

Nutritional Data Integrity in Complex Language Model Applications: Harnessing the WikiFCD Knowledge Graph for AI Self-Verification Across Multilingual International Food Composition Tables to Enrich Accuracy within Software Systems and AI-Enabled Interfaces

Katherine Thornton¹, Kenneth Seals-Nutt² and Mika Matsuzaki³

¹WikiFCD Collaborative, Olympia, WA, USA

²WikiFCD Collaborative, New York, New York, USA

³Johns Hopkins Bloomberg School of Public Health, 615 N Wolfe St, Baltimore, MD 21205, United States

Abstract

Estimation of nutritional intake, a key determinant of our health, requires reliable and accurate food and nutrient information. As the convenience of chat-style interactions appeals to many people, can we trust agents powered by large language models (LLMs) to answer questions about nutrition accurately? We introduce the Wikidata and WikiFCD AI Food Composition Chat Bot (ChatWikiFCD), a chat bot for food composition information powered by structured data from WikiFCD and Wikidata and enhanced by LLMs. This approach combines referenced statements from human-curated knowledge bases, which include mappings to FoodOn, with generative artificial intelligence (AI). The system includes a chat-based application that provides explainable responses linked back to published sources. The system supports multilingual input and will respond in the human language in which a question is posed. This system leverages the benefits of LLMs while also reducing the risk of hallucination and provides fine-tuned data for the food domain sourced from published food composition tables.

Keywords

Food Composition, Nutri-informatics, Wikibase, Wikidata, artificial intelligence, model augmentation, chat automation, hallucination detection, knowledge graphs, linked data

Introduction

Virtual assistants or chat bots are already used to inquire food-related information such as recipe recommendations [1] and cooking instructions [2]. As researchers interested in food data, we wonder if asking questions of systems powered by generative artificial intelligence (AI) related to food composition will yield correct responses? We introduce the prototype Wikidata and WikiFCD AI Food Composition Chat Bot (ChatWikiFCD), a chat bot for food composition information powered by structured data from Wikidata and WikiFCD. This chat bot is a work-in-progress. We combine the strengths of large language models for generating natural language with the human-curated structured data referenced to published sources of information drawn from knowledge graphs to self-verify claims that are generated by the language models. We provide an overview of the system design and include a sample of the questions we used to test the system performance. A diagram of the system is shown in Figure 1.

1. Related Work

People have applied LLMs to systems in the food domain for several years. Researchers evaluated the accuracy of responses from ChatGPT in the domain of nutritional recommendations related to

Proceedings of the Joint Ontology Workshops (JOWO) - Episode X: The Tukker Zomer of Ontology, and satellite events co-located with the 14th International Conference on Formal Ontology in Information Systems (FOIS 2024), July 15-19, 2024, Enschede, The Netherlands.

✉ katherine.thornton@yale.edu (K. Thornton); kenneth@seals-nutt.com (K. Seals-Nutt); mmatsuz2@jhu.edu (M. Matsuzaki)

ORCID 0000-0002-4499-0451 (K. Thornton); 0000-0002-5926-9245 (K. Seals-Nutt); 0000-0002-7020-3757 (M. Matsuzaki)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

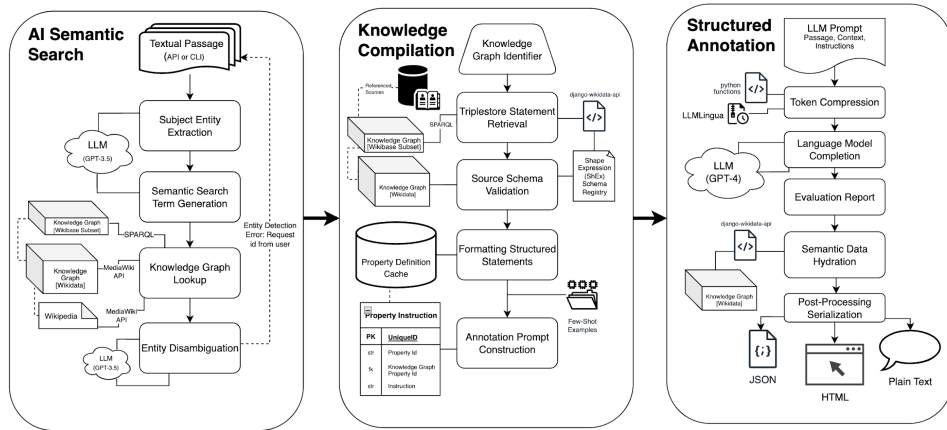


Figure 1: A diagram of the ChatWikiFCD system.

non-communicable diseases and found that responses to complex questions were of lower accuracy than responses to simple questions, and concluded that ChatGPT could not replace the expertise of a health professional [3]. Qi et al. tested food-recommendation chat bots backed by LLMs and identified explainability and personalization as strengths of the evaluated chat bots [4].

