Natural Language Processing Based Nutritional Application

by

Rachael Naphtal

S.B., Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Author . . . .
Department of Electrical Engineering and Computer Science
May 22, 2015

Certified by. . . . .
James Glass
Senior Research Scientist
Thesis Supervisor

Accepted by . . . .
Albert R. Meyer
Chairman, Masters of Engineering Thesis Committee

Natural Language Processing Based Nutritional Application

by

Rachael Naphtal

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2015, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

The ability to accurately and efficiently track nutritional intake is a powerful tool in combating obesity and other food related diseases. Currently, many methods used for this task are time consuming or easily abandoned; however, a natural language based application that converts spoken text to nutritional information could be a convenient and effective solution. This thesis describes the creation of an application that translates spoken food diaries into nutritional database entries. It explores different methods for solving the problem of converting brands, descriptions and food item names into entries in nutritional databases. Specifically, we constructed a cache of over 4,000 food items, and also created a variety of methods to allow refinement of database mappings. We also explored methods of dealing with ambiguous quantity descriptions and the mapping of spoken quantity values to numerical units. When assessed by 500 users entering their daily meals on Amazon Mechanical Turk, the system was able to map 83.8% of the correctly interpreted spoken food items to relevant nutritional database entries. It was also able to find a logical quantity for 92.2% of the correct food entries. Overall, this system shows a significant step towards the intelligent conversion of spoken food diaries to actual nutritional feedback.

Thesis Supervisor: James Glass
Title: Senior Research Scientist

Acknowledgments

I would first like to thank Jim Glass for all of the guidance and advice he has given me over the past year.

I also would like to express my thanks to Mandy Korpusik. Mandy is responsible for the language understanding part of the system, and was an invaluable resource. She was always helpful and informative, and I am glad I had the chance to work closely with her over the past year and a half.

I would also like to thank Scott Cyphers for all of his help with setting up the system and debugging errors, as well as the editing of my thesis. Thanks to undergraduates Eann Tuan and Mari Kobiashvili for their work this year with the user interface.

Finally, I would like to thank all of my family and friends for their constant support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dietary assessment is a key component in helping people both improve their nutrition and fight obesity. Tracking nutritional intake is a useful and effective tool in monitoring a variety of food based diseases. It is also important in the management of chronic diseases such as diabetes, heart disease, and cancer [1].

According to the National Institutes of Health, obesity is the second largest cause of preventable deaths in the United States [2]. More than 34.9% of adult Americans are obese, and over two thirds are overweight [3]. Obesity is also a significant problem around the world, with an estimated 13% of the world's population being obese according to the World Health Organization [4].

Accurate dietary assessment and monitoring is one of the most effective methods available to help fight obesity, and yet it is often a very difficult and time consuming task for both patients and nutritionists. Currently, nutritionists use three major methods to track individuals' food intake: written records, 24-hour recall records, and food frequency questionnaires.

One of the most common methods used by nutritionists, both when working with individuals and when conducting studies, is the Food Frequency Questionnaire, or FFQ [5]. FFQs are meant to gather information about a person's nutrition over a long period of time, such as a month or a year. They ask for general amounts eaten for around 100 types of food, as well as usual estimated quantities. The difficult part of FFQs is that they ask for a user to generalize tendencies. Often these generaliza-

tions can be inaccurate. FFQs also do not account for the inevitable variability of nutritional intake. For these reasons, the accuracy of FFQs is currently a source of debate [6].

The two other major methods used are 24 hour recall records and recall diaries at the time of eating. Both of these reflect more accurate tracking of nutrition on a daily basis. They are normally administered either by questioning from a trained nutritionist, or by writing diaries on paper, and thus require both literate users, and professionals to interpret the data [7]. Tracking each meal can prove very time consuming for the users, and thus many food diaries are easily abandoned. In fact, when individuals are asked to track each meal, most stop by the third day, or purposely change what they are eating to make it easier to log [5]. Multiple days of data are needed to do any statistical analysis of nutritional patterns, so this abandonment can pose a serious problem.

The system described in this thesis is a prototype of an application that would allow individuals to track daily nutritional intake while addressing many of the problems that patients and nutritionists are currently facing. By mitigating both the fatigue from users and the time needed for nutritionists to obtain useful information, this application could solve a major problem [8]. Specifically, our application uses spoken language as the method for gathering information, thus lowering the literacy barrier and also providing a more natural and efficient way for users to interact with the system.

One of the biggest barriers for other approaches that used spoken food records has been the time required from professionals to interpret and codify both the food items and quantities eaten. In this thesis, we describe methods used to perform this analysis and generate relevant nutritional database entries without the need for professionals. This approach could significantly lower the cost of maintaining detailed tracking, all while retaining the efficient form of logging through natural language.

In Chapter 2, we discuss the importance of our system in relation to existing nutritional applications, and outline the nutritional databases used. In Chapter 3, we discuss the main approaches used to generate nutritional matches based on food

items. Chapter 4 discusses how we dealt with logically interpreting quantities from natural language. The abilities of the system to handle refinement of food choices is discussed in Chapter 5.

Chapter 2

Background

2.1 Existing Nutritional Applications

Mobile phone and web based applications show significant promise in terms of their accessibility and lower rate of abandonment than traditional pen and paper [9]. There are a variety of applications that currently exist for tracking nutritional intake, with two of the most popular being myFitnessPal [10] and Calorie Counter [11].

myFitnessPal contains a variety of features such as a calorie counter and exercise tracker. To enter items into myFitnessPal, users can type search terms and see a list of many related food items which they can choose from. The food items include a variety of restaurant and brand based entries. The application then allows users to select quantities from a set list and add the selected food to a meal. Our approach aims to incorporate a variety of the capabilities of myFitnessPal in terms of logging food, while making the entire process more natural and more efficient. Specifically, myFitnessPal requires users to search and log each food individually, which can be time consuming and involves typing on a smart phone, a less natural form of interaction than speaking.

2.2 Available Nutritional Databases

One of the major considerations in building our application was which databases to use to acquire the most relevant and reliable nutritional information. We focused our

analysis on four major databases, and ended up using a hybrid of two of them in particular. The four databases that were explored for use in the system as well as their advantages and disadvantages are discussed below.

2.2.1 USDA National Nutrient Database for Standard Reference

The USDA National Nutrient Database for Standard Reference (SR) is the major source of food composition and nutritional data in the United States [12]. It is the database that is primarily used by other nutritional databases and applications in both the public and private sector. The data is compiled from a variety of sources including scientific literature, private members of the food industry, and government agencies, and is updated about once a year.

The current system primarily uses the USDA Database for Standard Reference for its source of nutritional information. The nutritional information we used is from release 27, which is current as of August, 2014. It contains nutritional information on 8,618 food items and values for up to 150 food components each.

There were a variety of advantages to using the SR as the major backbone of our system. In addition to being one of the most reputed sources of information, the SR was also uniformly organized in its layout and description of food items. Its consistent method of naming food entries allowed for refinement to much more accurate results than other databases allowed. Unfortunately, the SR focuses mainly on major generic types of food and does not include much information on complete meals or specific brands of food. For that reason, we explored a variety of other databases to help reconcile the food items that the SR does not contain.

2.2.2 USDA Food and Nutrient Database for Dietary Studies

Another major database that was explored for use in the system was the Food and Nutrient Database for Dietary Studies, or FNDDS which is also maintained by the USDA[13]. The data in the FNDDS is generated based on food items collected using

a part of the National Health and Nutrition Examination Survey known as "What we eat in America." Thus, entries in this database correspond to foods that Americans regularly eat. It differs from the SR in that it contains many more meals eaten by Americans, but does not contain all of the food composition data that the SR does. The current version of FNDDS contains information on 7,618 foods. The formatting of food items is less consistent, and the number of entries is smaller than the SR, so we decided to focus on the USDA SR in the current version of the system. However, the FNDDS could be a powerful future addition to the system.

2.2.3   Nutritionix

While both of the databases offered by the USDA contain information on many types of food items, neither contains information on food found in restaurants or packaged food by particular brands. For example, neither contains entries for Cliff Bars or specific menu items from popular chains. Thus, for many items not contained in the USDA Standard Reference, we explored another nutritional database called Nutritionix. Nutritionix provides an open API that allows users to query over 320,000 unique food items and collect their nutritional information [14]. In addition to items from the SR, Nutritionix also contains entries for restaurant and packaged foods. Their data is entered mainly by restaurants and retailers themselves, and they have over 130,000 restaurant food items available.

Nutritionix offers a unique amount of nutritional data and support for food items; however, its major disadvantage lies in the limited ability to query and search for specific food results. Unlike the USDA database which can be downloaded and run locally, using Nutritionix requires REST API calls to their servers, and the options for refinement are limited. Additionally, much of the functionality detailed in the API does not work reliably, and irrelevant results were occasionally returned. The items are also not consistently named, and it was difficult to discern if results were actually relevant. For example, "almond milk" may be returned when the food item queried was "almonds," and this mistake could be very frustrating to users. For this reason, Nutritionix is used in combination with the SR in the current system, but

these issues were addressed in its usage.

