Human-in-the-Loop Learning for Personalized Diet Monitoring from Unstructured Mobile Data

NILOOFAR HEZARJARIBI, Washington State University, USA
SEPIDEH MAZROUEE, University of California San Diego, USA
SAIED HEMATI, University of Idaho, USA
NAOMI S. CHAYTOR, Washington State University, USA
MARTINE PERRIGUE, Washington State University, USA
HASSAN GHASEMZADEH, Washington State University, USA

Lifestyle interventions with the focus on diet are crucial in self-management and prevention of many chronic conditions such as obesity, cardiovascular disease, diabetes, and cancer. Such interventions require a diet monitoring approach to estimate overall dietary composition and energy intake. Although wearable sensors have been used to estimate eating context (e.g., food type and eating time), accurate monitoring of dietary intake has remained a challenging problem. In particular, because monitoring dietary intake is a self-administered task that involves the end-user to record or report their nutrition intake, current diet monitoring technologies are prone to measurement errors related to challenges of human memory, estimation, and bias. New approaches based on mobile devices have been proposed to facilitate the process of dietary intake recording. These technologies require individuals to use mobile devices such as smartphones to record nutrition intake by either entering text or taking images of the food. Such approaches, however, suffer from errors due to low adherence to technology adoption and time sensitivity to the dietary intake context.

In this article, we introduce *EZNutriPal*[1], an interactive diet monitoring system that operates on unstructured mobile data such as speech and free-text to facilitate dietary recording, real-time prompting, and personalized nutrition monitoring. EZNutriPal features a natural language processing unit that learns incrementally to add user-specific nutrition data and rules to the system. To prevent missing data that are required for dietary monitoring (e.g., calorie intake estimation), EZNutriPal devises an interactive operating mode that prompts the end-user to complete missing data in real-time. Additionally, we propose a combinatorial optimization approach to identify the most appropriate pairs of food names and food quantities in complex input sentences. We evaluate the performance of EZNutriPal using real data collected with 23 human subjects who participated in two user studies conducted in 13 days each. The results demonstrate that EZNutriPal achieves an accuracy of 89.7% in calorie intake estimation. We also assess the impacts of the incremental training and interactive prompting technologies on the accuracy of nutrient intake estimation and show that incremental training and interactive prompting improve the performance of diet monitoring by 49.6% and 29.1%, respectively, compared to a system without such computing units.

[1] Resources for EZnutriPal are publicly available at https://github.com/niljaribi/EZNutriPal

Authors' addresses: Niloofar Hezarjaribi, Washington State University, Electrical Engineering and Computer Science, Pullman, WA, 99163, USA; Sepideh Mazrouee, University of California San Diego, Electrical and Computer Engineering, La Jolla, CA, 92093, USA; Saied Hemati, University of Idaho, Electrical and Computer Engineering, Moscow, ID, 83843, USA; Naomi S. Chaytor, Washington State University, Elson S. Floyd College Of Medicine, Spokane, WA, 99210, USA; Martine Perrigue, Washington State University, Elson S. Floyd College Of Medicine, Spokane, WA, 99210, USA; Hassan Ghasemzadeh, Washington State University, Electrical Engineering and Computer Science, Pullman, WA, 99163, USA.

XXXX-XXXX/2020/3-ART $15.00
https://doi.org/0000001.0000001

Additional Key Words and Phrases: Diet Monitoring, Wireless Health, Mobile Computing, Human-in-the-Loop Learning, Unstructured Data, Real-Time Prompting, Combinatorial Optimization, Assignment Problem.

ACM Reference Format:
Niloofar Hezarjaribi, Sepideh Mazrouee, Saied Hemati, Naomi S. Chaytor, Martine Perrigue, and Hassan Ghasemzadeh. 2020. Human-in-the-Loop Learning for Personalized Diet Monitoring from Unstructured Mobile Data. 1, 1 (March 2020), 24 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Chronic conditions such as obesity, diabetes, cardiovascular disease, and cancer are long known as leading causes of death and major contributors to the growing healthcare costs worldwide [36]. The ability to accurately monitor food intake, measure calorie consumption, and analyze diet patterns are central to the development of clinical interventions that aid with the prevention and self-management of chronic conditions [5]. Conventional methods to estimate nutrient intake utilize self-administered questionnaires focused on food frequency, meal recalls, and food diaries [14, 45]. These methods are, however, error-prone due to challenges related to (i) short-term memory, (ii) low user adherence, and (iii) recording and estimation errors [45]. Therefore, it is imperative that we develop a novel diet assessment methodologies that are affordable, accurate, and easy-to-use.