Researchers have demonstrated the utility of applying artificial intelligence to the domain of nutrition in [5]. Researchers have demonstrated the current limitations of ChatGPT regarding the domain of medical advice, which we are aware also applies to nutrition-related questions in a chat setting [6, 7, 8]. Another risk of using LLMs in application development is that they are known to provide plausible but incorrect responses, sometimes termed “hallucination” [9, 10, 11]. People have successfully used fact-checking techniques to mitigate the risk of LLM hallucination. Some experts have used text from Wikipedia to fact-check LLMs [12]. Researchers find knowledge graphs to be useful sources of information for fact-checking LLMs [13]. Researchers have also found that using data from Wikidata specifically improves the accuracy of responses for Llama, Alpaca, and GPT-3 [14].

Sequeda et al. demonstrated that for question answering tasks, a system using a LLM in combination with a knowledge graph representation of a SQL database returned answers that were 37.5% more accurate than a system using an LLM without a knowledge graph [15]. Addressing the challenge of reducing LLM hallucination, adding external sources of knowledge such as facts from relational databases, to LLM workflows, Peng et al. demonstrated improved accuracy of responses [16].

After experimenting with LLM question answering, researchers have observed that LLMs may return facts that seem plausible but are inaccurate, which they have named hallucination [9, 10?]. Others have created techniques for detecting LLM hallucination [?]. Rather than detecting hallucination, we hope to prevent it in order to minimize the risk of responding to food composition questions with inaccurate information. Combining LLMs with knowledge graphs has been successful for fact-checking LLMs [13]. Xu et al. successfully used data from Wikidata to improve the performance of Llama, Alpaca, and GPT-3 [14].

Researchers have raised concerns about the lack of explainability of responses from LLMs [17]. Increasing the explainability of AI-driven systems is an important ethical consideration for system designers [18]. The fact that our system surfaces sources for the facts it returns in responses provides explainability for responses the system communicates.

2. Wikidata and WikiFCD

Editors have been adding data to the Wikidata knowledge base since the project launched in 2012 [19]. People recognize Wikidata as one of the most prominent public knowledge bases available [20]. We map data to, and reuse data from, the Wikidata knowledge base in the WikiFCD knowledge base. We

also reuse data from Wikidata in the Wikidata and WikiFCD AI Food Composition Chat Bot.

While Wikidata contains a substantial subgraph related to foods, dishes, and cuisines, as of 2024 there is not much food composition data in Wikidata. Food composition data is made up of nutrients and their values. WikiFCD is a knowledge base of food composition data sourced from published food composition tables [21]. WikiFCD contains mappings to the Wikidata knowledge base as well as mappings to identifiers from FoodOn, the Farm to Fork ontology [22]. WikiFCD makes use of Wikibase¹, the extension of MediaWiki² used to enable Wikidata. We created WikiFCD as an independent knowledge base in order to make detailed food composition data readily available for reuse and querying [23].

Many of the sources we consulted to find food composition data are national-level food composition tables (FCTs) such as the *SMILING Food composition table for Indonesia* published in 2013 or the *ASEAN Food Composition Database* published in 2014. We created individual statements for each nutrient and value for each food item in each food composition table. In Figure 2, we see the first few statements containing nutrients and their values for the food item ‘Medlar, African, raw’ from the Malawi 2019 FCT. Each statement includes a reference back to the source publication in which it was published. In this way, people who reuse data from WikiFCD can identify the provenance of data, whether they reuse a single nutrient value, or tens of thousands of values.

The screenshot shows the Wikidata page for the item 'Medlar, African, raw' (Q567956). The page is in English and includes a navigation bar at the top with options like 'English', 'Kat', 'Talk', 'Preferences', 'Watchlist', 'Contributions', and 'Log out'. Below the navigation bar, there is a search bar and a 'Search wikifcd' button. The main content area displays the item's label 'Medlar, African, raw' and its description 'Food Item from Malawi 2019'. A table below shows the item's properties in English, including the label 'Medlar, African, raw' and the description 'Food Item from Malawi 2019'. The 'Statements' section is expanded, showing two statements: 'instance of' (food item) and 'carbohydrate' (20.7 gram(s) per 100 gram(s)). Both statements have a reference to the 'Malawian Food Composition Table 2019'.

Figure 2: Statements on the food item ‘Medlar, African, raw’ from the Malawi 2019 FCT with references.

3. ChatWikiFCD

We wanted to create a chat-based application for food composition data that would provide responses based on data from WikiFCD and Wikidata. We created an interface for people who would like to interact with ChatWikiFCD by a web form. We take the input from this form and use that to perform semantic search by combining AI with search engine software. We used multiple packages and services to build this system including LangChain , OpenAI , python , django-wikidata-api , wikidataintegrator and SPARQL.