2.2.4   Semantics3

While Nutritionix contains nutritional information for the majority of queries, its API makes it hard to find specific brands of store bought items. For example, the search "Milano" returns options for different types of bread, but no entry for Milano cookies. To address this missing area, we explored semantics3, an ecommerce API [15]. Semantics3 specializes in products available online and that have Universal Product (UPC) codes. It has information on up to date prices and also details such as the nutritional labels of all food items sold.

The data in the semantics3 database is gathered from a variety of reputable online retailers and is updated regularly. This is a major advantage over Nutritionix, which relies on user or company submissions to update information. It also results in a much wider array of packaged food items. The major disadvantage of using semantics3 lies in the difficulty of separating relevant results from completely irrelevant ones. While many products exist in the database, the descriptions of each vary widely in their formatting. This makes it extremely hard to decipher if returned results really correlate to the food item in question. For example, it is hard to filter out results such as peanut M&Ms when the query was for peanuts. For this reason, the semantics3 system is not currently integrated into the application. However, adding the ability to provide information on all the food items found in semantics3 would increase the utility of the system.

2.2.5   Summary of Databases

Each of the databases described covers a different subset of possible food entries and offers different capabilities in terms of ease of use. The priority in building this system was to allow users to get reasonable database responses for the majority of basic food items, while allowing refinement and brand specificity whenever possible. After assessing each database in relation to those goals, the USDA SR was chosen as

the major food database. Specifically, the variety of items, reliability, and dependable structure of the SR made it the best choice. An additional database was needed to fill the major gaps of knowledge in the SR database, so a subset of the Nutritionix database was also added to the system. Eventually, the use of all four databases could present a more robust set of food items, but a combination of the SR and Nutritionix proved the most useful place to start.

Chapter 3

Generating Nutritional Matches for Food Items

The goal of our application is to create a dialogue system that uses spoken meal descriptions to extract food concepts and eventually map them to entries in nutritional databases. This process involves two major components: extracting the food concepts from the natural language, and determining their relationship to actual nutritional information.

3.1 Extracting Information from Natural Language

The first major step in translating the natural text to database entries is interpreting the food concepts and relationships between them from the text. This process involves three main steps: identifying the user's spoken input, assigning semantic tags to relevant food concepts, and finally grouping the food concepts with their associated foods [16].

To help with audio capture and recognition, the system uses the SpeechRecognition API from the W3C Web Speech API [17]. This API can support both user specified start and stop capture, continuous listening and specified language. Currently, the system only works in Google Chrome.

Once the words are recognized, the system semantically tags food concepts and
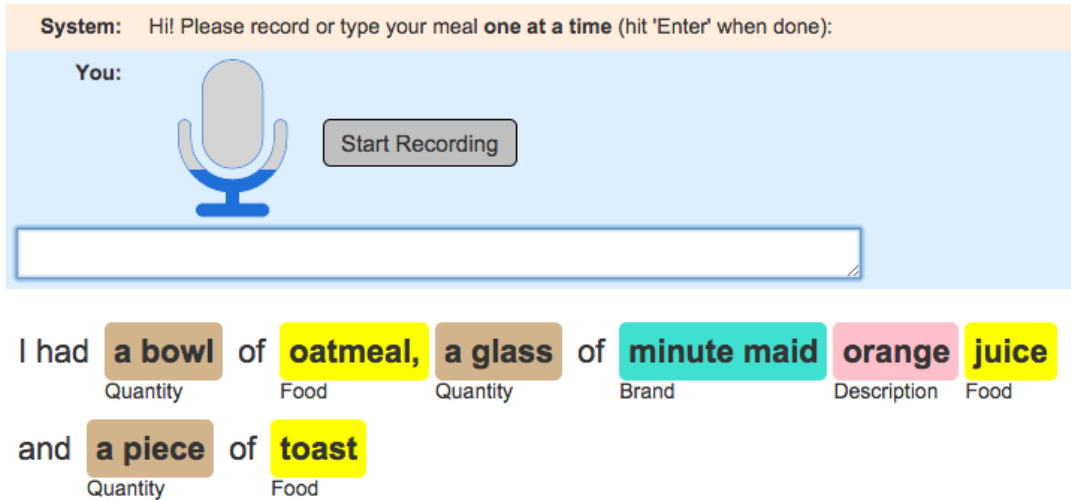
Figure 3-1: After the food diary goes through semantic tagging, labels are applied to the relevant elements.
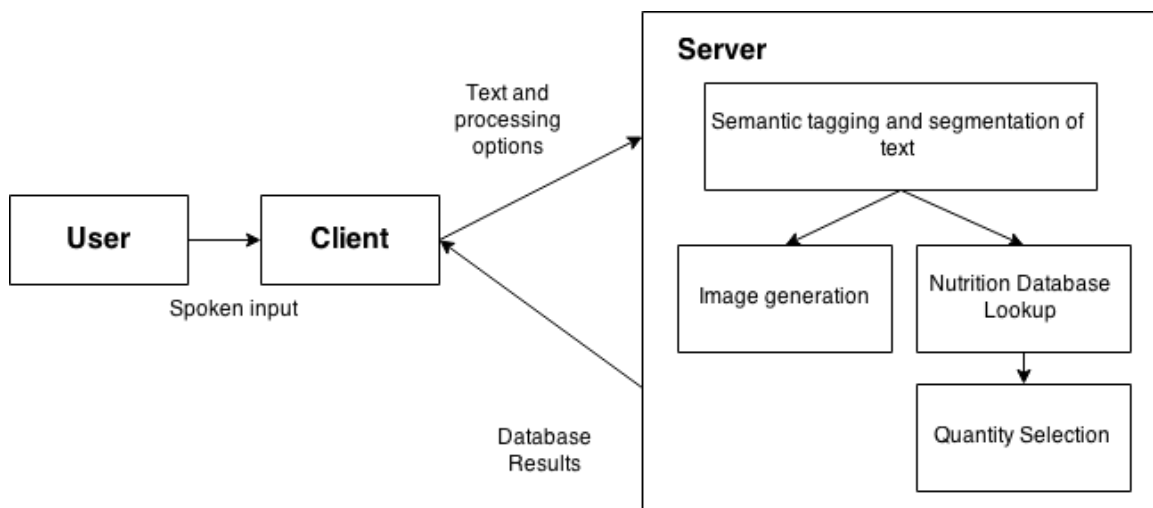
Figure 3-2: The current system has a variety of steps involved in the final processing of spoken input.

properties in the meal description. This can be seen in the system diagram in figure 3-2. The goal of this step is to recognize each segment which is a food item, description, quantity, or brand. To accomplish this, the system uses a conditional random field (CRF) model[18] [19]. The data for this model came from food diaries and labeling tasks performed on Amazon Mechanical Turk [20].

Finally, the system uses a model to associate attributes with corresponding foods. For example, in the sentence "I had a bowl of frosted flakes and a cup of orange juice," it is important to distinguish that the quantity "a cup" should be associated with orange juice and not with frosted flakes.

3.2 Generating an Item Cache

Once the food items and their relevant descriptions and brands are interpreted, the next major task is linking them to entries in nutritional databases. As discussed, one of the major goals of this project was to make the application as efficient as possible for the user. With that in mind, the application should be able to reconcile most food items to reasonable database entries without additional user input. The application aims to start with a reasonable nutritional profile for each spoken food item, then allow further opportunities to refine database results to match the specific food item eaten. In order to best accomplish this, we constructed a cache of many typical food items and their corresponding database entries. This cache is used to generated default results for many food items.

3.2.1 Gathering Food Items

In order to generate a list of food items, we used an open source database known as Freebase. Freebase is a collaborative database that was started in 2007 by a software company named Metaweb [21]. The data in it is from user submitted contributions as well as a variety of other sources. Freebase contains a large variety of categories of information, including a long table of food items. Of the 8,619 food item names in Freebase, 916 contained data submitted by users on their equivalent USDA SR

Instructions

Find the relevant nutritional database entry for the given food item.

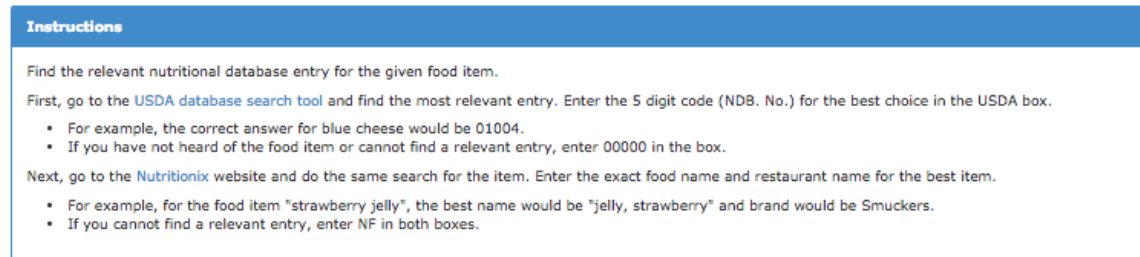First, go to the USDA database search tool and find the most relevant entry. Enter the 5 digit code (NDB. No.) for the best choice in the USDA box.

