RecycLens - Team extra-fluffle

Report

Introduction

RecycLens is an AI recycling assistant that is designed to help users determine how to properly dispose of items based on their local recycling regulations in the state of NY.

Importance of problem

The goal of RecycLens is to reduce recycling contamination and user confusion by providing accurate, county-specific disposal guidance through a conversational, image-based interface grounded in verified local data. Since disposal rules vary from county to county, many people do not know if they should toss an item into recycling, the landfill, hazardous waste, or another drop-off location, and this confusion can reduce the amount of waste that is recycled. According to the EPA, of the 267.8 million tons of municipal solid waste generated by Americans in 2017, only 94.2 million tons were recycled or composted. RecycLens clears this confusion by allowing users to upload an image or description of their item, enter their location, and receive location-specific disposal guidance. By building this application, we hope to encourage better and safer recycling practices. This is a common confusion in people's daily lives, but RecycLens is meant to clear this confusion by allowing users to upload an image or description of their item, enter their location, and receive location-specific disposal guidance. By creating a quick solution to answer these questions, people can dispose things correctly more often instead of navigating through social media or other guidelines that aren't correct for their location. This would also ideally help recycling centers not have to sort as much misplaced recyclables.

Why LLMs are used

We want to address this problem using an LLM where users first can input their location (zipcode, county, etc.) and interact with a chatbot to ask their questions about recycling. Some examples we envision are: - "How and where should I dispose batteries?" (if needed, the LLM should follow up with more questions about specific type of batteries and condition of batteries) - "Is my Chipotle bowl recyclable? Does it go in paper only, cans, or plastic?" The LLM's response should contain detailed answers along with a map of the closest relevant centers near the user's location. We plan to give structured data to the LLM so it can consistently output things like the county's official recycling page URL, contact information, and hours of operation. LLMs are suited for this problem because people face complex everyday scenarios and best describe their situation through natural language, and a normal predictive model can't really give as much catered information as possible to the user in the form of a series of instructions along with supplemental info. They're also useful since we can train them on a very particular knowledge base of recycling sites in New York, so they're able to summarize a lot information available scattered throughout the Internet (which is time-consuming for humans to do) in a concise manner on our web application. Our final project combines visual analysis and LLM reasoning with county-level recycling rules. This way, users can have clarity on instructions, contamination warnings, and a map of the nearest disposal centers. By using Retrieval-Augmented Generation (RAG) with county-verified data, our system strives to reduce misinformation and encourage safer and more sustainable disposal practices.