¹<https://wikiba.se/>

²<https://www.mediawiki.org/wiki/MediaWiki>

3.1. Subject Entity Extraction

Once people input their question into the system, we use OpenAI's GPT-3.5³ to locate entities from the natural language text. After GPT-3.5 identifies entities, we ask it to generate keywords related to each entity in the form of tokens. We define token as a piece of text that the model will process, and we ask GPT-3.5 to do this task with as few tokens as possible. In order to increase the breadth of the search, we ask for aliases and possible variants of the entity's name. We use the aliases, variants, and keywords to be able to find a candidate in any way it may be described in our system, and to support disambiguation in case multiple candidate entities are stored in the Knowledge Graph Lookup step of this process. Throughout the conversation, we regularly perform this step to dynamically extract entities as the model detects new subjects being mentioned by the person.

```
{
  "name": "stawberry",
  "aliases": ["strawberries", "garden strawberry", "fragaria"],
  "keywords": ["fruit", "nutrients", "berry", "food item"]
}
```

Figure 3: Aliases and keywords for the food item 'strawberry'.

3.2. Knowledge Graph Lookup

To find items in WikiFCD that correspond to subject entities, we use the python package wikidataintegrator (WDI). The WDI package supports search via the WikiFCD SPARQL endpoint as well as the MediaWiki API interface for WikiFCD. Using either of these two search methods allows us to find the WikiFCD Q-identifiers (Qids) for the relevant entities in WikiFCD.

3.3. Entity Disambiguation using LLMs

Many foods have similar names, for example 'tuna' is the name of a fish⁴ as well as the name of the fruit of *Opuntia cacti*⁵. To address this challenge, we developed an LLM prompt chain designed to support disambiguation between food and nutrient-related entities. In this context, we define a prompt as the natural language string we use as input instructions for an LLM to perform a task which results in a structured response [24]. We use `django-wikidata-api` library to retrieve structured data for each relevant Qid in WikiFCD. As part of the prompt we ask GPT-3.5 to determine the closest match to the entity and to state a rationale for the selection.

```
[
  "Q116463", "Q198635", "Q257236", "Q271424",
  "Q424284", "Q379426", "Q458201", "Q73603",
  "Q5233", "Q518506", "Q405406", "Q360017",
  "Q373467", "Q372032", "Q372009", "Q397425"
]
```

Figure 4: Multiple candidate Qids found related to 'strawberry'.

³<https://platform.openai.com/docs/models/gpt-3-5-turbo>

⁴<https://en.wikipedia.org/wiki/Tuna>

⁵<https://en.wikipedia.org/wiki/Opuntia>

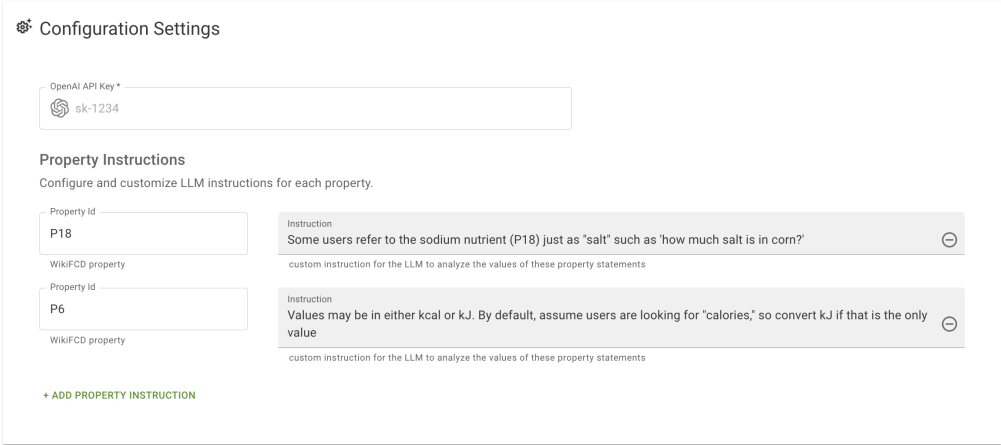
4. Source Knowledge Compilation

Working from the Qid of the entity in question, we gather information from the structured data of the knowledge base to verify the LLM response.

4.1. Annotation Prompt Construction

The next step of the workflow is to annotate the text of the response to the question. We provide instructions to the LLM to generate a response to the original question within a specific framework that limits hallucination. We task the LLM to identify claims made in its response and assess their validity. Due to the increased level of difficulty of this task, we use GPT-4 for this step as it is a more capable, higher-performing model [25]. We structure our annotation prompts to include seven components: 1. Instructions to establish the focused domain of nutrients and food composition data, 2. Task explanation of how to parse the statement structurally and use the provided contextual data as the only source of truth, 3. Few-shot condensed examples to improve the consistency of response formats, 4. The reduced JSON object containing structured data related to the entities relevant to the conversation, 5. Property-specific guidance instructions (discussed below), 6. Safe-guarding and verification guidance on what types of questions it should not attempt to respond to, and 7. Question text along with prior conversation history within a dynamic context window.

Each property in WikiFCD contains structured metadata statements that provide additional details about the meaning and usage of the property. For properties that are equivalent to Wikidata properties, we provide mappings to Wikidata. We use these statements on the properties and fine-tuned sub-prompts to create a dynamic cache of property instructions. People can apply additional rules to be considered in the property cache with natural language. An example of human-readable property instruction text is available in Figure 5. We use this cache to instruct the LLM on how to interpret the statements related to the entity and to refine validation rules within the prompt. Using the cache allows us to fine tune validation rules without modifying the prompt structure itself. We task the LLM to return a JSON-encoded array for each response with information about the start index and stop index per claim, and each property identifier used in a statement that the LLM used to determine claim validity.