- For example, the correct answer for blue cheese would be 01004.
- If you have not heard of the food item or cannot find a relevant entry, enter 00000 in the box.

Next, go to the Nutritionix website and do the same search for the item. Enter the exact food name and restaurant name for the best item.

- For example, for the food item "strawberry jelly", the best name would be "jelly, strawberry" and brand would be Smuckers.
- If you cannot find a relevant entry, enter NF in both boxes.

Figure 3-3: In the mapping AMT task, users were given the above instructions.

database entries. For example, the Freebase entry "apple" referenced the corresponding entry in the SR for "apples, raw, with skin." These 916 items were loaded into the application's mySQL database, and then used as part of the cache.

3.2.2 Mapping Food Items to Database Results

Due to the incomplete nature of how the Freebase data was maintained, many items that had logical best answers in the USDA SR database did not have Freebase equivalents. To figure out which of these item names had nutritional matches and what those matches were, we used crowd sourcing through Amazon Mechanical Turk.

As shown in Figure 3-3, each Human Intelligence Task (HIT) asked the worker to reference both the USDA SR and the Nutritionix databases. The goal was to find a best value for each food item, and using both databases increased the number of items for which this was possible . As discussed, the preference in terms of reliability was to use the SR database, but it was necessary to use Nutritionix in order to serve a wider variety of food items.

Figure 3-4 shows an example of a typical HIT. Each worker was given a food item, in this case "Tomato Soup," and asked for equivalent values in both the USDA and Nutritionix databases. For the Nutritionix matches, we requested both the item name and brand in order to be able to most accurately map their responses to true Nutritionix entries. Figures 3-5 and 3-6 show examples of the information seen by the workers when they go to the USDA SR and Nutritionix websites respectively. Each HIT was given to three workers, with the majority used to determine the best option. Of the items that returned with any nutritional matches, 61.0% had agreement of all

Food item: tomato soup

USDA:

06498

Nutritionix:

Nutritonix food name:

Tomato Soup

Nutritionix brand:

Campbell's Condensed Soup

Submit

Figure 3-4: This is an example of a smaple AMT HIT for the item tomato soup.

three workers. Another 33.2% had agreement between two of the three. In the 5.8% of cases where agreement between at least two workers were not found, we determined what the most relevant option was manually. For future iterations, these could be given to more workers to try and find agreement. The 7,702 food items that did not have USDA equivalents all were incorporated into these tasks.

3.2.3   Status of the Cache

After information was collected from Freebase and the AMT tasks, the application had a cache of 4,418 food items. Of the 7,702 items that were given as AMT HITs, 2,134 of them returned with most relevant hits in the USDA SR. AMT also produced Nutritionix matches for 3,433 food items. In total, 3,502 food items were able to be mapped to nutritional database entries using AMT. Combined with the 916 items in Freebase that already had mappings, these helped form the cache of 4,418 items.

There were a variety of reasons why this final number is less than the total number of food items in Freebase. Since Freebase is an open source database, a variety of the food items contained in it were not truly types of food. Additionally, many of the entries were for specific foods from diverse regions. Both of the databases used concentrate primarily on American cuisine, and thus they did not contain entries for all of the items.

27

NDL Home   Food Search   Nutrients List   Ground Beef Calculator   Documentation and Help   Contact Us

Basic Report: 06498, Soup, tomato, dry, mix, prepared with water

Return to Search Results   Full Report (All Nutrients)   Statistics Report   Download (CSV)   Print (PDF)

Nutrient values and weights are for edible portion.

Apply Changes

Nutrient	Unit	1 Value per 100 g	1 cup 8 fl oz 265g
Proximates			
Water	g	89.88	238.18
Energy	kcal	38	101
Protein	g	0.93	2.46
Total lipid (fat)	g	0.61	1.62
Carbohydrate, by difference	g	7.17	19.00
Fiber, total dietary	g	0.4	1.1
Sugars, total	g	3.90	10.34
Minerals			
Calcium, Ca	mg	29	77

Figure 3-5: The SR database contains a variety of nutritional values for each entry.

Restaurants   Foods   Submit Your Data   API   Blog   Login

nutritionix   Search restaurants or foods

You are here: Campbell's Condensed Soup > Tomato Soup

Tomato Soup
Campbell's Condensed Soup

Nutrition Facts
Tomato Soup
Serving Size: 0.5 cup condensed soup
Servings Per Container 3.5

Amount Per Serving
Calories 90   Calories from Fat 0

                              % Daily Value*
Total Fat 0g   0%
  Saturated Fat 0g   0%
  Trans Fat 0g
  Polyunsaturated Fat 0g
  Monounsaturated Fat 0g
Cholesterol 0mg   0%
Sodium 480mg   20%
Total Carbohydrates 20g   7%
  Dietary Fiber 1g   4%
  Sugars 12g
Protein 2g

More Campbell's Condensed Soup Results

Other items from Campbell's Condensed Soup

	Calories
Broccoli Cheese	100
Southwest-Style Bean & Barley	90
Cream of Chicken	120
Vegetarian Vegetable	90
Old Fashioned Vegetable	80

Figure 3-6: Nutritionix allows users to view nutritonal values of items on their website.

This cache represents a large variety of food items and helps increase accuracy and efficiency of the system for those items. As additional food items not contained in the cache are logged on the system, the cache can easily continue to both grow and be refined. Appendix A details the steps necessary to add new items to the cache.

## 3.3   Generating Nutritional Matches outside of the Cache

While the cache helps generate best matches for a large number of food items, the list is not comprehensive, and the system needs to be able to handle user entries that were not part of the original Freebase entries. Specifically, the system aims to be able to deal with every food item that has a reasonable best entry in the USDA SR database. There were two important considerations when deciding how to determine which results to return to the user. The first was recall, assuring that the correct item was in the result set as often as possible. Another important consideration was precision. To accomplish the goal of an efficient and seamless user experience, keeping the result set as small as possible was very important.

One big consideration in determining best matches was how to deal with the differences between data labeled "food item," "brand," and "description." In some cases, the description is an important part of the actual food name, and in some it is not as important for returning the best match. For example, as shown in figure 3-7, if a user said "I had almond milk," "almond" is a description of "milk," but the most relevant database entries would most likely contain the exact phrase "almond milk." In contrast, if a user specified something like "I had round pancakes," the system needed to be able to find results for "pancakes," and realize that "round" was not an important description in this case.

Brands posed a similar problem. In many situations, users want to specify specific brands in order to get the most accurate nutritional information. When the specific brand is an entry in the SR, that is often the most specific and best option to return.

Figure 3-7: This example shows the challenge in dealing with informative vs non informative descriptions.
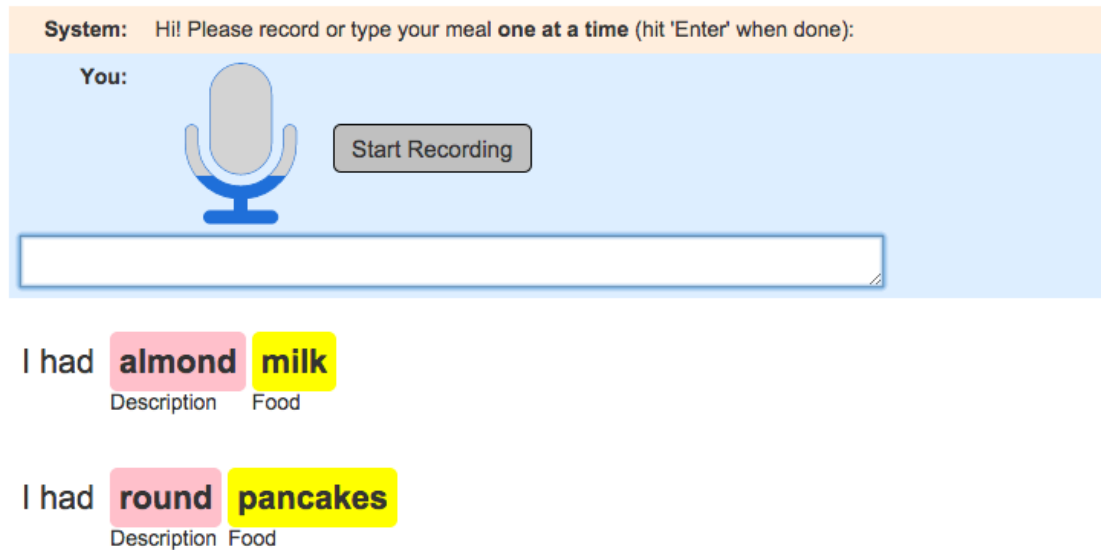
However, if the specific brand is not in the database but a generic version is, the generic version will still give relevant nutritional information and should be returned.