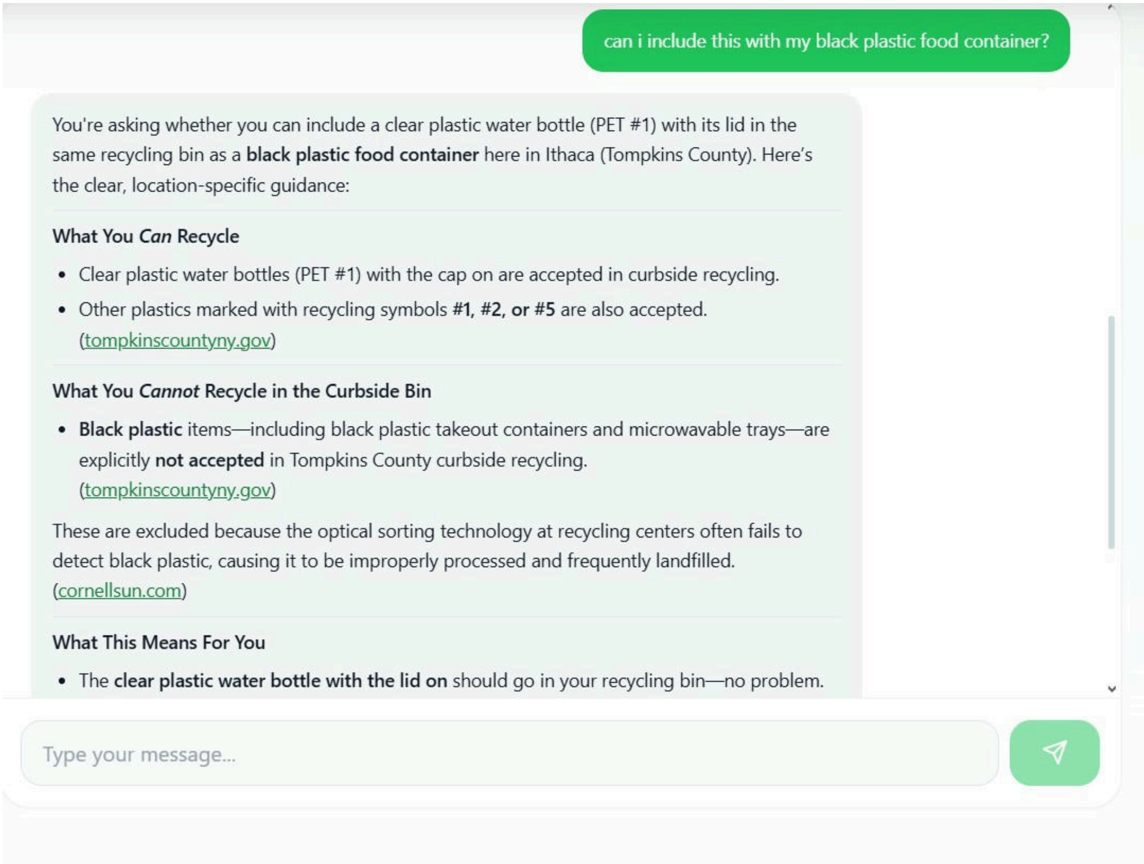
Justification of approach

Our approach uses a hybrid system that combines a RAG knowledge base with web-search capabilities so our chatbot can always give a grounded, accurate answer. The RAG layer provides county-verified recycling rules that we collected ourselves, which helps prevent hallucinations and ensures the model is using real local information. But if a county has missing or unclear data, the web-search fallback lets the model pull updated details and avoid giving an empty or incorrect response. This combination works well, because recycling rules vary a lot between locations, and people usually ask open-ended questions that need both interpretation and reliable facts. Using both RAG and web search allows the system to understand the item, apply the correct local guidelines, and still remain flexible when the knowledge base doesn't cover a specific scenario. We considered rule-based systems and static lookup tools that map items to disposal rules. However, these approaches struggle with ambiguous items, follow-up questions, and mixed materials, and they do not adapt well to county-level variation. Traditional classifiers could identify materials from images but cannot explain disposal steps or incorporate local policy constraints. Our LLM-based hybrid approach allows natural language interaction, clarification through follow-up questions, and grounded reasoning using verified county data, making it better suited to real-world recycling decisions. We plan on using GPT-4o vision for our vision model so that users can upload an image of their item and we can identify that material, condition, potential contamination, and any visible symbols to help them figure out how to properly dispose of the item. Additionally, we plan to use GPT-4o with web search for our LLM reasoning component to help interpret rules, apply local constraints, and generate instructions for the user. Our RAG knowledge base provides base answers for each county's regulations to reduce hallucinations and ensure accuracy. This approach also allows the system to scale from a single county to all 62 counties in New York without redesigning core logic.

Design process

We have built RecycLens in stages, starting from a simple Shiny prototype and adding more features as we went. First, between 11/18 and 11/20, we focused on validating the idea and getting the basic pipeline working. We built a small React + Express MVP that could call the OpenAI API and return a response, and we also tested Mapbox integration to make sure we could eventually show nearby recycling locations. During this phase, we also met with Professor Soltoff to confirm that our idea was feasible and aligned with the project goals. Next, from 11/20 to 11/25, we started building the knowledge layer. We manually collected recycling and solid waste information for all 62 New York counties by visiting each county's official website. We stored the key URLs and details in a structured CSV file and then began turning that into a RAG knowledge base by chunking the text, creating embeddings, and organizing it by county. In loading the website data, we faced a design choice between saving the website as a PDF or HTML input before converting to Markdown. After some testing, we decided to use HTML to ensure less manual work in collecting the data if in the future, more data from other counties will be added to the RAG store. We use the PDF functionality for key pdfs that are found in the sites that are important to include. We did this to have a reliable, county-level data source that the model could use instead of guessing. From 11/26 to 12/04, we integrated the RAG system into our app and tested the full pipeline. We connected retrieval to the model calls so that when a user asks a question, the system first looks up relevant county chunks and feeds them into the LLM. We also added fallback logic: if RAG does not return enough information, the model can use web search, and if that still fails, it falls back to more general rules. During this time, we tested the system with different item types (like food containers, electronics, batteries) and different counties to see where it struggled and adjusted our prompts, chunk sizes, and metadata. We also worked on the user interface. We designed a simple flow where users upload an image, enter their location, and then get back: (1) a clear explanation of how to dispose of the item, (2) any contamination warnings, and (3) a map and cards for nearby facilities. We used these tests to refine the layout, wording, and loading behavior to make the experience feel smoother. Finally, we began switching from the early Shiny prototype to our final stack with React on the frontend and Express on the backend. In this step, we wired the React UI to the full hybrid pipeline (vision analysis to RAG retrieval to LLM reasoning to map results) so that the final product matches the design goals we set at the beginning. One major trade-off we faced was between completeness and reliability. Allowing unrestricted web search improves coverage but increases the risk of inconsistent or unofficial guidance, while relying only on RAG limits coverage when county data is sparse. Our hybrid design balances these concerns by prioritizing verified county data and using web search only when necessary. We also iterated on prompt structure to ensure the model asked clarifying questions rather than guessing when images or user input were ambiguous.

Once our core features worked, we moved to testing whether the LLM's results were accurate. For example, in the screenshot below, we manually verified that black plastics are not allowed in Tompkins County Recycling, so it was able to get this detail correct.



Limitations

All of the following limitations may affect user trust, response latency, or the completeness of guidance in edge cases, which reinforces the importance of transparency, official source links, and conservative recommendations for safety-critical items.