The utility of mobile devices (e.g., smartphones) and wearables (e.g., wrist-bands, necklaces) for diet monitoring is encouraging due to the pervasive nature of these devices and their continued widespread utilization [35, 42]. Although wearable sensors have been utilized to detect contextual attributes of eating (e.g., eating time, food type, and eating style), these technologies have not advanced to the level where they can measure the dietary composition and calorie intake. Mobile devices, on the other hand, have been used to ease the process of self-administered diet monitoring for detailed dietary intake estimation such as calorie consumption measurement. This process involves recording food intake information through structured input such as structured text [15]. Current text-based mobile dietary monitoring technologies require the end-user to utilize structured text input and explicitly specify food intake information such as 'food name', 'portion size', and 'eating time'. Such a self-monitoring approach is burdensome and adherence to technology adoption has been reported to decline over time [11, 31].

To address the limitations of existing dietary self-monitoring technologies, we propose an entirely new approach in which multi-modal unstructured mobile data such as speech and free-text are used along with intelligent computational models for real-time, personalized, and accurate dietary monitoring. Our proposed system, called *EZNutriPal*, allows for easy recording of food intake, incremental learning of new foods, and dealing with complex input data collected in uncontrolled environments. EZNutriPal features a unique combination of novel human-computer-interaction, computational intelligence, and combinatorial optimization methodologies for diet monitoring in real-world settings.

Our goal is to develop a computational framework that uses user-expressed unstructured input for dietary recording, develops incrementally trained natural language processing techniques to tag nutrition data, performs an accurate mapping between food names and food-related entities, and conducts string matching for food name identification from a nutrition database (e.g. USDA nutrition database). The novel contributions offered in this article can be summarized as follows.

- *Named entity recognition:* we devise a named entity recognition (NER) approach to substantially reduce the number of queries made to the nutrition database. Prior research used a pattern matching approach where natural language processing tags were generated for every word in the expressed input and looked for an existing pattern among combinations of natural language processing tags [25]. Such an approach, although simple, results in many irrelevant queries being made to the nutrition database prior to calorie intake computation. As a result of developing a named entity recognition model in EZNutriPal, we reduce the number of database queries by 76.8% (see Section 5). The proposed named entity recognition approach directly tags the input sentence for nutrition-related data, thus leaving out irrelevant words from further processing.

- *Incremental learning:* EZNutriPal uses a human-in-the-loop approach where new input patterns are added to the training data and new classification models are automatically trained utilizing the natural language processing unit. The newly added training data are labeled by a human subject. The labeled entities are then added to the end-user's individualized nutrition dataset. Prior work assumed that the nutrition database is static in the sense that it contained a limited number of food names and associated nutrient per unit as a reference for nutrient intake computation [25]. As a result, the nutrition database could not grow dynamically to recognize new food items automatically. The new incremental learning feature in EZNutriPal allows the system to learn the user's unique diet behavior (i.e., unique food items and the way they are expressed) in a short period of time (e.g., 6 days on average).
- *Real-time prompting:* one limitation of current nutrition monitoring technologies is that they assume that the expressed input data is complete without any missing data regarding dietary intake. As an example, for an expressed food name without a portion size, current technology assumes a default portion size. Therefore, the actual dietary intake numbers could be overestimated or underestimated depending on the quality of the input data. EZNutriPal addresses this problem by prompting the end-user, in real-time, for missing data to improve the accuracy of nutrition intake computation. We show that real-time prompting improves the accuracy by 29% (see Section 5).
- *Optimization for accurate mapping:* we recognize that it is possible that complex sentences with multiple unique food items and several portion size values are expressed in an unstructured fashion. Prior research, however, assumed that a single sentence per recording session is expressed [25]. That is, in any dietary expression, only one food item with its associated food quantity is expressed by the end-user. In order to find an optimal mapping from food names to expressed food quantities, we pose an Integer Linear Programming (ILP) [6] formulation with the objective function of minimizing the mapping error as computed by the overall distance between food names and food quantities based on their location in the sentence. We then model this problem as an assignment problem and propose a combinatorial approach, based on the Hungarian algorithm, to find an optimal mapping from the food names to food quantities.

