Based on our vision for what we want our chatbot to be capable of, we selected our problem domain to be cooking- specifically we wanted to scrape and store recipes from a trusted website (and other websites recommended by this trusted source) to be able to make recommendations to a user regarding recipe ideas given inputted ingredients or desired/undesired cooking techniques and come up with possible suggestions for ingredient substitutions (based on semantic distance in WordNet or some other lexical database). The specific URL selected as the staring location for the scraper was foodwishes.blogspot.com based on the consistent quality of the recipes, the simple nature of the layout of the site allowing for easy scraping and avoiding request timeouts, and the list of sister/recommended sites allowing for escaping the original website domain without compromising on recipe quality.

Our web crawler scrapes from the starting domain and based on previous runs filters out sites containing a list of terms found in websites that are guaranteed to not be relevant- social media links, JavaScript, and other more site-specific keywords. From there, it separates links into two folders per crawling iteration- links on the original domain, and links outside of the original domain. This was done to help ensure that the crawler would escape the original domain. On each crawling iteration, it returns two lists of all links found back to the main function, where each list is then iterated through, scraping each site contained. For better organization and observation of the scraping behavior, new folders are created on each call of the crawl function to allow the user to examine what is being found as the crawler continues to advance.

Because of the nature of cooking websites containing a disproportionate amount of irrelevant/unhelpful information compared to more directly topic/information-oriented websites, a much larger quantity of websites were scraped and the fifteen relevant links had to be selected manually rather than being algorithmically determined and returned by the program itself. Additionally, due to the non-insignificant runtime of the web crawler, the decision was made to separate the crawling script and the file cleaning script.

For the generation of the knowledge base from the scraped websites, the selected websites were placed in a folder and read into the file cleaner script. Each document is tokenized and a count unigram dictionary is generated after filtering out stopwords and a selection of additional terms based on known irrelevant keywords that appeared disproportionately when measuring importance with both term frequency and tf-idf. Due to the large quantity of boilerplate text even on relevant websites, to get quality manually selected terms, the list of algorithmically selected important terms had to be expanded outside the range of 25-40. The 10 best domain-specific words selected from the important terms were: cuisine, cup, food, rice, tablespoons, pinch, making, butter, sugar, and chocolate. While a knowledge base surrounding these words would not be entirely sufficient for the intended purposes of the chatbot, the terms allow information regarding some ingredients, techniques, and general food knowledge. From these ten terms, a dictionary was created that is indexed by a combination of the first relevant keyword found in a given sentence – the fact/information being stored – and a count to both allow the key to be unique and to capture a sense of how much information related to a specific keyword exists in the knowledge base; for mass retrieval of information related to a topic, searching based on the keyword and iterating over the increment will allow for easy retrieval of all information regarding a specific topic. Below is a sample conversation based on the ideal version of the chatbot being imagined.

REMI- Welcome! I’m REMI, the Recipe Exploration and Modification Intelligence! What’s your name?

USER- I’m Cole!

REMI- Hi Cole! How can I help you today- would you like me to recommend a recipe, or do you need help with one you’ve already found?

USER- I’d like you to recommend me a recipe- I have the following ingredients but I am not sure what to do with them: potatoes, beef, cabbage, and rice.

REMI- Alright, let me see if I can find something delicious for you…

REMI- Okay Cole, I found something that matches 3/4 ingredients you gave me: Corned Beef and Cabbage Shepherd’s Pie. Would you like the recipe, or should I try to find something different?

USER- Sounds great, I’d like the recipe!

REMI- Okay, here it is:

[insert recipe here]

USER- Oh, I don’t have any [ingredient], is there anything I could use instead?

REMI- No problem, I can give you some ideas that might work, but you should double check my recommendations just in case.

REMI- [Queries WordNet for similar terms]

Here’s my suggestions for potential replacements.

USER- Great, thanks REMI!

REMI- No problem, enjoy!

Knowledge Base (also attached separately):

Text

Description automatically generated