The screenshot shows a 'Configuration Settings' interface. At the top, there is a field for 'OpenAI API Key *' with the value 'sk-1234'. Below this is the 'Property Instructions' section, which includes a sub-header 'Configure and customize LLM instructions for each property.' There are two rows of configuration. The first row shows a 'Property Id' of 'P18' and a 'WikiFCD property' field. To the right, an 'Instruction' box contains the text: 'Some users refer to the sodium nutrient (P18) just as "salt" such as "how much salt is in corn?"' with a small circular icon to its right. Below the instruction is a smaller line of text: 'custom instruction for the LLM to analyze the values of these property statements'. The second row shows a 'Property Id' of 'P6' and a 'WikiFCD property' field. Its 'Instruction' box contains: 'Values may be in either kcal or kJ. By default, assume users are looking for "calories," so convert kJ if that is the only value' with a similar icon. Below this is another line of text: 'custom instruction for the LLM to analyze the values of these property statements'. At the bottom left of the configuration area, there is a green button labeled '+ ADD PROPERTY INSTRUCTION'.

Figure 5: Property instructions for the LLM within the ChatWikiFCD system.

5. Structured Annotation

Once the annotation prompt is complete, we use it to initiate the annotation step of our workflow to produce verification statements for claims and organize evidence from WikiFCD.

5.1. Prompt Token Compression

Prompt compression is a technique that researchers developed to enhance performance [26]. We use prompt compression in this workflow in order to reduce the size of the prompts we transmit to and receive from the LLM. In order to make the system more efficient, we use this strategy to accelerate inference time and reduce operational costs associated with each request to the LLMs' completion API endpoints. We leverage a number of python utility functions that we developed to trim and condense characters from our prompt construction step. For static prompt templates, we also use LLMs to detect areas where instructions can be refined and avoid redundancy. We then pass our compiled prompts to LLMlingua⁶ which we use to compress the prompt into as few tokens as possible while preserving the original meaning and intent.

5.2. Language Model Completion

We use the OpenAI GPT-4 completion endpoint for the next stage of the workflow [25]. We then use LangChain to parse the text responses we get back from GPT-4 because it can make use of Pydantic's Base Model Class⁷. Thus, we can define data structures using Python's native typing system to construct programmable interfaces using object-oriented programming techniques. We request that the LLM return the response as well as information about the properties from the knowledge base used in each annotation. This information enables us to link claims to supporting statements from the knowledge base, and reduces the risk of LLM hallucinating responses drawn from training data alone. Using the WikiFCD Qid of each food entity, we create a SPARQL query to gather relevant statements about that food item from the knowledge base. We can then consult this set of structured data when attempting to verify claims about the entity.

5.3. Evaluation Report

As requested in the LLM prompt instructions, the model generates a response passage and includes citations in a series of JSON-encoded dictionaries. In Figure 6 we see that for every claim, the model stores a start index, stop index, the identifier of which entity the claim is being made about, and a set of property identifiers of which statement values were used to make that claim. We use this information to generate metrics. We tabulate the number of claims, the characters in each claim, and the percentage of the passage each claim represents. We combine these metrics with our set of human-annotated examples for comparison. Changes between the system scores and the scores from the human-annotated examples allow us to track system performance over time. This information is the feedback we use to identify opportunities to refine prompt construction techniques or expand property instruction information.

```
[
  {"start_index": 0, "stop_index": 6, "entity": "Q568262", "properties": ["P10"]},
  {"start_index": 31, "stop_index": 47, "entity": "Q568262", "properties": ["P17"]},
  {"start_index": 55, "stop_index": 79, "entity": "Q570124", "properties": ["P19", "P20"]},
  {"start_index": 101, "stop_index": 132, "entity": "Q567947", "properties": ["P90"]},
  {"start_index": 180, "stop_index": 183, "entity": "Q568262", "properties": ["P13"]},
  {"start_index": 187, "stop_index": 194, "entity": "Q567947", "properties": ["P19", "P20"]},
  {"start_index": 223, "stop_index": 272, "entity": "Q567947", "properties": ["P10"]},
  . . .
]
```

Figure 6: A code snippet with an example of an annotation response.