EZNutriPal is a novel dietary monitoring framework that works on unstructured mobile data (text and/or speech). An array of data processing units is proposed, including natural language processing, prompting, mapping, and string matching to extract computationally-verified dietary information from unstructured data and to compute nutrient information such as calorie intake. By constructing a named entity recognition approach and developing an incremental training mechanism, application-specific natural language processing is performed to efficiently tag nutrient data. To this end, an intent classification model is trained for classifying the sentences into nutrition-related or non-nutrition-related groups. Specifically, an entity classification model is devised to tag the input sentences with nutrition-related entities. This model is trained on a dataset that contains labeled entity training data. The classification models in EZNutriPal are trained incrementally as more training data are expressed by the end-users and annotated by a human observer. Our extensive analysis of real data demonstrates the significance and feasibility of EZNutriPal for continuous and real-time diet monitoring with an overall accuracy of 89.7% of nutrient intake estimation.

2 RELATED WORK

With a continuously growing incidence, chronic conditions such as heart disease, cancer, and diabetes remain the leading causes of death and disability [7, 48]. For example, more than 39% of adults and 13% of youth are obese globally [22]. Chronic diseases often create complex health challenges. For instance, being obese causes insulin resistance in diabetic patients and increases the risk of cardiovascular disease and cancer [33]. Yet, chronic conditions are known to be preventable and manageable through early detection and lifestyle interventions such as those focused on improving diet and physical activity.

Lifestyle interventions that focus on improving diet require an approach for dietary monitoring such as estimating calorie intake. One of the earliest methods in estimating calorie intake is self-monitoring utilizing questionnaires about food frequency, meal recalls, and food diaries [14, 45]. However, these methods are error-prone, burdensome, biased to memory, and suffer from low user adherence [45]. Calculating calorie consumption by end-users results in underestimation and/or overestimation of the dietary intake. Prior research reported that calorie intake of unhealthy foods is overestimated by 17% and that of healthy foods is underestimated by 16% [10]. Another study reported that users underestimate the nutrient intake by more than 37% [21]. Therefore, novel methodologies are warranted for efficient and accurate dietary self-monitoring [35, 42].

The utility of mobile devices and wearable sensors for health monitoring is encouraging [1, 16, 32, 40]. Mobile devices such as smartphones are personal commodities that are always carried by users. Studies show that more than 58% of the people own a smartphone [28]. As a result of these developments, patients' interest in utilizing mobile devices for health monitoring is increasing (e.g., at 62.8% and 70.6% for monitoring mental health and symptoms, respectively [46]). From a technology development standpoint, mobile devices are also advantageous due to various embedded sensors used in mobile devices and the continuous wireless connectivity that is available on mobile devices. Therefore, mobile devices offer unique features for eating behavior administration and nutrition monitoring [30].

Wearable sensors and mobile devices are widely used for physical activity monitoring based on data collected from sensors such as accelerometer, gyroscope, and piezoelectric sensor [2, 13, 20, 24, 41]. These systems are used to assess diet and stress [13, 18, 38]. However, the devices that are concerned with assessing multiple health factors may use a manual data entry process [39, 47]. Several researchers developed wearable systems for monitoring dietary context such as eating time, food type, and eating style. A system based on piezoelectric sensors embedded in a necklace was proposed for food type detection from skin motions [3]. In another work, a method was proposed for detecting and characterizing chewing bouts using piezoelectric sensors embedded in temporalis muscle [17] where the system achieved an accuracy of 96.3%. Another study proposed a bite counting method by tracking wrist motions that obtained 89% accuracy in the targeted counting tasks [44]. Although these systems collect eating-related data seamlessly, their utility is limited to portion control or eating detection rather than estimating nutrient intake.

