

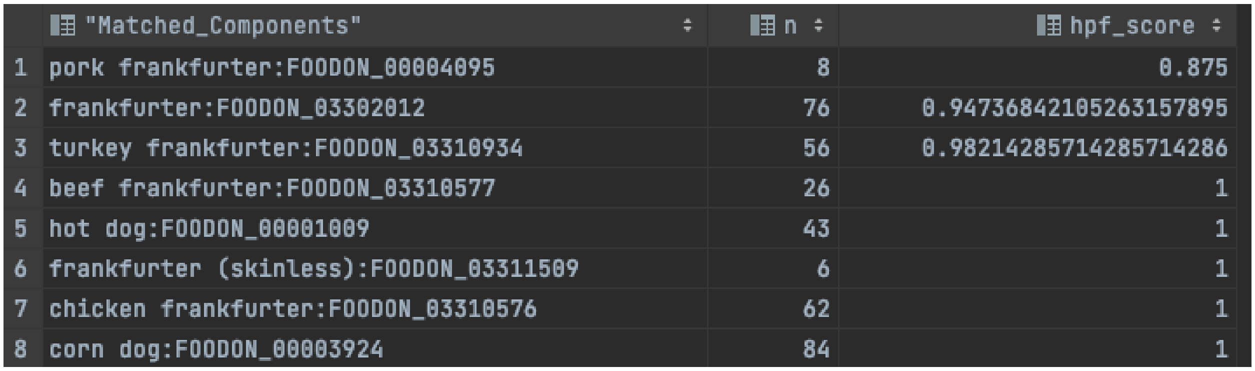
Similar to interacting with the user, Nourish has to be trained to interact correctly with the knowledge base. To achieve this, a toolkit has been developed. Each "tool" in the toolkit can be considered as a recommendation tool. The included tools are funding, locations, healthy alternatives, and nearest help. Each section below includes a sample snippet demonstrating how the conversational AI responds to the user using the respective tool.



Healthier Alternatives Tool

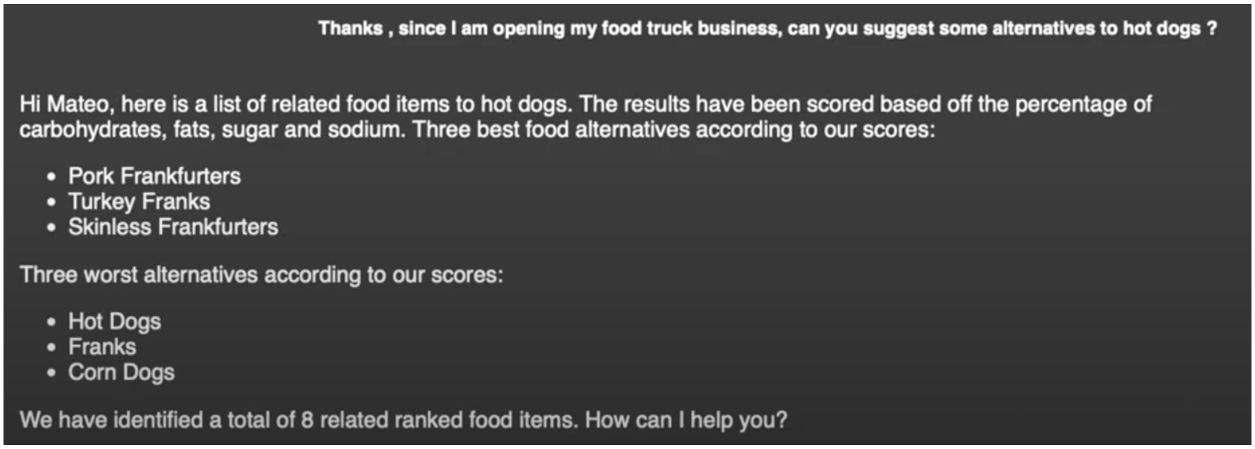
Example user question: "Can you suggest alternatives to hot dogs?"