⁶<https://www.llmlingua.com/>

⁷https://docs.pydantic.dev/dev/api/base_model/

5.4. Data Hydration

Data hydration is defined as the process of incorporating data into a computational object. After the system completes the generation of the evaluation report, we recombine the structured data from WikiFCD and Wikidata that we removed during the annotation and compression steps. This data hydration step allows us to use all contextual data related to the the food items in question. The additional contextual data makes the responses from our system more informative. Developers who build applications that reuse data from our system will be able to use this contextual data. We provide an example response file in our Github repository for the user interface system⁸. We use the contextual data in the ChatWikiFCD application we introduce below.

```
{
  "id": "Q567947",
  "label": "Guavas",
  "description": "Food Item from Malawi 2019",
  "aliases": [
    "guava", "Gwafa", "Psidium guajava"
  ],
  "composition table": {
    "id": "Q567639",
    "label": "Malawian Food Composition Table 2019",
    "description": "first edition of the Malawian Food Composition Table",
    "url": "https://dl.tufts.edu/concern/pdfs/d217r336d",
    "wikibase_url": "https://wikifcd.wikibase.cloud/wiki/Item:Q567639",
    "country_flag_image": "https://upload.wikimedia.org/wikipedia/commons/0/08/Flag_of_Malawi.svg"
  },
  "image": "https://upload.wikimedia.org/wikipedia/commons/thumb/0/02/Guava_ID.jpg/800px",
  "wikibase_url": "https://wikifcd.wikibase.cloud/entity/Q567947",
  "wikidata_id": "Q3181909",
  "wikidata_url": "https://www.wikidata.org/entity/Q3181909"
}
```

Figure 7: Structured data related to the food item ‘guava’.

6. Demonstration Food Composition Chat Bot

We created a chat-based application that provides self-fact-checking of responses posed to an LLM related to food composition information. This is an alpha version of the chat bot, and we consider it a work-in-progress. ChatWikiFCD is an interactive conversational application that leverages structured data from two knowledge graphs, WikiFCD⁹ and Wikidata¹⁰. The ChatWikiFCD source code is available on Github¹¹.

6.1. Technologies Used in ChatWikiFCD

We designed this prototype of the ChatWikiFCD application following a single-page approach. The rendering engine is React¹². We used Vite¹³ as the web framework, Axios¹⁴ as the API client integration, and Material-UI¹⁵ as the design system.

6.2. Configuration Options in ChatWikiFCD

We provide a form where people interested in using ChatWikiFCD can provide their own OpenAI key. We also offer a form to input a WikiFCD identifier if they have a specific food item in mind about which they would like to chat about.

⁸<https://github.com/ScienceStories/wikifcd/blob/322eeabe/src/tests/fixtures/chat-sample-response-complex.json>

⁹https://wikifcd.wikibase.cloud/wiki/Main_Page

¹⁰https://www.wikidata.org/wiki/Wikidata:Main_Page

¹¹<https://github.com/ScienceStories/wikifcd>

¹²<https://react.dev/>

¹³<https://vitejs.dev/>

¹⁴https://axios-http.com/docs/api_intro

¹⁵<https://mui.com/material-ui/>

6.3. Question Selection

The ChatWikiFCD interface invites people to ask their own question about food items and nutrition information using a chat-style form, as seen in Figure 8. We provide a set of example prompts and questions that demonstrate the capabilities of our system. We invite people to pose questions to ChatWikiFCD using natural language.

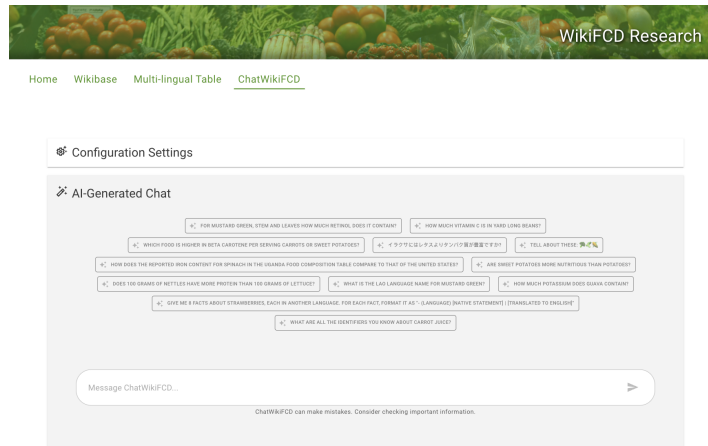


Figure 8: An introductory screen invites people to ask ChatWikiDP a question about food composition information.