Several researchers proposed the use of computer vision in energy intake monitoring. This method was initially proposed as a replacement method for pen/pencil in order to ease the diet monitoring for adolescents [9]. However, prior research also emphasizes the challenges associated with utilization of pen/pencil techniques in younger generations who demonstrated less desire in adopting such technologies [4]. In such systems, end-users take an image of their food. The image is then classified using machine learning techniques for food type detection. Many approaches based on food images aim to help subjects estimate the portion size associated with the food images. Previously, researchers considered the correlation between the region of food in the image and its portion as a method of estimating portion size [34]. DietCam is an automatic food classification method based on computer vision techniques. The system utilized a method called multi-view multi-kernel Support Vector Machine (SVM) [23], which required end-users to take three different images from multiple angles. In another study, the authors proposed a system consisting of a smartphone and a laser attachment [43]. This method was developed to measure daily food intake by capturing the depth of food objects in the images. Researchers proposed a framework in another research study, which utilized a reference object for estimating food volume from a single-view 2D image [12]. In another work, a method was proposed for 3D reconstruction from a 2D image taken by a wearable device [27]. A known-size plate was utilized as a necessary reference for volume measurement. In another work, a framework called mFR is proposed that allows the users to send the images of their food and beverages to a server for food/beverage type detection and volume estimation [8]. In order for this system to be useful, the images of the food/beverage should be taken before and after food consumption. Image-based nutrition monitoring is time-sensitive; it requires the end-users to record data prior to eating. Moreover, these methods lack the solution for complex unstructured food items, and they cannot assess the detailed content of the food which results in reduced accuracy. Additionally, volume estimation is a sophisticated task, due to different food shapes and appearances. Another weakness of these methods is an inconsistent image quality using different smartphones. Current research in this area requires further data collection in uncontrolled settings and with the use of mobile devices. Currently, many studies in this area used web images, which may not be a good representative of real-world scenarios. Additionally, providing visual information exclusively is not enough to recognize the food.

Another class of nutrition monitoring technologies is commercial mobile applications, which require structured user input, wherein the end-user is responsible for feeding data for each required field by the system in a structured manner [37]. Typing manually is burdensome, error-prone, and time-consuming; moreover, it reduces adherence over time. EZNutriPal is an entirely new framework for dietary monitoring from unstructured mobile data such
as speech and te
of diet monitori
that analyze sp
Google home ha
the use of voice
has a small keyb
is preferred.

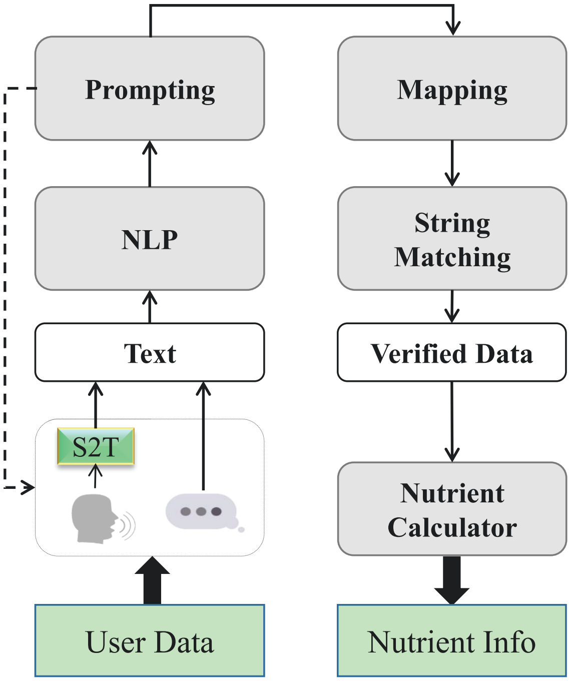


Fig. 1 High-level overview of the EZNutriPal: The system consists of data acquisition, natural language processing, prompting, mapping, string matching, nutrition database, and nutrient intake calculator units.

3 EZNUTRIPAL FRAMEWORK DESIGN

EZNutriPal aims to monitor nutrition intake from unstructured input data such as spoken language and free-format text which are common interaction modalities on mobile devices. A high-level schematic of the framework is shown in Fig 1. At the core of the system are natural language processing and string matching units that identify parts of the expressed input data that are relevant to the task of diet monitoring. However, in addition to these units, EZNutriPal devises a prompting unit and a mapping unit to enhance the quality of the gathered data. In what follows, we present an overview of the framework followed by a more detailed explanation of each unit.