With these considerations in mind, we used a hierarchical model, searching first for the most specific answers given the description and brand details provided, and then backing off to include more general answers. The overall goal was to assure that no relevant results were missed, while only showing the most specific and likely ones if possible. The overall flowchart used by the system is shown in Figure 3-8.

A key insight used in determining this hierarchy was the information found in the formatting of food descriptions in the USDA SR. In general, each database entry begins with a food item, which is followed by further descriptions, each separated by commas. For example, Figure 3-9 shows the entries that begin with the word "tomatoes." This way of determining which adjectives correspond to which database items was used in filtering entries based on descriptions. Brands were difficult to differentiate, as they normally occur as a modifier after the food item, similar to any other adjective. Generally, the brand is one of the most important descriptions, and

Figure 3-8: This shows the system flow to decide the most relevant database match. For each food item with accompanying description and brand, these steps are used to decide which items to return.
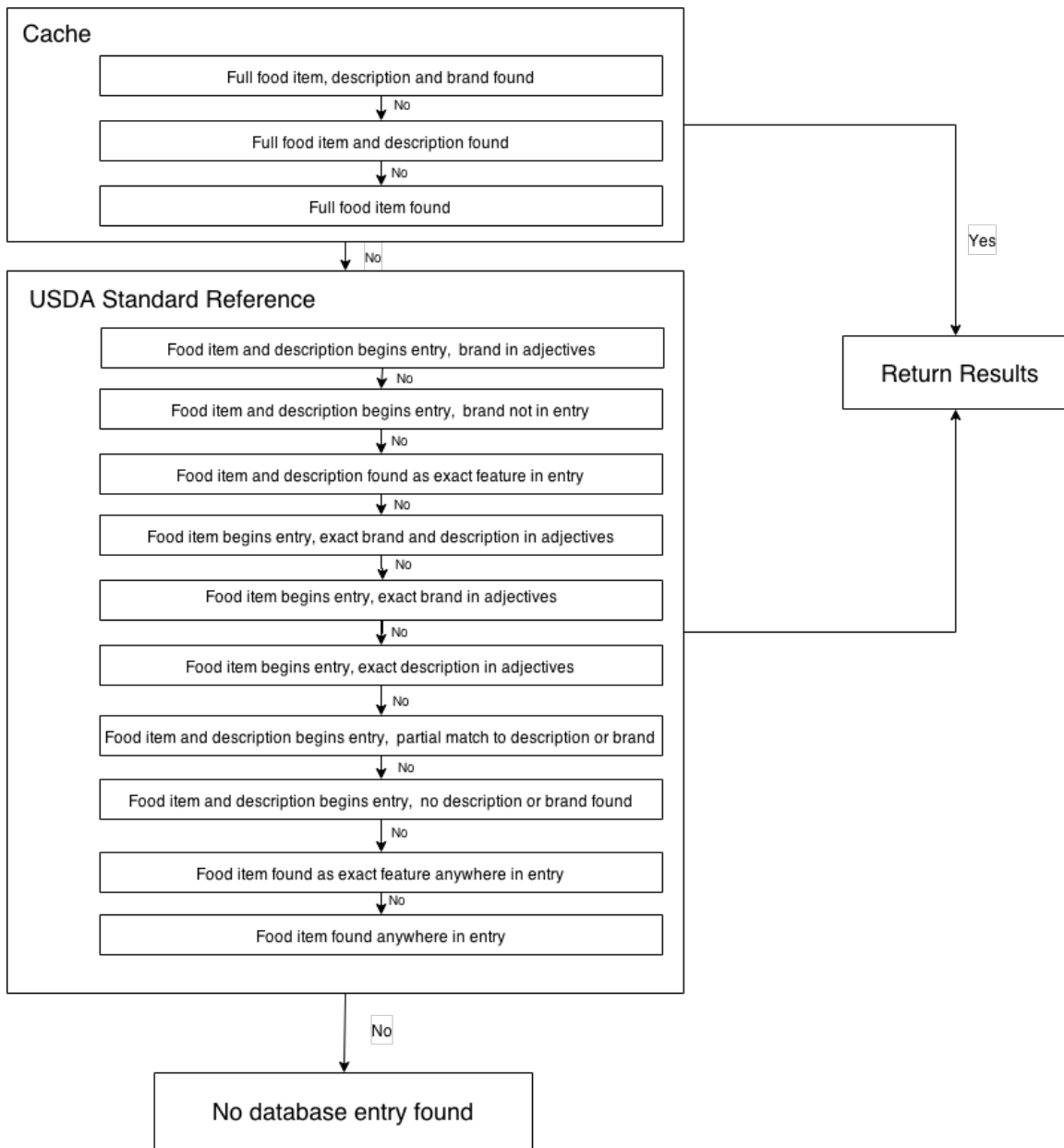
```
┌─ Cache ──────────────────────────────────────────────┐
│  ┌─────────────────────────────────────────────┐     │
│  │  Full food item, description and brand found │     │
│  └─────────────────────────────────────────────┘     │
│                     │ No                              │
│  ┌─────────────────────────────────────────────┐     │
│  │  Full food item and description found        │     │
│  └─────────────────────────────────────────────┘     │
│                     │ No                              │
│  ┌─────────────────────────────────────────────┐     │
│  │  Full food item found                        │     │
│  └─────────────────────────────────────────────┘     │
└──────────────────────────────────────────────────────┘
                      │ No                         Yes
┌─ USDA Standard Reference ─────────────────────────────┐   ┌──────────────────┐
│  ┌──────────────────────────────────────────────────┐│   │  Return Results  │
│  │ Food item and description begins entry, brand in  ││   └──────────────────┘
│  │ adjectives                                        ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item and description begins entry, brand not ││
│  │ in entry                                          ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item and description found as exact feature  ││
│  │ in entry                                          ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item begins entry, exact brand and           ││
│  │ description in adjectives                          ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item begins entry, exact brand in adjectives ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item begins entry, exact description in      ││
│  │ adjectives                                         ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item and description begins entry, partial   ││
│  │ match to description or brand                      ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item and description begins entry, no         ││
│  │ description or brand found                          ││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item found as exact feature anywhere in entry││
│  └──────────────────────────────────────────────────┘│
│                     │ No                              │
│  ┌──────────────────────────────────────────────────┐│
│  │ Food item found anywhere in entry                 ││
│  └──────────────────────────────────────────────────┘│
└───────────────────────────────────────────────────────┘
                      │ No
          ┌────────────────────────────┐
          │  No database entry found   │
          └────────────────────────────┘
```

Figure 3-9: This figure shows examples of entries in the USDA Standard Reference database for tomatoes.

11693	Tomatoes, crushed, canned
11527	Tomatoes, green, raw
11695	Tomatoes, orange, raw
11531	Tomatoes, red, ripe, canned, packed in tomato juice
11885	Tomatoes, red, ripe, canned, packed in tomato juice, no salt added
11533	Tomatoes, red, ripe, canned, stewed
11537	Tomatoes, red, ripe, canned, with green chilies
11530	Tomatoes, red, ripe, cooked
11660	Tomatoes, red, ripe, cooked, stewed
11884	Tomatoes, red, ripe, cooked, with salt
11529	Tomatoes, red, ripe, raw, year round average
11955	Tomatoes, sun-dried
11956	Tomatoes, sun-dried, packed in oil, drained
11696	Tomatoes, yellow, raw

can get the most specific result if it is found, so that was kept in mind in determining the order of queries. A more detailed description of the database tables used in the system is provided in Appendix B.

3.4 Dealing with Plurals

An important consideration when finding nutritional database matches for food items, both in the cache and outside of it, was the importance of reconciling plural and singular words. The same database results should be returned whether the user specifies that they had a "pancake" or "pancakes," and yet combining the two is not a trivial problem. The USDA SR also has some inconsistency in their entries in terms of plurals. For example, there is an entry for "Egg, whole, cooked, poached", and also one for "Eggs,scrambled, frozen mixture" however both should be relevant for queries for "egg" or "eggs."

The rules for converting between singular and plural words in English are quite

complicated, so we used a Java package called Evo Inflector [22], based on a paper by Damien Conway [23]. Evo Inflector uses a formulaic approach and can convert between singular and plural words in both directions. When tested on the entire set of 163,518 from Wiktionary, it was able to return correct answers for 68.4% of them, including all 979 specified as basic. The possibility of incorrect conversions is still a concern, but the use of this package shows a marked improvement. In every database call, the "singular" and "plural" versions of every food item are computed, and items that match any of them are returned. This assures that results for the correct food item are returned, regardless of pluralization.

3.5   Performance of System

The performance of the system was assessed using AMT. The HITs aimed to assess the performance of the system on three main components: the language tagging, the database matching, and the quantity selection. As shown in Figure 3-10, in each HIT, workers recorded or typed at least two meal descriptions. For each meal description they entered, the sentence with tags was shown along with the table of food items and quantities. Workers then were asked to assess whether each part of the system performed correctly. An example of a partially completed HIT is shown in Figure 3-11.