This tool's primary job is to supply the user with healthier alternatives to various food products. In order to do this, it accesses the knowledge graph to find associated products and their hyperpalatable score. The chat agent recognizes the intent of the question to find healthy alternatives based on the user asking a question such as "Since I am opening a food truck business, what are some alternatives to hot dogs?" The LLM tool parses the food type "hot dogs" and passes that as the input into the neo4j parameterized query that does a full text search on entity labels that have a high string similarity to "hot dog." The results of the full text search are then fed into another cypher query, grouping the results by score. This is crucial to prevent bias in selecting the top of a list when there might be several entities found with the same score. The query then finds a two deep path going either up or down from each entity in a score group and returns the full list of iris for the score found to have children. The stopping criteria for recursive traversal is crucial to provide enough related foods to suggest non trivial alternatives. In this case it would find frankfurter, turkey dog, etc. Once an entity list has been identified, the entity list is returned to python which then passes the list as a parameter to a SQL query. The list is directly passed in python to help prevent hallucinations, costs, and time. Python then passes that list into another parameterized SQL query to map the FoodOn entities to branded food items on exact matches. Once the USDA ID has been identified, each entity can be linked to HPF scores based on their branded food observations. The final output of all these queries can be seen below.



	"Matched_Components"	n	hpf_score
1	pork frankfurter:FOODON_00004095	8	0.875
2	frankfurter:FOODON_03302012	76	0.94736842105263157895
3	turkey frankfurter:FOODON_03310934	56	0.98214285714285714286
4	beef frankfurter:FOODON_03310577	26	1
5	hot dog:FOODON_00001009	43	1
6	frankfurter (skinless):FOODON_03311509	6	1
7	chicken frankfurter:FOODON_03310576	62	1
8	corn dog:FOODON_00003924	84	1

This table is then returned to python and given to GPT-3.5 to format the final answer. The top and bottom three scored items are returned. Once the data is returned, the tool then returns the answer in a way that is concise and meaningful to the user.



Thanks , since I am opening my food truck business, can you suggest some alternatives to hot dogs ?

Hi Mateo, here is a list of related food items to hot dogs. The results have been scored based off the percentage of carbohydrates, fats, sugar and sodium. Three best food alternatives according to our scores:

- Pork Frankfurters
- Turkey Franks
- Skinless Frankfurters

Three worst alternatives according to our scores:

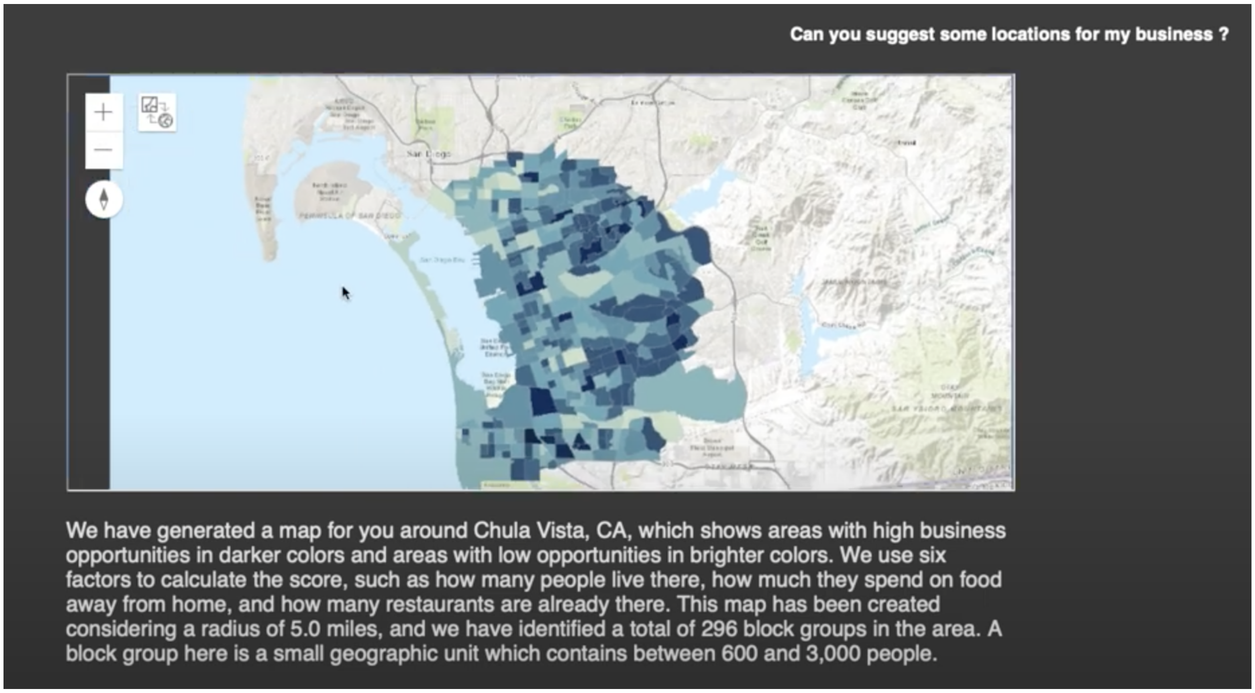
- Hot Dogs
- Franks
- Corn Dogs

We have identified a total of 8 related ranked food items. How can I help you?