6.4. Visualizing Generative Responses

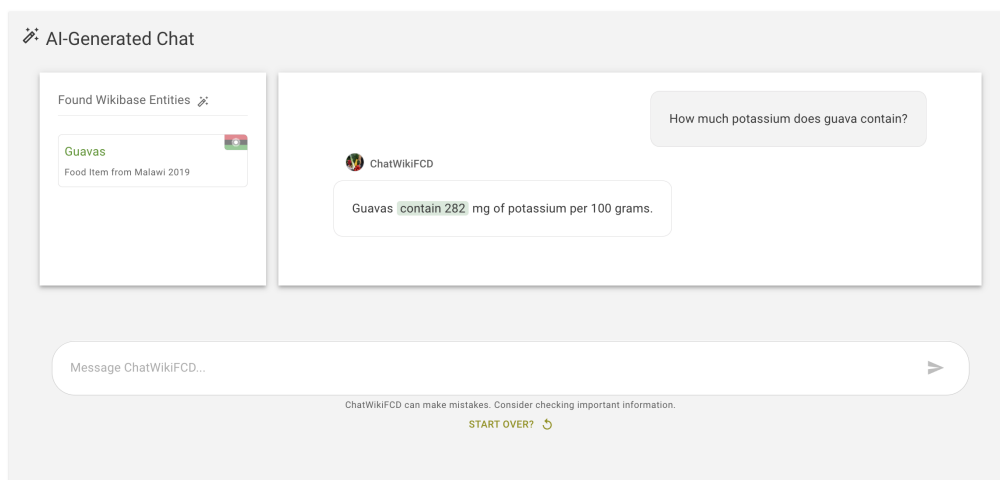


Figure 9: AI-generated response with annotation

People asking questions of the system gain confidence that it has correctly identified the subject of their question when they review a rich information card populated with data from WikiFCD and Wikidata along with an image from Wikimedia Commons, as seen in Figure 10.

When an entity is detected for which we can provide data from a specific food composition table, we extend the interactive card with an image of the country flag in the top right corner of the card as a quick indicator of the source country. Similarly with the entity card as a whole, upon hovering over the flag indicator, a menu is revealed. This menu includes the name of the FCT, short description, a link to the original source, and a deep link into the WikiFCD entity for the FCT. Figure 11 demonstrates the interaction of a food item from the ‘Malawian Food Composition Table 2019’ dataset.

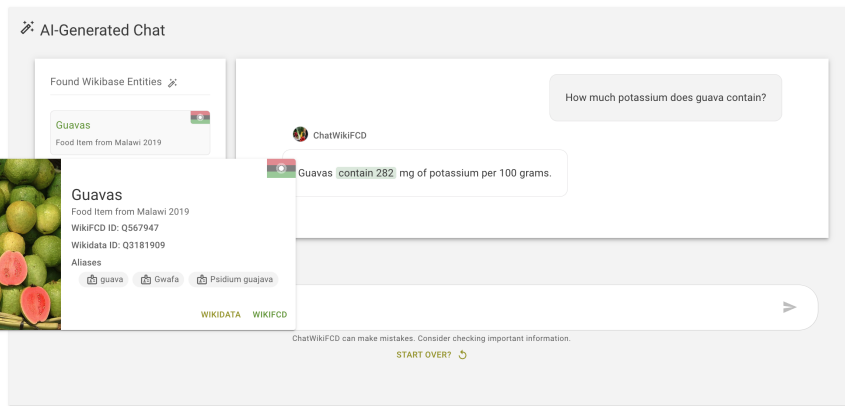


Figure 10: A rich card showing information about ‘Guava’.

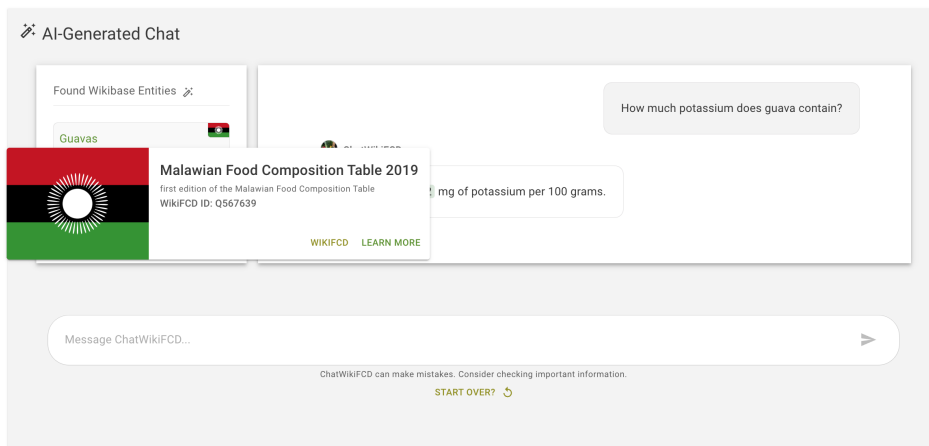


Figure 11: A rich card showing information about the source Food Composition Table

We provide visual indications of how the text response compares to facts drawn from WikiFCD in the interface. Figure 9 shows that generated responses contain highlighted sections that indicate valid claims matching facts from WikiFCD. Hovering over a particular annotation reveals a ‘Verified in WikiFCD’ menu that consists of deep links to source materials as well as references for the claims from the knowledge base, as seen in Figure 12.

6.5. Multi-subject Questions

If people have follow-up questions, a conversation might develop. We support conversations in a variety of human languages. For example, in Figure 13, we share an example of a conversation in Luganda. The English translation of the question is “How does the reported iron content for brown rice in the Uganda Food Composition Table compare to that of the Malawi table?”. The English translation of the response is “Based on the data provided by the two countries’ composition tables, the Malawi table says 3.2mg per 100g, whereas the Uganda table says 1.8mg per 100g”. The sidebar on the left-hand portion of the figure provides the references for these facts with links to the items for these food composition tables in WikiFCD. We use images of the flags from Wikimedia Commons for each country to represent the national food composition table for that country.