In the first iteration of this assessment task, 383 workers completed it. Overall, they wrote or spoke 870 food diaries, containing 1860 food items. Of the 804 food items that were tagged correctly, 674 were also labeled correctly. This means that 83.8% of food items that were labeled correctly were successfully mapped to equivalent database results.

Many of the items that did not successfully map to database results were specific prepared items that were not in the cache. For example, there is no entry for "quiche" or for "apple turnover" in the USDA SR, and it was not in the Nutritionix cache so a correct result was not found. This showed the importance of continually refining and expanding the cache. As additional food diaries are collected, the new food items

33

Figure 3-10: This figure shows an example of the AMT task used to assess performance.

Food Log

Instructions

Please record **two** meals (e.g. what you ate for breakfast, lunch, dinner, or snack today or yesterday) using as much detail and accuracy as possible. Be creative--we will not accept repeat answers. Try to include as much additional information as you remember, such as brand names and quantities. Please press the record button below to describe your meal orally, or type in the textbox. It's more fun to record your meal verbally, but it requires using Chrome.

Please interact with the system to narrow down the USDA hits to one food. Then **check the boxes** in the rightmost column if the labels are correct, if the quantity is correct, and if the final USDA hit correctly matches the food you actually ate.

If you encounter any errors or have feedback from your experience using the system, please let us know!

Examples

I had a boiled egg, a Thomas's english muffin, and an ounce of organic butter.

For lunch I ate a plate of spaghetti with a spinach salad and feta cheese.

System: Hi! Please record or type your meal **one at a time** (hit 'Enter' when done):

You:

Start Recording

(Optional) Please let us know how we can make this HIT better.

Submit

Figure 3-11: This figure shows the AMT task once a food has been logged.

I had **a large** **banana** and **a cup** of **yogurt**
   Quantity   Food   Quantity   Food

Food	Quantity	USDA Hits	Are the color-coded labels in the above sentence correct?	Is the quantity correct?	Is the USDA hit correct?
Banana	Quantity: 1 large (8" to 8-7/8" long)	Bananas, raw, Calories: 121 Source: USDA • See more options	☑	☑	☑
Yogurt	Quantity: 1 cup (8 fl oz)	Yogurt, plain, whole milk, 8 grams protein per 8 ounce, Calories: 149 Source: USDA • See more options	☑	☑	☑

collected can be added to the cache in order to make it more extensive and the system more robust.

Chapter 4

Generating Matches for Quantities

4.1 Overview of Problem

In order to successfully track calories and daily values of nutrients, an important step was figuring out how to correctly discern quantities of food items eaten. As discussed, efficiency was a major goal of the system, and correct interpretation of quantity was crucial in accomplishing that. Ideally, users should be able to describe their meal and immediately view correct calorie information, but with an incorrect quantity those numbers could be significantly incorrect. There are a variety of components that make this task difficult, mostly due to the lack of standardization of a variety of words used to define quantities. With that in mind, the measure of a successful system was defined as one that could reasonably translate quantity amounts said into equivalent quantities that directly map to nutritional values.

4.2 Strategy for Finding Best Quantity

There are two main steps in determining the best quantity. First, the system identifies the most relevant numerical amount from the spoken quantity. To do this, it uses a pattern matching technique, looking for digits or words in the quantity tag. The system can recognize digits and fractions written, as well as words for many integers and common fractions. In the absence of a relevant number, the assumed numerical

Figure 4-1: This figure is an example of a food item with a quantity chosen.

quantity is one. As shown in Figure 4-1, this amount is easily modified by the user.

The next step is determining proper units for the quantity. All entries in the USDA SR database have corresponding options for units of quantity. Each quantity option in the database also has a specified gram weight that can be used to determine the overall number of calories. The system aims to map the spoken quantity units to the most relevant option in the SR database. The Nutritionix database only returns one unit option per food item, in which case that unit is used. An overview of the steps taken to select the best quantity are shown in figure 4-2.

To do the unit mapping, the application first checks if any of the units from the database are found in the tagged quantity, and selects one as the most relevant if it is found. If no exact unit matches are found, but one is similar (for example, the user says "a large" and a database option is 'large, 2" diameter'), then that unit will be chosen. The system also leverages common conversions to allow relevant matches to be found. For example, if a user specifies that they had "1 cup," and cup is not an option but "tbsp" is, then tbsp will be chosen and the numerical amount changed accordingly. An example of this is shown in Figure 4-3, where teaspoons are converted into cups.

Figure 4-2: These are the steps taken by the system to decide which of the weight options in the USDA SR is the most relevant to the quantity said by the user.
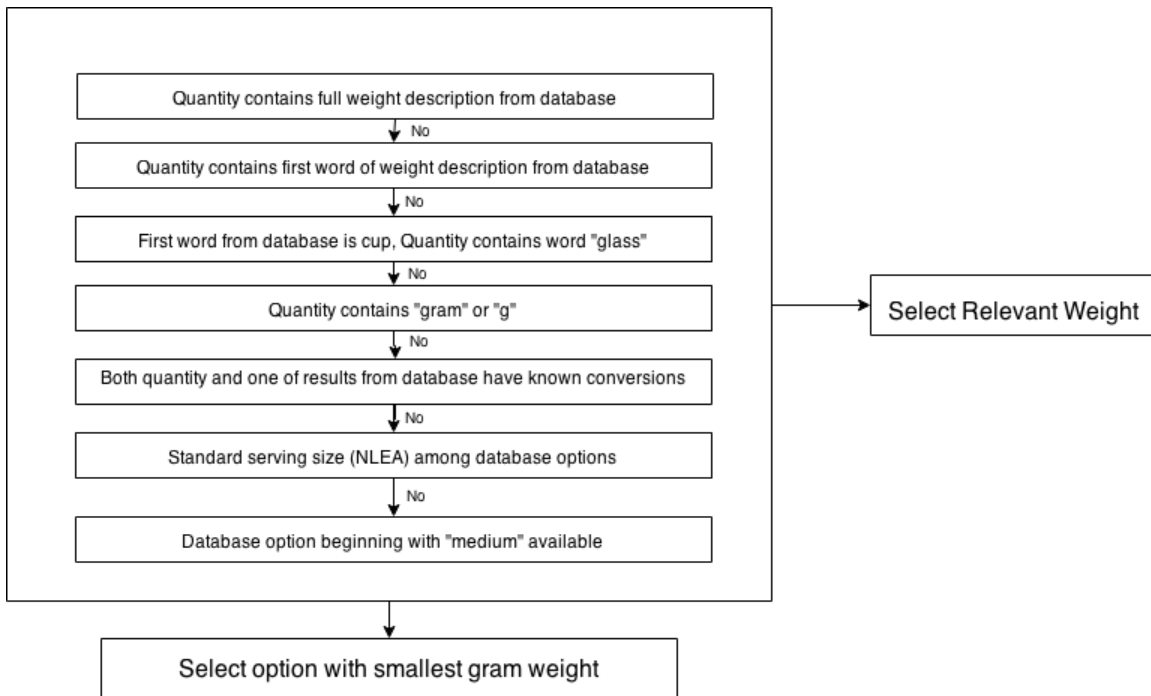
Figure 4-3: This is an example of a quantity unit conversion.

Figure 4-4: This shows a food item entry without a quantity specified.

4.3 Modifying Quantity Amounts

For each spoken food item that is successfully mapped to one database entry, the system finds a most likely quantity based on what is said, however it is important to allow users to manually edit the quantity via the user interface. For example, Figure 4-1 shows the response for a "cup of oatmeal." Users can modify the quantity directly by typing in the text box or selecting a different unit from the dropdown menu in the "Quantity" column of the table.

Users can also edit the quantity using speech or text. For example, in Figure 4-4, the user initially does not specify how much milk they drank, and so it defaults to one cup. After they specify further, the quantity is updated accordingly, as shown in Figure 4-5. Every update to the quantity will also change the number of calories shown next to the food item.

Figure 4-5: This figure shows a food Item Entry With Quantity Modified via Speech.

4.4 Evaluation of Performance

As described in Chapter 3, an AMT task was used to determine the performance of the system for picking a relevant quantity. Workers entered food diaries, and then assessed the accuracy of the system in terms of the food item chosen and corresponding quantity. Since the method for choosing a quantity is based on the relevant results from the database, it is difficult to determine if the quantity algorithm is working unless the food items are chosen correctly. Of the 804 items where the food item was marked as correct, 742, or 92.2%, also had a reasonable quantity chosen. Many of the errors were found in items were no quanity was specified, but the default value was not what was intended. For example, the default choice for "dill pickle" is a slice, because none of the other options for best choices were found. Another challenge with the quantities was in regards to items from NUtritionix which only have one choice for quantity. Sometime this is a confusing choice, such as 'oz" is the option for "french toast," and that was not the quantity that users were looking for.