Tool: Location

Example user question: “Can you suggest some locations for my business?”

The Location Recommendation Tool is designed to assist users in finding suitable locations. It begins by identifying the user's address/prospective location that was saved in a postgres table when the user signs up for Nourish. It proceeds by generating a 5-mile radius buffer, unless the user specifies a different radius during the chat conversation. This buffer acts as a filter to identify census block groups that intersect with it, and these groups are then enriched with an opportunity score. This score reflects the potential advantages present in each group. By utilizing this approach, the Location Recommendation Tool offers personalized recommendations by considering user-provided location information, applying the buffer as a filter, and incorporating the opportunity scores of the intersecting census block groups.



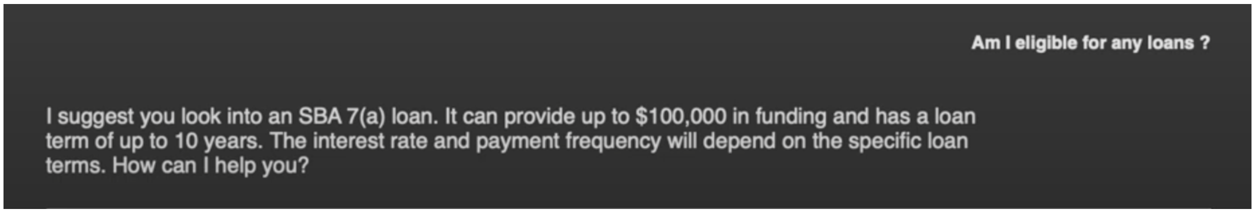
Can you suggest some locations for my business ?

We have generated a map for you around Chula Vista, CA, which shows areas with high business opportunities in darker colors and areas with low opportunities in brighter colors. We use six factors to calculate the score, such as how many people live there, how much they spend on food away from home, and how many restaurants are already there. This map has been created considering a radius of 5.0 miles, and we have identified a total of 296 block groups in the area. A block group here is a small geographic unit which contains between 600 and 3,000 people.

Tool: Funding

Example user question: “Am I eligible for loans?”

The Funding Document Tool is built by leveraging the Llama index module to create indexes of the available SBA and USDA funding documents. Documents were stored on a Google Drive folder and an API was used to access the directory with all the files. LlamaIndex was then used to create embeddings on the document store by chunking the documents. Llama Index then queries these embeddings by taking the user input and running a string similarity comparison to help find relevant documents. Once the document chunks have been selected, the information is passed to GPT to reason over the response. Llama index provides an invaluable service by dynamically reducing the amount of textually context needed for GPT to provide a relevant answer. The alternative would have been to give all the text information straight to GPT along with the user business profile and funding needs. While GPT would be able to identify the answer, this architecture does not scale since it would only be able to process a few documents before exhausting the tokens available and thus ending the conversation. Giving GPT a large amount of unnecessary text also increases the chances of hallucination and runtime. Llama index helps avoid the cost and runtime issues by directly filtering and reducing the context. The primary objective of this tool is to provide concise summaries of loan eligibility criteria and any additional information the user might want about the loan suggested. By utilizing the Llama index and its capabilities, the Funding Document Tool streamlines the process of accessing relevant information about loan eligibility and provides comprehensive details about different types of loans.



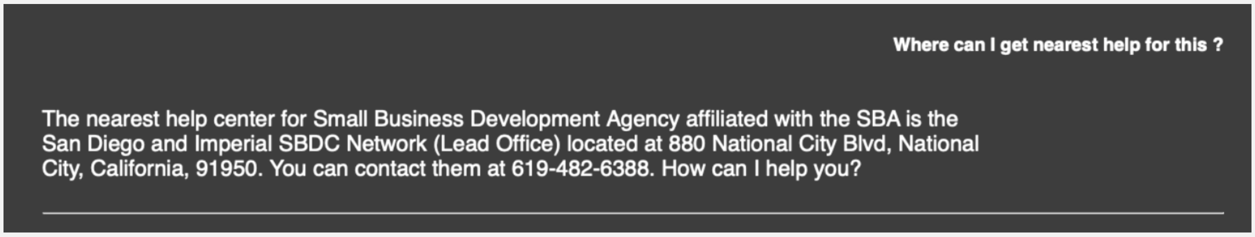
Am I eligible for any loans ?