If people ask the system questions that contain multiple subjects, the task becomes more complex. The fact-checking system must attribute claims to the entities mentioned accurately. If the matching between entities and claims is inaccurate, then the results will also be inaccurate. For conversations in which multiple entities are mentioned, we present cards for each entity in the sidebar, as seen in

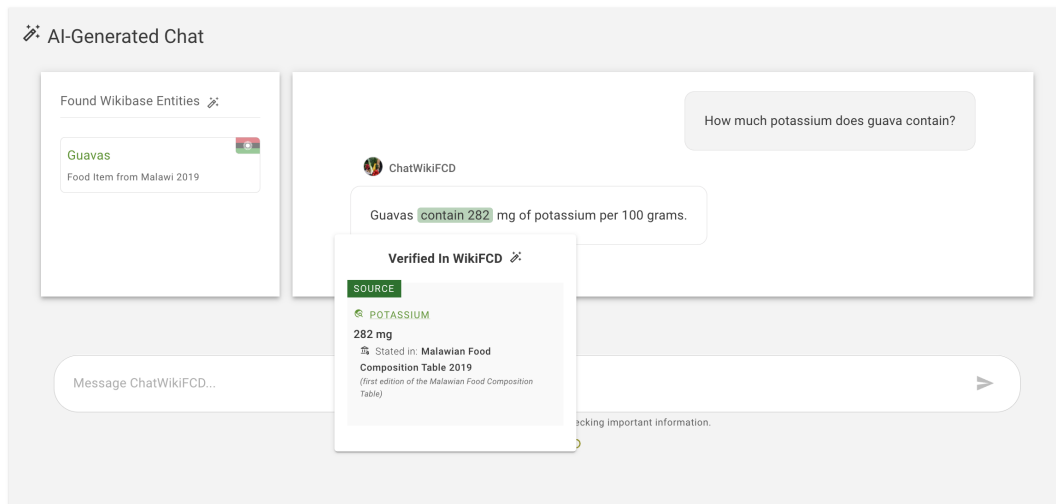


Figure 12: A screenshot of a ChatWikiFCD interaction including questions and responses.



Figure 13: A screenshot of a ChatWikiFCD interaction in Luganda including questions and responses.

Figure 13 where both the Uganda Food Composition Table and the Malawi Food Composition Table are tracked throughout the conversation.

7. Conclusion

Understanding a diet requires nutrition information related to the composition of food items. Food composition data are the building blocks of nutrition information about foods. If application developers seek to offer natural-language interfaces for systems powered by large language models (LLMs), providing the system with structured data from a trusted external source can improve data quality. Leveraging data from a human-curated knowledge base is an effective technique to mitigate the risk of LLM hallucination. We have demonstrated that our approach of using the structured data from the knowledge base ensures that the LLM will return responses that are connected to source publications. When we enrich the prompts with data from our knowledge base, we reduce the risk of a general foundation model solely relying on its own training data for providing responses.

The WikiFCD wikibase provides structured data about many thousands of food items. ChatWikiFCD provides an interactive interface through which people can ask questions of the data in WikiFCD using natural language. We provide interactive interface as an additional pathway for people to engage with food composition data. We hope that the interactivity helps people who found the Wikibase difficult to navigate to access data from WikiFCD. We present deeplinks back to items in the WikiFCD wikibase throughout the ChatWikiFCD interface to facilitate review of the data.

Building applications that are powered by LLMs enriched by data from knowledge bases is a strategy to facilitate transparency and explainability for people using them. Health information is a domain in which inaccurate responses could have harmful implications for people. Reusing data from a knowledge base that includes references, like WikiFCD, allows us to present those references in our ChatWikiFCD application. Supporting the deeplinks to the supporting facts in the WikiFCD knowledge base provides people with a pathway to verify the response ChatWikiFCD provides. This strategy builds on the work people have invested in the scientific analysis of food composition, the work of data curators who have contributed it to structured knowledge bases, while also harnessing the natural language strengths of generative artificial intelligence. Let's infuse data we already trust into our applications powered by AI.

Acknowledgments

We thank the Joint Food Ontology Working Group for productive discussions about FoodOn and data related to food. We thank the Wikidata community for continuing to improve the Wikidata knowledge base. We thank the community of editors of Wikimedia Commons for sharing multimedia resources.