Chapter 5

Refinement of Nutritional Matches

In order for the tracking of nutritional intake to be correct, it is critical to find the most accurate and specific database matches for each food item. While the system aims to find the most logical match for each spoken food, it is critical that the user be able to modify the result if it is incorrect, or provide more specificity if needed. In some cases, more information might be needed from the user as to which database option to choose. In the current system, users can refine database matches in a variety of ways to help achieve the highest accuracy of database matching possible.

5.1   Modification via Natural Language

If users want more specific results, the system allows them to modify food items via natural language, using either speech or typing. This is useful in situations where a user forgot to specify details about a specific food item, and wants to add them after seeing a database result. For example, as shown in Figure 5-1, when a user says "I had milk," the database result returned is for whole milk. If this is incorrect, the user can change the result simply by saying "It was skim," and the database hit will update accordingly. It is also easy to add descriptions to existing foods even if there are multiple food items in the table. As shown in Figure 5-2, if the user says a description in addition to the food item they want to add it to, the food item is updated accordingly. This is all similar to the ability to modify quantity which is

Figure 5-1: Food Item Entries can have the description modified via speech. The system defaults to the latest option if none is specified.

described in the previous chapter.

5.2 Database Search Backoff

While it is important to be able to generate one best match, sometimes the answer chosen may not be in the set of results shown. In that case, the user needs to be able to expand the search parameters in order to find the most relevant database item. This is implemented in the current system via a "See more options" button.

As discussed in Chapter 3 and illustrated in Figure 3-8, the application uses a hierarchical model with successive steps to determine the best set of results. By

Figure 5-2: If the food name is used, the food item entry can be directly modified via speech.

I had **a cup** of **yogurt** and **a strip** of **bacon** for breakfast
       Quantity   Food   Quantity   Food

Food	Quantity	USDA Hits
Yogurt	Quantity: 1 cup (8 fl oz) ⇕	Yogurt, plain, whole milk, 8 grams protein per 8 ounce, Calories: 149.45 Source: USDA • See more options
Bacon	Quantity: 1 slice raw ⇕	Pork, cured, bacon, unprepared, Calories: 116.76 Source: USDA • See more options

It was **greek** **yogurt**
     Description Food

Food	Quantity	USDA Hits
Yogurt	Quantity: 1 container ⇕	Yogurt, Greek, plain, nonfat, Calories: 59 Source: USDA • See more options • Back
Bacon	Quantity: 1 slice raw ⇕	Pork, cured, bacon, unprepared, Calories: 116.76 Source: USDA • See more options

Figure 5-3: This shows the first result of searching for bread.

default, it shows the results from the highest level of specificity that returns results. The "See more options" button allows the user to see the results returned from lower levels on the hierarchy. For example, in Figure 5-3, the application system first reconciles "bread" to the USDA entry "Bread, white, commercially prepared." If that is incorrect, but the user is not sure what other options are possible to refine it, they can click "See more options" and see what is shown in Figure 5-4. Here, they can select one of the other adjectives to narrow down the search, or see more options to expand the search again, as shown in Figure 5-5.

5.3 Finding Informative Adjectives

For many food items, the basic information logged may not be informative enough to narrow down the options in the database to one best option. For example, if the user says they had "milk," and the default whole milk option is incorrect, they need to effectively be able to narrow down the other possible options. Adding additional descriptive adjectives is the most effective way to do this, but sometimes it is unclear which additional pieces of description would be most helpful. To aid with that, the system uses the structure of entries in the SR database to generate a list of most relevant adjectives to help the system narrow down to one database result. For example, Figure 5-4 shows an example of the adjectives that are generated to help

46

Figure 5-4: After See More Options Is pressed, the search for bread expands to different types of bread.

Figure 5-5: After See More Options is pressed for a second time, the search expands to all foods containing the word bread.

narrow down a list of possibilities for the item "pancakes."

The USDA SR database has a consistent format for describing each of its entries. In general, each entry begins with a food item, and then a series of adjectives follows, each separated by commas. Generally the descriptions become increasingly more specific. For example, one database entry is "Cheese, cottage, lowfat, 2% milkfat." In this case, cottage is the most relevant adjective, and the others become more relevant as there is more need for specificity. This format of descriptions can also be seen in Figure 3-9, which shows the the different entries that are relevant to the food item Tomatoes.

While multiple entries for a given food item may vary by a couple of adjectives, some adjectives are much more informative than others. One option is to prompt the user with a list of all adjectives related to a food item to help narrow down the entries. However fewer options is easier, and if some do not relate to actual nutritional differences, they do not need to be included. Thus, the goal of the system is to produce the smallest set of adjectives to prompt the user, while still allowing them to narrow down to the most specific database item necessary.

To determine the best adjectives, we looked at each entry as a graph of ordered nodes, with each node representing an adjective. Food entries that share an adjective are merged into one branch. Figure 5-6 shows an example tree generated for the 15 entries in the USDA SR database that begin with the word Tomatoes. Each adjective is represented with a square node, with the calories per 100 grams for each database entry shown in the circular nodes.

The trees generated are then used to figure out which adjectives are needed. For each node of the tree, if the items associated with it are all similar enough in calories, then it is not important to differentiate between them. Database entries that are within 15 calories or 10 % of each other are considered close enough to be combined. For example, all of the food items that begin with "red, ripe, canned" are similar in calories. Additionally, once a node only represents one food item, all additional adjectives are unnecessary. In the tomatoes example, that means that if the tomatoes are green, it is not important to ask if they are raw in order to decide

on the correct food item. Thus, the graph is compressed and users are only prompted with adjectives that remain in the compressed version. Figure 5-7 shows an example of the compressed tree for tomatoes.

Using these adjectives, the system can efficiently prompt users for the most helpful information to narrow down the database search. With each adjective choice, the trees can be remade without the adjectives chosen, and thus the search can converge quickly onto the most relevant database item.