3.1 Overview

As shown in Fig 1, input to our system can be in the form of unstructured data. Therefore, the end-user is not limited to a specific format for expressing their diet intake. The input is fed into a natural language processing unit, which is composed of two classification models, *intent classification* and *entity classification*, placed in series. The input text is classified into nutrition- or non-nutrition-related data using the intent classification model. The intent classification model is trained on an intent training dataset, which is extended over time as new intents are expressed by the end-user. Assuming that the input is classified as nutrition-related, the sentence is then transferred to the next module of the natural language processing unit (i.e., entity classification). The entity classifier is used to tag the sentence with nutrient-related entities such as food name, portion size, and unit. The entity classification model is trained using an entity training dataset.

During intent classification, if the input sentence is classified as a non-nutrition-related sentence, the end-user is notified about the sentence not being recognized. Thereafter, the sentence is reviewed and labeled by an expert, and subsequently added to the intent training dataset. If the entity classification model does not find the required entities, the end-user is prompted for the missing information. The end-user may provide additional data following the prompt; however, the main purpose of the prompting is to add more labeled data to the training dataset to obtain a more accurate model.

The next processing unit in EZNutriPal is referred to *mapping unit*, which is responsible for finding *food quantities* (e.g., portion size and unit) for each food name in the expressed input sentence. Prior to calculating nutrient intake, a string matching unit is utilized to locate the food name in the nutrition database. If the food name is not matched with any items in the database, an approximate matching approach is used to identify the most similar entry to the input food name in the nutrition database. Thereupon, tuples of verified nutrition data are obtained and passed on to a nutrient calculator unit for calculating nutrients of interest based on the inferred verified data.

Different nutrient values can be calculated by the nutrient calculator unit. Let $\mathcal{K} = \{K_1, \ldots, K_m\}$ be the set of nutrients of interest. Examples of such nutrients include calorie, protein, carbohydrate, and sugar. Assume that the verified data contains a set of identified food items, their portion sizes, and their expressed units. Let $\mathcal{F} = \{F_1, \ldots, F_n\}$ be the verified data extracted from the input which is fed into the nutrient calculator unit. Each entry in the verified data is a tuple $F_i = <f_i, p_i, u_i>$ where f_i, p_i, and u_i denote food name, portion size, and unit associated with i-th entry in the input data, respectively. Given a verified set of F_i entries and a nutrient of interest K_j, the nutrient calculator will first query the database to obtain the amount of nutrient value associated with each f_i and K_j. To calculate nutrient intake, a database of food names and their specified nutrients per serving need to be available. An example of this database is the United States Department of Agriculture (USDA) national database [19], which is also used for validation of our diet monitoring framework. The nutrient value, v_{ij}, obtained from this database, is the amount of nutrient per unit u_i. We can compute v_{ij} as shown in Equation 1.

$$v_{ij} = Query(f_i, K_j, u_i) \tag{1}$$

After querying the database for each f_i in the verified data, the nutrient calculator computes the total amounts of nutrient intake for the expressed sentence as in Equation 2.

$$V_j = \sum_{i=1}^{n} v_{ij} p_i \tag{2}$$

Algorithm 1 shows the nutrition monitoring procedure using EZNutriPal. In this algorithm presentation, S_i, refers to the raw sentence, which is the end-user's input in the form of either unstructured text or converted-to-text unstructured spoken language.

Algorithm 1 Nutrition Monitoring Algorithm Using EZNutriPal

```
procedure EZNutriPal-algorithm
Input: Set of raw sentences $S = \{s_1, s_2, ..., s_p\}$, Threshold, $K_j \in K$
Output: a list of different nutrient values
    for each $s_i$ in $S$ do
        [[FoodName], [PortionSize], [Unit]] ← NLP($s_i$)
        $F_i$ ← Mapping($S$, [FoodName], [PortionSize], [Unit])        ▷ Adds mapped tupples to $F_i$
        for each $F_i$ in $F$ and $K_j$ do
            [$p'_i$] ← TupleExtractor($p_i$)
            [$u'_i$] ← TupleExtractor($u_i$)
            [$f'_i$] ← StringMatching($f_i$, Threshold)
            <$f_i$,$p_i$,$u_i$> ← <$f'_i$,$p'_i$,$u'_i$>
            $v_{ij}$ ← Query($f_i$,$K_j$, $u_i$)
        $V_i = \sum_{i=0}^{n} v_{ij}$        ▷ Nutrient value associated with sentence $S_i$
```