I suggest you look into an SBA 7(a) loan. It can provide up to $100,000 in funding and has a loan term of up to 10 years. The interest rate and payment frequency will depend on the specific loan terms. How can I help you?

Tool: Nearest Help

Example user question: "Where can I get nearest help for this?"

The Nearest Help Tool is specifically designed to assist users in locating the closest Small Business Development agency office to access the necessary support and assistance. This tool employs the nearest facility algorithm provided by ArcGIS Online, enabling the calculation of distances from the user's home address to all the potential SBA locations available within the GIS system. By evaluating these distances, the tool suggests the center with the least travel distance to the user, ensuring they can easily access the nearest SBA office for their needs.



Where can I get nearest help for this ?

The nearest help center for Small Business Development Agency affiliated with the SBA is the San Diego and Imperial SBDC Network (Lead Office) located at 880 National City Blvd, National City, California, 91950. You can contact them at 619-482-6388. How can I help you?

Creating the Toolkit

To accompany all the tools, Nourish had to correctly detect a user's intent in a prompt, and then choose the correct tool. Using the tool, Nourish had to then return the answer in a way that's concise and relevant. Much of the work revolved around prompt engineering and tweaking the verbage and information passed to GPT to control how friendly the response was as well as the level of information given. In order to judge how the agent was processing user queries, logs were examined for how much drift occurred. The initial testing found that the ReAct Conversational Agent worked better than the Zeroshot agent. The ReAct conversational agent reduced tool drifting and irrelevant answers as well as runtime. For instance, the Zeroshot agent chain struggled to call the right tool to locate the nearest Small Business Administration (SBA) and often selected the funding tool instead. The observed wondering in this case was caused by a fine distinction in extracting more information about an SBA loan versus asking whom to talk to for help filling out the loan. Once the agent chain was transitioned to a conversation agent chain, the drifting dramatically decreased and more accurate answers were given. However, the agent became far more impersonal in the conversation and often referred to the user in the third person. To fix this, it was discovered that one of the parameters in LangChain Tool configuration, called ReturnDirect, was set to True. This meant the agent was not really following the prompts and instead returning the output without rephrasing the answers. Once changed to False, this fixed the problem.

Findings and Reporting

Commercial Product Viability

An application that is targeted towards helping individuals / organizations working in the food industry is a viable concept for a commercial product. Through focus groups, NSF mentorship and interviews with professionals, it was ascertained that there is indeed a need for a tool to help stakeholders navigate the complexities of working in the food industry. For individuals / organizations that want to focus on the business of healthy food, this point is even more true, as making considerations for food healthiness is another complicated layer in an already complicated process.

The developed solution represents an early POC testing phase of a future commercial product. To gauge its effectiveness in a commercial setting, the key metric to consider is the response of end users. Given the extensive nature of this project, it is imperative to leverage user testing and valuable feedback from the target audience. Through active engagement with users, the product developers can identify any gaps or areas where information may be lacking. This user-driven approach enables a business to effectively address user concerns.

By actively fine-tuning results and seeking data sources that align with user concerns, a business solution can enhance the overall robustness of the product. This iterative process allows the business to continually improve and refine the system, ensuring its relevance and reliability for users. Incorporating user feedback and constantly adapting to their needs enables the product to evolve and meet the evolving demands of its users, fostering a strong user-business relationship.

Building The Knowledge Base

A knowledge graph can be successfully constructed from a wide variety of data sources and combined to create a cohesive source of information that is usable in a commercial setting. It can be accompanied by other data sources to create the knowledge base used by an LLM model.

Ontologies

1. An Ontology such as FoodOn can be converted into a Neo4j graph database. Given an ontology parser (such as OwlReady2), one can iterate through every node and deconstruct ontology relationships into the traditional (start, end, relationship) graph triple by introducing “dummy” conjunctions such as “AND”, “OR”, “BLANK” and “NOT” to act a place holders for complicated ontological relationships.