References

- [1] J. Chu, Recipe bot: The application of conversational ai in home cooking assistant, in: 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), IEEE, 2021, pp. 696–700.
- [2] S. Chan, J. Li, B. Yao, A. Mahmood, C.-M. Huang, H. Jimison, E. D. Mynatt, D. Wang, " mango mango, how to let the lettuce dry without a spinner?": Exploring user perceptions of using an llm-based conversational assistant toward cooking partner, arXiv preprint arXiv:2310.05853 (2023).
- [3] V. Ponzo, I. Goitre, E. Favaro, F. D. Merlo, M. V. Mancino, S. Riso, S. Bo, Is chatgpt an effective tool for providing dietary advice?, *Nutrients* 16 (2024). URL: <https://www.mdpi.com/2072-6643/16/4/469>. doi:10.3390/nu16040469.
- [4] Z. Qi, Y. Yu, M. Tu, J. Tan, Y. Huang, Foodgpt: A large language model in food testing domain with incremental pre-training and knowledge graph prompt, 2023. arXiv: 2308.10173.
- [5] T. P. Theodore Armand, K. A. Nfor, J.-I. Kim, H.-C. Kim, Applications of artificial intelligence, machine learning, and deep learning in nutrition: A systematic review, *Nutrients* 16 (2024). URL: <https://www.mdpi.com/2072-6643/16/7/1073>. doi:10.3390/nu16071073.
- [6] A. J. Nastasi, K. R. Courtright, S. D. Halpern, G. E. Weissman, A vignette-based evaluation of chatgpt's ability to provide appropriate and equitable medical advice across care contexts, *Scientific Reports* 13 (2023) 17885.
- [7] H. L. Walker, S. Ghani, C. Kuemmerli, C. A. Nebiker, B. P. Müller, D. A. Raptis, S. M. Staubli, Reliability of medical information provided by chatgpt: assessment against clinical guidelines and patient information quality instrument, *Journal of Medical Internet Research* 25 (2023) e47479.
- [8] L.-L. Liao, L.-C. Chang, I.-J. Lai, Assessing the quality of chatgpt's dietary advice for college students from dietitians' perspectives, *Nutrients* 16 (2024). URL: <https://www.mdpi.com/2072-6643/16/12/1939>. doi:10.3390/nu16121939.
- [9] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, J. Weston, Neural text generation with unlikelihood training, in: International Conference on Learning Representations, 2019.
- [10] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, *ACM Computing Surveys* 55 (2023) 1–38. URL: <https://doi.org/10.1145/3571730>.
- [11] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. arXiv: 2311.05232.
- [12] R. Zhao, X. Li, S. Joty, C. Qin, L. Bing, Verify-and-edit: A knowledge-enhanced chain-of-thought framework, in: Proceedings of the 61st Annual Meeting of the Association for Computational

- Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 5823–5840. URL: <https://aclanthology.org/2023.acl-long.320>. doi:10.18653/v1/2023.acl-long.320.
- [13] S. Feng, V. Balachandran, Y. Bai, Y. Tsvetkov, Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge, 2023. doi:<https://doi.org/10.48550/arXiv.2305.08281>. arXiv:2305.08281.
 - [14] S. Xu, S. Liu, T. Culhane, E. Pertseva, M.-H. Wu, S. Semnani, M. Lam, Fine-tuned LLMs know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over Wikidata, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 5778–5791. URL: <https://aclanthology.org/2023.emnlp-main.353>.
 - [15] J. Sequeda, D. Allemang, B. Jacob, A benchmark to understand the role of knowledge graphs on large language model’s accuracy for question answering on enterprise sql databases, arXiv preprint arXiv:2311.07509 (2023).
 - [16] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, et al., Check your facts and try again: Improving large language models with external knowledge and automated feedback, arXiv preprint arXiv:2302.12813 (2023).
 - [17] J. A. McDermid, Y. Jia, Z. Porter, I. Habli, Artificial intelligence explainability: the technical and ethical dimensions, Philosophical Transactions of the Royal Society A 379 (2021) 20200363.
 - [18] N. Balasubramaniam, M. Kauppinen, A. Rannisto, K. Hiekkänen, S. Kujala, Transparency and explainability of ai systems: From ethical guidelines to requirements, Information and Software Technology 159 (2023) 107197.
 - [19] D. Vrandečić, Wikidata: A new platform for collaborative data collection, in: Proceedings of the 21st International Conference Companion on World Wide Web, ACM, 2012, pp. 1063–1064.
 - [20] A. Hogan, E. Blomqvist, M. Cochez, C. d’Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, Synthesis Lectures on Data, Semantics, and Knowledge 12 (2021) 1–257.
 - [21] K. Thornton, K. Seals-Nutt, M. Matsuzaki, D. Damion, Reuse of the foodon ontology in a knowledge base of food composition data, Semantic Web Journal (2023).
 - [22] D. M. Dooley, E. J. Griffiths, G. S. Gosal, P. L. Buttigieg, R. Hoehndorf, M. C. Lange, L. M. Schriml, F. S. Brinkman, W. W. Hsiao, Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration, npj Science of Food 2 (2018) 1–10.
 - [23] K. Thornton, K. Seals-Nutt, M. Matsuzaki, Introducing wikifcd: Many food composition tables in a single knowledge base, in: CEUR Workshop Proceedings, volume 2969, CEUR-WS, 2021.
 - [24] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, 2023. arXiv:2306.08302.
 - [25] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.
 - [26] H. Jiang, Q. Wu, C.-Y. Lin, Y. Yang, L. Qiu, LlmLingua: Compressing prompts for accelerated inference of large language models, 2023. doi:<https://doi.org/10.48550/arXiv.2310.05736>. arXiv:2310.05736.