An example of how EZNutriPal process input data for calorie intake estimation is shown in Fig 2. The raw sentence (i.e., "I had cup of mac and cheese and 10 almonds for lunch") is sent to the natural language processing unit. This unit first classifies the sentence into nutrition or non-nutrition related data using an intent classification model. In this example the sentence is classified as nutrition-related; therefore, the sentence is passed on to the entity classification model for tagging purposes. The entity classification model is responsible for tagging the sentence into nutrition specific data (i.e., food name, portion size, and unit) using a rule-based model trained on entity training data which is categorized to a specific number of entities. After tagging, 'cup', 'mac and cheese', '10', and 'Almonds' are labeled/tagged as 'Unit', 'FoodName', 'PortionSize', and 'FoodName', respectively. Among all the nutrient specific data specified in the entity training data, some of them are defined as required information. Every time the sentence is tagged, the prompting unit examines the sentence for the required tags. If one or more of the required nutrition tags are missing in the sentence, EZNutriPal prompts the end-user for the missing information. In this example portion size for "mac and cheese" is missing; therefore, prompting unit generates a message, "How much did you eat?", and sends it back to the data acquisition unit, waiting for the end-user to respond. New tags are added to the training data and the entity classifier is retrained. If the end-user does not respond to the prompt, a default value is assigned to the corresponding entity. Thereafter, the data is sent to the mapping unit to extract the verified data. Each food name in the sentence is mapped onto food quantities associated with that. In this example, the portion size associated with 'mac and cheese' is missing; therefore, a default value is mapped to the food name. Moreover, the unit of measurement is missing for 'almonds', the mapping unit assigns a default value for that. The result of mapping is two tuples of data <'mac and cheese', 'defaultVal', 'cup'> and <'almonds', '10', 'defaultVal'>. The string matching unit locates the food names in the