USDA Food Data Central Database:

1. It is possible to tie entities from this database to the FOODON ontology using NLP and lexicographical mappings (LexMapr)
2. It is possible to create meaningful clusters of hyper palatability on this dataset by using the percentage by calories of Carbohydrates, Sodium, Sugars and Fats
3. In general, this database offers a wealth of knowledge for commercial food items and could be sufficient by itself as a data source for a commercial product. The two main aspects of this database, nutrients and ingredients, are ripe for data augmentations such as clustering and NLP respectively. We highly recommend the continued integration of this database for the future of the product and any related products.

Geographic Data

1. Many different feature layers in Arcgis can be Geo-Enriched to create custom feature layers that are designed for specific use cases. It is often rare that an Arcgis feature layer will contain all the required data for an application, so the ability to combine different attributes from feature layers into specific feature layers is very important. Additionally, if a feature layer does not exist yet, one can be constructed using pandas and theArcgis API..
2. It’s possible to create a high level overview to compare block-groups based on different demographic and business variables to help suggest potential locations to start a business.

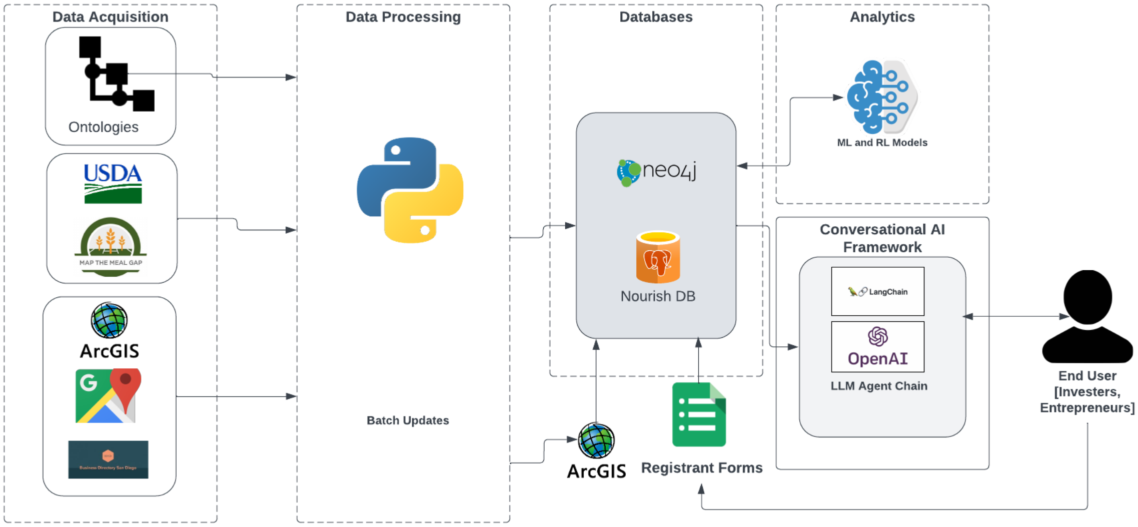
Document Based Data

1. By leveraging the Llama Index framework, it is possible to successfully incorporate extensive documents into a knowledge graph in the form of vector stores/indexes. These vector stores played a crucial role in enabling document querying based on user questions. To accomplish this, LLM models were utilized to parse the questions and leveraged the querying capabilities of the Llama Index on the stored documents. This approach facilitated effective retrieval of relevant information by matching user queries with the indexed documents, empowering our system to provide accurate and contextually appropriate responses.

Large Language Model Integration

Nourish has shown that it is possible to integrate different data structures such as ontologies, relational databases, document based data and geographic information in such a way that can efficiently support conversational answers given by a Large Language Model. Nourish shows that it is possible to take a human inputted question, transform it into a spatial, cypher or relational query, and then answer back in a human readable form; demonstrating that aspects of the data model can be taught to the AI as a way so that it can generate ad hoc queries. In addition, we have shown that creating a strong enough knowledge base reduces “hallucinations” given by Large Language Models and helps return meaningful answers.

Solution Architecture, Performance and Evaluation



Performance

The robustness of the knowledge graph was assessed through rigorous testing of the recommender systems’s response validity. Evaluating language models like LLMs poses a challenge as their usage primarily involves complex text generation, making it difficult to rely on specific metrics such as cluster purity or classification performance. Consequently, the GPT model undergoes trial and error evaluations to gauge the quality of its generated output across various user queries.