1. County data completeness: Not all counties provide detailed or consistent recycling information, so our RAG results may vary in quality. We work with this by using our hybrid system: if the RAG knowledge base doesn't return enough information, the model uses web search or general guidance so the user still gets an answer.

For example in the screenshot, while we were collecting data for the RAG, we noticed a page that had rules that only applied this year. It's likely that these rules and dates will change in 2026.

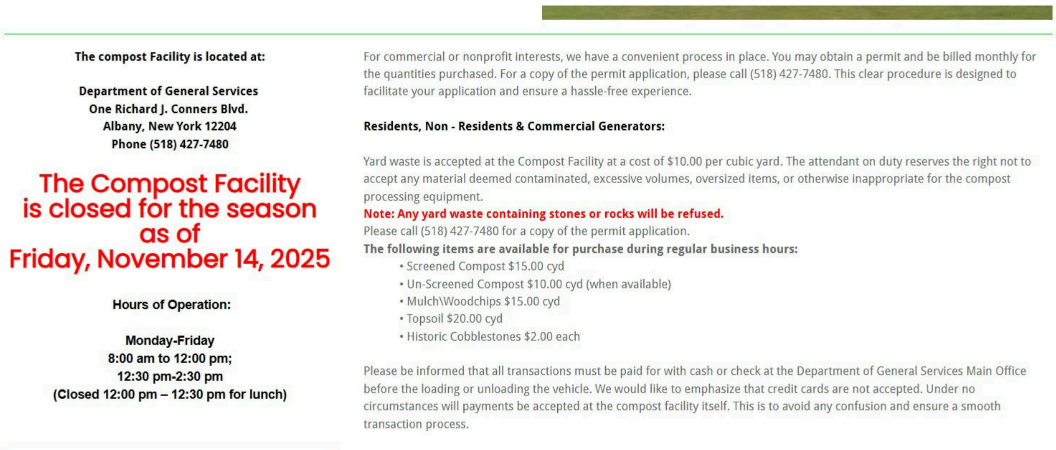


Figure 1: Screenshot of outdated information on a county website

2. Changing recycling rules: Recycling guidelines often change, especially for plastics, electronics, and hazardous waste. We handle this by always linking the official county website in the response and planning periodic updates to the knowledge base so our system stays aligned with current rules. Our web-search component also ensure our data stays up to date.

3. Image uncertainty: User-uploaded images can be unclear, contain multiple items, or be hard for the vision model to interpret. We address this by allowing users to add optional context and by prompting the model to ask clarifying questions when the image is ambiguous.

4. Hazardous waste complexity: Items like batteries, chemicals, and sharps have very strict, location-specific rules, and mistakes can be unsafe. To work with this, the system always flags hazardous items clearly, defaults to safer recommendations, and relies heavily on official county hazardous waste guidelines. This is also why we include a "general recomendation" section for disposing common, everyday items.

5. Latency from multi-step processing: The pipeline includes vision analysis, retrieval, reasoning, and facility searches, which can create slower responses. We manage this by caching county data, optimizing retrieval steps, and limiting web-search calls to situations where RAG information is insufficient. We understand that we might have longer wait times if we want more accurate results, but that is a trade off we are willing to make since we want to prioritize accurate information.

6. New York–only scope: The system currently covers only New York State counties, so users outside this area may not receive fully accurate guidance. We try to design the system to be expandable to other states and use web search to provide broader, general information in the meantime.

Generative AI reflection

At the start, we used Generative AI mainly to help us brainstorm ideas, understand unfamiliar concepts, and draft early versions of our pipeline. As we got further, our use of AI shifted to more targeted questions, like debugging retrieval issues, understanding the documentation of

Llama-Index, or refining how we structured our API calls. This showed us how our understanding improved over time and matched the course idea that AI works best as a support tool rather than something you rely on blindly. In the final product, generative AI is at the center of our workflow. We call GPT-4o for vision analysis and again for reasoning with our RAG knowledge base and web search. Building this made us see both the strengths and weaknesses of LLMs. The model can understand natural language, work with images, and explain recycling rules clearly, but it can also hallucinate or give outdated information, as evidenced with the broken URLs and occasional missing contact information. We also used generative AI to help refine our prompting to be more specific and eliminate as many hallucinations as possible. This connected directly to what we learned in class about grounding, retrieval, and the limits of LLMs. We also had to think about how to use AI responsibly because incorrect recycling advice, especially for hazardous items, can be unsafe. This led us to consider how we could incorporate safety into our design. In creating the RAG knowledge store, we included metadata of the source URL we got information from to enable our plans later to include links to official county websites, add a common warnings page, and use a hybrid approach so the model is backed by real data. Additionally, we added a disclaimer statement right below our outputted advice to warn users to be cautious of fully-trusting AI output. We tried to tie in themes we learned in our INFO 4940 course regarding safety, transparency, and responsible model behavior. Overall, this project helped us understand generative AI more deeply and why it needs structure, real data, and human oversight to be reliable.