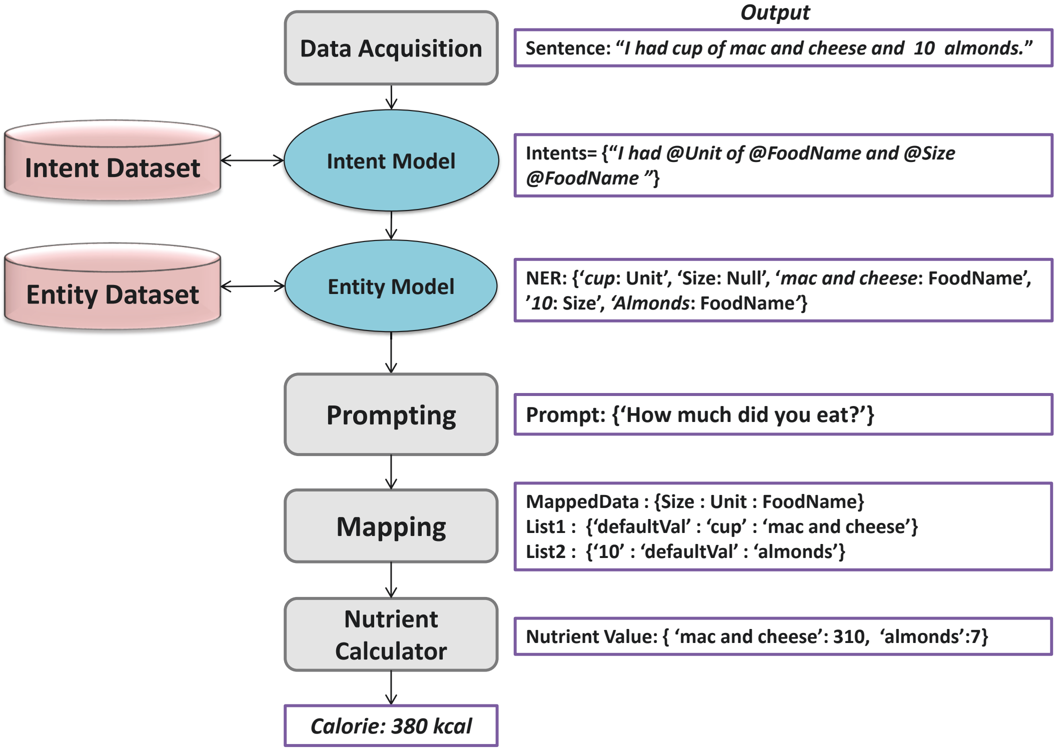

Output
Data Acquisition
Sentence: "I had cup of mac and cheese and 10 almonds."
Intent Dataset
Intent Model
Intents= {"I had @Unit of @FoodName and @Size @FoodName "}
Entity Dataset
Entity Model
NER: {'cup: Unit', 'Size: Null', 'mac and cheese: FoodName', '10: Size', 'Almonds: FoodName'}
Prompting
Prompt: {'How much did you eat?'}
Mapping
MappedData : {Size : Unit : FoodName}
List1 : {'defaultVal' : 'cup' : 'mac and cheese'}
List2 : {'10' : 'defaultVal' : 'almonds'}
Nutrient Calculator
Nutrient Value: { 'mac and cheese': 310, 'almonds':7}
Calorie: 380 kcal

Fig. 2 (1) input is text/audio; EZNutiPal outputs a string; (2) intent classification model is trained using intent training data and classifies each input sentence to nutrition or non-nutrition specific class; (3) entity classification model tags the sentence into nutrition specific words; (4) prompting module prompts for unrecognized sentences or missing information in those sentences; (5) mapping unit maps the food names to their associated nutrient specific data; and (6) nutrient calculator unit calculates the nutrient value.

nutrition database and obtains the corresponding nutrient values from the database. In this example, the nutrient of interest is calorie intake; therefore, the calorie values for one cup of 'mac and cheese' and one'almond' are 310 and 7 kCal, respectively. Finally, the nutrient intake calculator computes the nutrient values based on the expressed food items in the input sentence. In this example, the calorie value calculated by nutrient intake calculator is 390 kCal.

3.2 Natural Language Processing Unit

Natural language processing is the ability of a machine to understand humans' pronouncement as it is spoken in order to provide responses to them. Conventional natural language processing aims to tag individual words in a given sentence. This approach, however, is unable to extract structured information from unstructured text. In this work, our effort is focused on entity-relationship models that allow EZNutriPal to understand complex user-expressed intents and therefore infer recognize various nutrition-related data in the input sentence. For this purpose, specific types of nutrition-related data are searched in the input text, including the order of information, the location of each nutrient data in the text, and the previous tags. A recognition model is trained using this information for future predictions.

A typical processing pipeline for named entity recognition is shown in Fig 3. In our work, input to the named entity recognition is the sentence input to the EZNutriPal by the end-user. The output of this pipeline includes two m ed for taggin

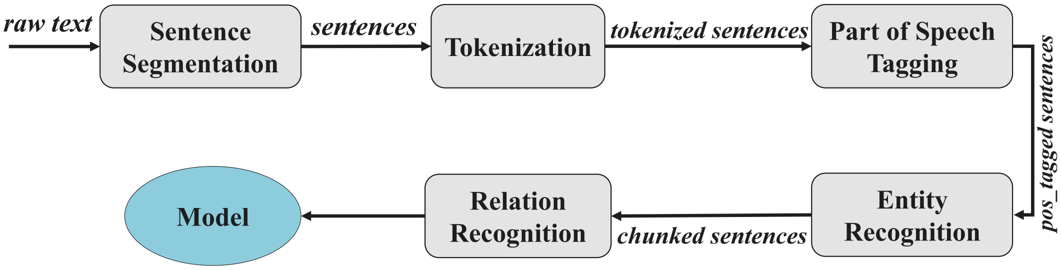


Fig. 3 Processing flow for named entity recognition.