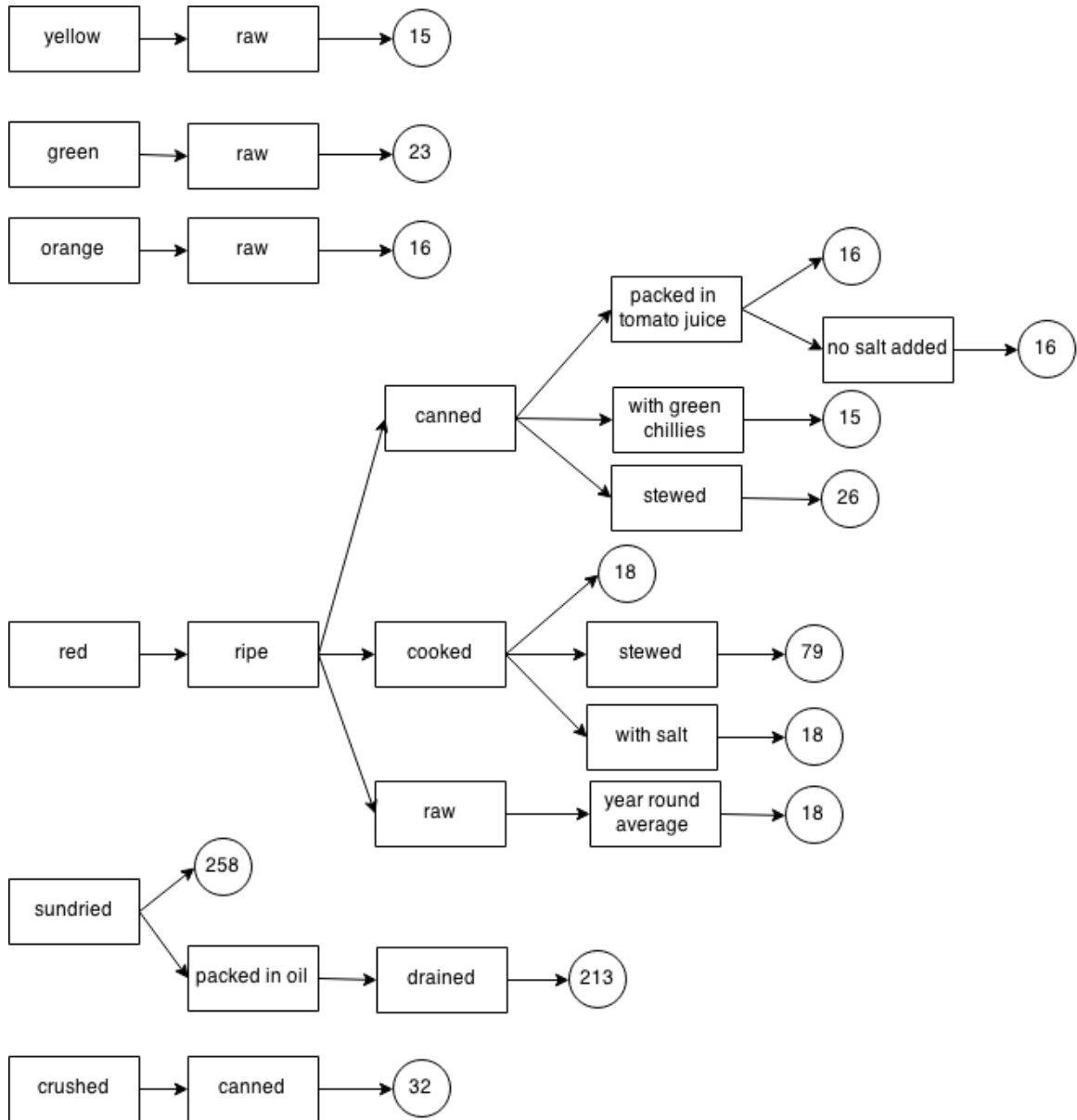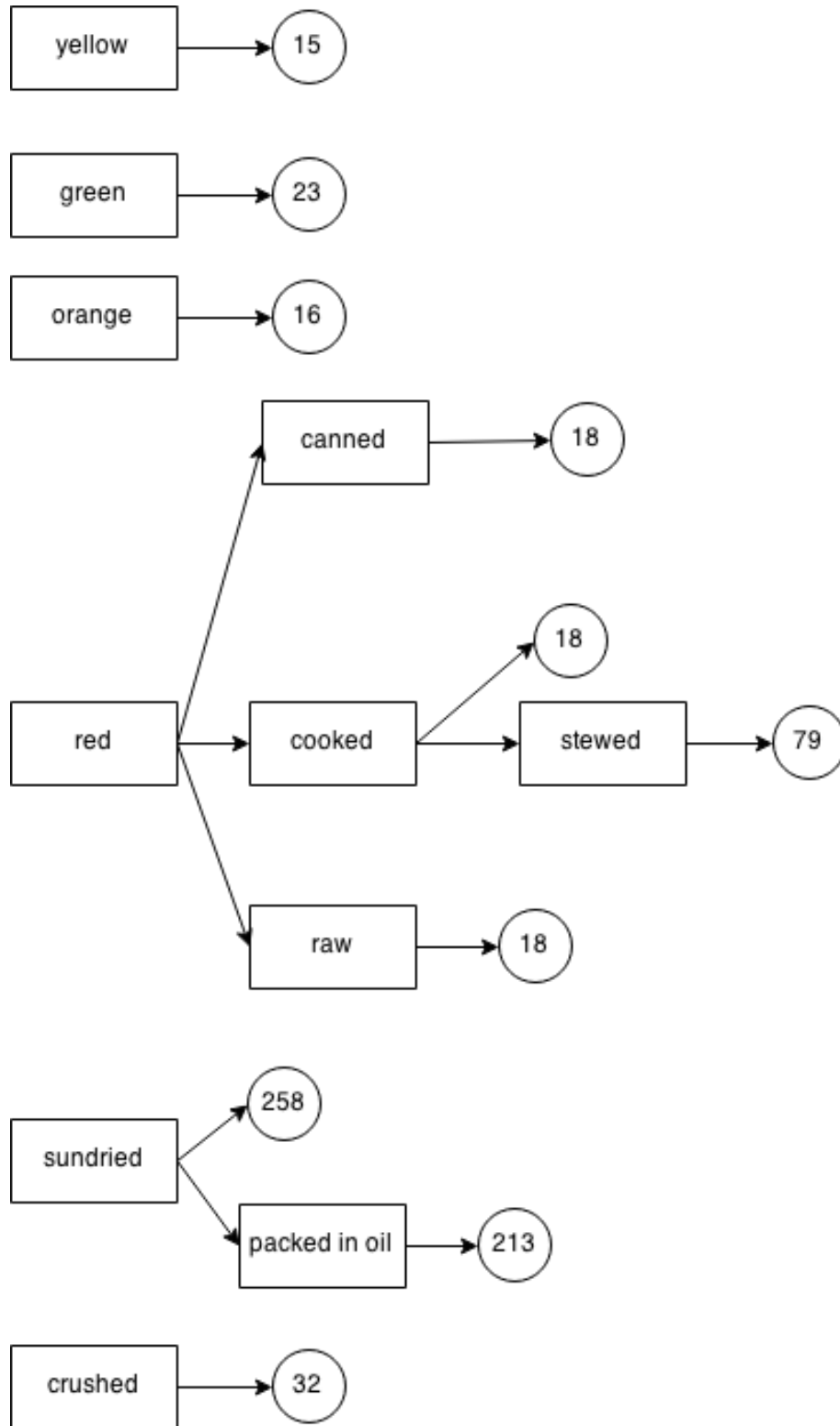Figure 5-6: This is an example of a tree of adjectives for tomatoes before compression.

Figure 5-7: This shows a tree of adjectives for tomatoes after compression.

Chapter 6

Conclusion

6.1   Summary of Contributions

Tracking nutritional intake is a key component to managing a variety of diseases as well as aiding good nutrition. However, most methods currently used are quickly abandoned or require a large amount of human labor to interpret. The application described, which converts spoken food diaries into nutritional database entries, presents a solution to this problem.

Using the USDA SR and Nutritionix databases, we constructed a cache of over 4,000 possible spoken food items and their nutritional matches. We also created a hierarchical system for determining the most relevant database entries from the brand, descriptions and food items specified. Through our multiple options for database refinement, users can further narrow down database choices and fix incorrectly chosen entries. On a set of food records gathered from 500 users on AMT, we were able to reconcile 83.8% of food items that were semantically tagged correctly to relevant nutritional database entries. We also explored a variety of pattern based solutions for understanding quantities and their numerical interpretations, and were able to reasonably interpret 92.2% of quantities from the AMT food diaries.

6.2 Future Work

While the work outlined represents significant work on the system prototype, a variety of future steps could lead to better performance and higher usability.

6.2.1 Nutritional Analysis

One key component missing from the current prototype is the analysis of actual nutritional intake based on the items found. Right now, the main goal of the system has been to recognize what was said and translate it into database items. In order to have this system really be functional and used by both patients and nutritionists, it needs to bring together this information to provide statistics such as percentages of daily nutrients from each meal. The information is available, but more work with the user interface is needed to bring the prototype to a more useful state.

6.2.2 Personalization

Adapting the system to specific users presents the opportunity for both increased usability and better performance. As described, a variety of food items have many logical best choices, and the only option is for the system to ask the user. If the system could store these answers, it would provide better guesses a larger percentage of the time. Personalization would also allow users to specify which types of food they are more likely to eat (for example skim versus whole milk), and thus require less information to be gathered in each diary. Personalization of database responses would increase the performance and efficiency of the system, and thus prove an important step in increasing usability. The system's ability to remember food items from a specific person is also key if it is going to be used to track food over multiple days.

6.2.3 Cache Expansion

A final major area that can be expanded is the cache. As more food diaries are gathered, both the caches of USDA SR and Nutritionix items can continue to expand.

The new food items generated from more food diaries can be converted into cache entries in the same way described in Section 5-3, until eventually most items will be found in the cache. A cache of most popular food items could increase the efficiency and performance of the system overall.

Appendix A

Overview of Code

Below is an overview of the major classes and methods contained in the code of the application which implement the components described in this thesis.

A.1   Database Reconciliation

In performing the overall database reconciliation of descriptions to food items described, a six major classes were involved. There are three major classes of database hits used: USDAItem, CompositeItem, NutritionixItem. As described, USDALookup is the class that contains most of the logic for the database lookups , and contains methods to generate USDAResults. The USDAResults also contain USDAWeights used in quantity reconciliation.

- **USDALookup:** -The USDALookup class contains all of the logic for finding the best hits from the database based on the input from the user (either from a call to Images or FeatureLookup). Some of the main methods contained in this class are outlined below.

  - **leveledFoodItemLookup:** - This is the method that performs the overall food lookup based on a food item, brand, descriptions, quantity, and starting level. It returns a USDAResult to be sent back to the client. The starting level is defined as which level the system is currently on in the

database flow shown in Figure 3-8. The starting level is used to support the "See more" buttons described in section 5.2, and the added adjectives are used in cases where the user selects additional adjectives from the list as detailed in section 5.3.

– **foodIemInitialLookup / foodItemAdjectiveLookup:-** These are the methods that respond to requests from the client. foodItemIntialLookup is used in the initial call to reconcile the list of food items to database items, and foodItemAdjectiveLookup is used in subsequent refinement steps.

– **executeQuery/ executeWeightQuery/ executeNutritionixCache-Query:-** All three of these methods are used to actually connect to the mySQL and query the given databases. They are used whenever information needs to be obtained from any of the database tables.

– **getRelevantQuantities:-** Given a list of USDA SR 5 digits id numbers (NDB_Nos), this method returns all of the relevant quantity options from the SR database.

– **findFreeBaseEquiv:-** This is the method used to determine if there are any equivalent items in the USDA cache.

– **findNutritionixEquiv:-** This is the method used to determine if there are any equivalent items in the Nutritionix cache.

- **USDAItem:** - USDAItem represents a singular item in the USDA SR database. It contains the item's NDB_No, name and descriptions, as well as a variety of nutritional information about the specific item.