Significant evaluation efforts were dedicated to prompt engineering, involving adjustments to the phrasing and information provided to GPT. These modifications aimed to control the response tone and the level of information conveyed. To assess how well the agent processed user queries, logs were examined for any drift that occurred. For example, in the

Zeroshot agent chain, there were difficulties in correctly identifying the tool for locating the nearest Small Business Administration (SBA) office, often opting for the funding tool instead. This drifting was attributed to a fine distinction between extracting more information about an SBA loan versus seeking guidance on filling out the loan application. Transitioning the agent chain to a conversation-based approach significantly reduced drift and improved answer accuracy. However, this change also resulted in a more impersonal conversation style, with the agent frequently referring to the user in the third person. Therefore, striking the right balance between drift, friendliness, and accuracy has been an ongoing endeavor.

Another challenge on top of drifting was also combatting hallucinations. LLMs are well known for their ability to confidently provide answers not grounded in reality. One such example is provided by the healthy alternative food tool. The tool would properly select and run the neo4j and sql queries, but would toss out the answer at the thought process in favor of its own knowledge. While the hallucinated answers were related to the specified food (such as vegan alternatives), the answers were vague, shallow, and not based on observations that GPT had access to. Another example of hallucinations was seen with the funding tool; the funding tool would hallucinate various details about the business profile and funding needs of the individual. This was fixed through prompt engineering by providing explicit instructions to the LLM to not disregard information found and given to it on the returning responses.

Evaluation

The robustness of the knowledge graph continues to be assessed through rigorous testing of the recommender systems's response validity. Evaluating language models like LLMs poses a challenge as their usage primarily involves complex text generation, making it difficult to rely on specific metrics such as cluster purity or classification performance. Consequently, the GPT model undergoes trial and error evaluations to gauge the quality of its generated output across various user queries. Furthermore it can be really hard to evaluate LangChain chains and agents. There are two main reasons for this:

1) Lack of data
There is generally little to no data to evaluate chains/agents over before starting a project. This is usually because Large Language Models (the core of most chains/agents) are terrific few-shot and zero shot learners, meaning the users are almost always able to get started on a particular task (text-to-SQL, question answering, etc) without a large dataset of examples. This is in stark contrast to traditional machine learning where you had to first collect a bunch of datapoints before even getting started using a model.

2) Lack of metrics
Most chains/agents are performing tasks for which there are not very good metrics to evaluate performance. For example, one of the most common use cases is generating text of some form.

Evaluating generated text is much more complicated than evaluating a classification prediction, or a numeric prediction.

Future State

In the future endeavors of Nourish, it is important to explore additional agent types, such as self-ask-with-search and few-shot response, to enhance the quality of responses from the conversational agent during interactions. This exploration will contribute to more meaningful and contextually appropriate conversations with users.

Expanding the context length is another crucial objective to achieve. By allowing the chat agent to engage in longer conversations with users, Nourish can facilitate more comprehensive and in-depth exchanges, enabling a richer user experience.

Currently, the chat agent may experience hallucinations due to potential memory loss or the presence of numerous tools to sift through. Incorporating agents capable of running distinct chains based on user needs and integrating intent identification should help solve this problem. This approach will ensure more accurate and focused responses, minimizing hallucination, and improving the overall performance of the chat agent.

In the future, to better enhance the user experience, a feature should be added to support specific formats of response representation by leveraging LangChain Output parsers. This will enable Nourish to tailor the presentation of responses to meet specific requirements, resulting in a more user-friendly and intuitive interaction with our system.

Conclusion

All in all, the Food Knowledge Graph Analysis accomplished the goal of creating and integrating a complex knowledge graph with a language model to provide a user with relevant information on where they might open a successful business, how they might acquire funding, and healthier alternatives to the foods that they might sell. While this project is still in development, several large obstacles were overcome. The integration between ontologies and nutrition was successful as well as integrating geospatial information to help provide more relevant information to the end user. Complex loan documents were also parsed and summarized in a conversational syntax. While there is more work to be done on ensuring the robustness of the tools, conversation flow, and preventing hallucinations; the chatbot application successfully integrated the knowledge graph with a LLM and provided tangible and actionable insights to the end user.

Citations

Fazzino TL, Rohde K, Sullivan DK. Hyper-Palatable Foods: Development of a Quantitative Definition and Application to the US Food System Database. Obesity (Silver Spring). 2019 Nov;27(11):1761-1768. doi: 10.1002/oby.22639. PMID: 31689013.

Xu, Alex. "How to Build a Smart Chatbot in 10 Mins with Langchain." *How to Build a Smart Chatbot in 10 Mins with LangChain*, 6 June 2023, blog.bytebytego.com/p/how-to-build-a-smart-chatbot-in-10.