- **CompositeItem:** - A CompositeItem represents multiple USDAItems that were determined to be similar via the process described in section 5.3. Composite Items have a list of all the IDs which they represent, as well as information on the food item and features that these items have in common. Their nutritional information is determined by finding the mean of the values from the relevant food items.

- **NutritionixItem:** - NutritonixItem represents an item from the Nutritonix cache. Its foodID is always -1 to help differentiate it from a USDAItem, but it also contains a Nutritonix ID which helps the UI link it to the correct source URL.

- **USDAWeight:** - USDAWeight represents an option for a weight amount represented by the USDA SR WEIGHT table (see section B.1.3). It contains the list of foodIDs that it is relevant to, as well as a description and gram weight.

- **USDAResult:** - USDAResult is the final object that is passed back to the client application to represent the database results for a specific food item. It contains the item, description and brands relevant to the food item, as well as all adjectives to ask about, relevant hits, and relevant weights. This class also contains the logic to find the most relevant weight amount and unit and reorder the weights list so that it is first.

A.2   Adjective Trees

A variety of classes are involved in the formation of trees and compression of trees to create most the relevant adjectives as described in Chapter 5.3. The main classes and their significance are described below.

- **FSTGenerator:** - FSTGenerator is the class that contains the code for creating a tree and compressing it. The key method in this class is makeTree which creates a tree from a list of ReturnableItems.

- **FSTNode:** - FSTNode is the interface to represent any type of node in the tree. Three classes implement FSTNode: USDAItem, NutritonixItem and AttributeNode. Every FSTNode is able to compress the rest of the tree in a top-down manner by calling a compressTree method as well as return the items and attributes that are upstream of it. When an attribute is removed in the compression stage, it is marked as hidden as opposed to removed and not included in the attributesKept.

- **AttributeNode:** - An AttributeNode is a node in the tree that represents an adjective and not a specific food item (i.e., not a leaf node). It implements all of the methods described in FSTNode.

- **ReturnableItem:** - A ReturnableItem is an interface for any of the leaf nodes. Both USDAItem and CompositeItem implement this interface.

A.3   Loading the Database Cache

The database cache can be continually expanded to include more food items as they are gathered. Overall, most of the code for generating new items for the cache is found in the CacheGenerator class. The steps needed to go from food diary sentences to new items loaded into the cache are outlined below.

1. Create a file with the text of food diaries. Put each food diary on its own line.

2. Run the method processSentences in the class CacheGenerator, replacing the input and output file names and path. This method needs to be run while the system is running as it utilizes the NutritionixContext.

3. Using the output of processSentences, make all combinations of brand, description, and food item (in that order) that should be added to the cache. Put them into a file which contains each combination on a seperate line.

4. Run the method findNewItemsforAMT in CacheGenerator, again changing the desired path and file names. The output file contains the list of all food items that are not currently found in the cache.

5. Use the list created to start a Nutriton Database Mapping batch task on AMT. The output of this task will contain the new items that should be added to the cache.

6. Separate the results of AMT into items that should be loaded into the USDA and Nutritionix caches. Run the Nutritionix ones through the parseTestList

60

method in NutritionixFreebaseLoader to get their equivalent data for the cache. Load both sets into the cache using the corresponding database loading scripts.

Appendix B

Database Tables Used

There are primarily two types of mySQL database tables used by the application: ones containing information from the USDA Database for Standard Reference, and tables to hold the cache. The fields contained in each and what they represent are described below.

B.1   USDA SR Tables

The system primarily uses three tables available as part of the USDA SR database. The three tables used and their key fields for this application are detailed below. For further descriptions of these tables, see the USDA SR documentation[12].

B.1.1   FOOD_DES

FOOD_DES contains the descriptions of all of the food items available, and is used to find the most relevant food item and map it to a NDB_No. Each food item in the database has a unique NDB_No.

B.1.2   NUT_DATA

NUT_DATA contains the detailed nutritional information for each of the foods. Each nutritional value is per 100g.

Table B.1: FOOD_DES Database Table Outline

Field	Description
NDB_No	5-digit Nutrient Databank number
Long_Desc	200-character description of food item

Table B.2: NUT_DATA Database Table Outline

Field	Description
NDB_No	5-digit Nutrient Databank number
Nutr_No	Unique identifier code for a nutrient
Nutr_Val	Amount in 100g

B.1.3 WEIGHT

WEIGHT contains the relevant weights for each of the food items, and their conversion factors for calculating actual nutritional values using data from NUT_DATA.

Table B.3: WEIGHT Database Table Outline

Field	Description
NDB_NO	5 digit Nutrient Databank number
Seq	Sequence Number
Amount	Unit Modifier
Msre_Desc	Description (for example cup, tbsp, etc.)
Gm_Wgt	Gram Weight

B.2 Tables for the Cache

As described, the system maintains both a cache of most relevant USDA SR equivalents and Nutritionix equivalents to food items.

B.2.1 USDA Cache (freebaseEquiv)

This table is the cache of food items and their USDA SR equivalents.

Table B.4: freebaseEquiv Database Table Outline

Field	Description
name	Name of Food Item
fbid	Id in freebase if applicable
srid	NDB_No of most relevant item in SR

B.2.2 Nutritionix Cache (nutritionixCache)

This table is the cache of food items and their Nutritionix equivalents.

Table B.5: nutritionixCache Database Table Outline

Field	Description
name	Name of Food Item
fbname	Name from AMT
fbbrand	Brand from AMT
nutritionixID	ID of Equivalent Nutritionix Item
itemName	Name of Equivalent Nutritionix Item
brandName	Brand of Equivalent Nutritionix Item
servingQuant	Numerical Serving Amount of Equivalent Nutritionix Item
servingAmount	Quantity Unit of Equivalent Nutritionix Item
calories	Calories of Equivalent Nutritionix Item

Bibliography

[1] Ronilda Lacson and William Long. Natural language processing of spoken diet records (sdrs). In *AMIA Annual Symposium Proceedings*, pages 454–458, 2006.

[2] West Virginia Health Statistics Center. Obesity: Facts, figures, guidelines. https://www.wvdhhr.org/bph/oehp/obesity/mortality.htm.

[3] Food Research and Action Center. Overweight and obesity in the us. http://frac.org/initiatives/hunger-and-obesity/obesity-in-the-us/.

[4] World Health Organization. Obesity and overweight fact sheet, January 2015. http://www.who.int/mediacentre/factsheets/fs311/en/.

[5] F. E. Thompson, A. F. Subar, C. M. Loria, J. L. Reedy, and T. Baranowski. Need for technological innovation in dietary assessment. *J Am Diet Assoc*, 110(1):48–51, Jan 2010.

[6] J. S. Shim, K. Oh, and H. C. Kim. Dietary assessment methods in epidemiologic studies. *Epidemiol Health*, 36:e2014009, 2014.

[7] A-K Illner et al. Review and evaluation of innovative technologies for measuring diet in nutritional epidemiology. *The International Journal of Epidemiology*, 41:1187–1203, 2012.

[8] Stumbo PJ. New technology in dietary assessment: a review of digital methods in improving food record accuracy. *Proceedings of the Nutrition Society*, 72:70–76, 2013.

[9] Megan E Rollo, Susan Ash, Philippa Lyons-Wall, and Anthony Russell. Trial of a mobile phone method for recording dietary intake in adults with type 2 diabetes: evaluation and implications for future applications. *Journal of Telemedicine and Telecare*, 17(6):318–323, 2011.

[10] MyFitnessPal. Myfitnesspal. http://www.myfitnesspal.com/.

[11] Calorie Counter. Calorie counter. www.caloriecount.com/.

[12] Nutrient Data Laboratory US Department of Agriculture, Agricultural Research Service. Usda national nutrient database for standard reference, release 27, August 2014. http://www.ars.usda.gov/ba/bhnrc/ndl.

[13] Agricultural Research Service U.S. Department of Agriculture. Usda food and nutrient database for dietary studies 2011-2012., 2014. Food Surveys Research Group Home Page, http://www.ars.usda.gov/ba/bhnrc/fsrg.

[14] Nutritionix. Nutritionix api. http://www.nutritionix.com/.

[15] Semantics3. Semantics3 api. http://www.semantics3.com/.

[16] Mandy Korpusik, Rachael Naphtal, Nicole Schmidt, Scott Cyphers, and James Glass. Nutrition system demonstration. In *Proc.SLT*.

[17] W3C. Web speech api. https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html.

[18] M. Korpusik, N. Schmidt, J. Drexler, S. Cyphers, and J. Glass. Data collection and language understanding of food descriptions. In *Proc.SLT*, pages 560–565.

[19] Mandy Korpusik. Spoken language understanding in a nutrition dialogue system. Master's thesis, Massachusetts Institute of Technology, 2015.

[20] Amazon mechanical turk. mturk.com.

[21] Google. Freebase. http://www.freebase.com/.

[22] Evo inflector. https://github.com/atteo/evo-inflector.

[23] Damien Conway. An algorithmic approach to english pluralization. http://www.csse.monash.edu.au/ damian/papers/HTML/Plurals.html.