The details of the natural language processing unit devised in EZNutriPal is illustrated in Fig 4. The natural language processing unit contains 4 main modules including intent training data, intent classification model, entity training data, and entity classification model. The input to EZNutriPal is classified using a rule-based model trained using the existing intent training data. This training dataset of intents grows over time as new intent data are added to the framework; further, the intent classification model is continuously retrained using the extended training data. If the data is classified as a nutrition-related intent, the entity classification model tries to tag the user's input data based on the predefined nutrition-related entities. The un-tagged nutrition-related data are added to the entity training data and used to retrain the entity classification model. This incremental learning a ata b

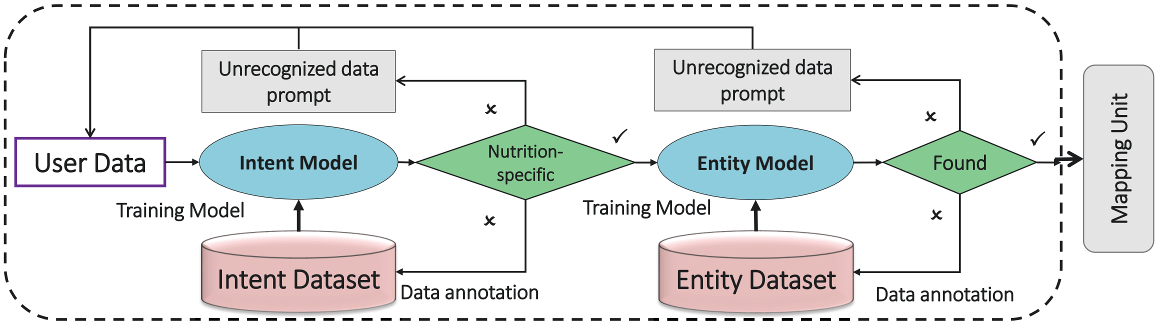


Fig. 4 Natural Language Processing unit in the EZNutriPal framework

In what follows, each module within the natural language processing unit is explained in more detail. The intent classification model is executed each time the end-user enters data into the system. This module is trained using the available intent training data. Thereafter, the trained model is utilized to classify the user data as nutrition-related or non-nutrition-related intents. Upon classification of the input as a nutrition-related intent, the entity classification model is called. However, if the input data is not classified as being nutrition-related, the end-user is notified about their input being not recognized for nutrition monitoring. The recorded data, however, is reviewed by a human expert, annotated, and added to the intent training data. The intent training dataset is initially filled with the most likely intents utilized by the end-user to express their nutrition intake. As EZNutriPal continues collecting data from the end-user, the newly annotated intents are added to the intent training dataset, which triggers retraining of the intent classification model.

The entity training dataset contains nutrient-specific data such as food-name, portion-size, and unit of measurement. The entity classification model tags end-user's input sentence based on existing entities in the entity training dataset. As an example, when the entity classification model is applied to the sentence "I had a cup of milk for breakfast", the model will assign tags 'PortionSize', 'Unit', and 'FoodName' to the words 'a', 'cup', and 'milk', respectively. Moreover, complex food names such as '2% milk', 'pepperoni pizza', and 'fuji apple', which exist in the entity training dataset, can be located in the input sentence using the entity classification model. The different representations of the unit of measurements are added to entity training dataset as well. For example, the word glass in "one glass of milk" is mapped to cup in the dataset.

We recognize that it is possible that some of the entities might be missing in a given input sentence. In such a case, the end-user is prompted in real-time about the missing entity/tag (i.e., food name, portion size, and unit). For example, entering the sentence "I had bread for breakfast" will result in the system prompting the
en m
th on
m to
en

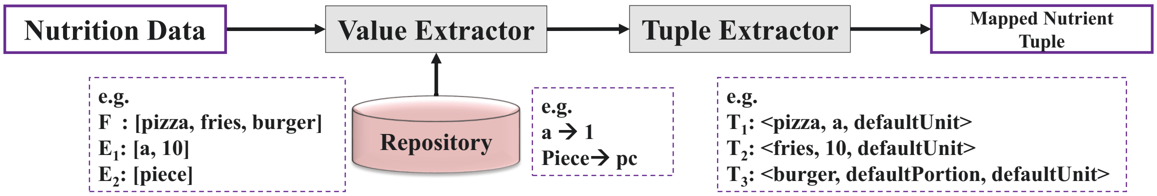


Fig. 5 Mapping unit, which performs two tasks: 1)Entity→Value, which maps each nutrient data to their associated values using a repository; 2) Tuple Extractor links nutrition specific data to their associated food names using mapping algorithm. E_1 and E_2 are portion size and unit, respectively.

3.3 Mapping Unit

After processing the data through the natural language processing unit, the raw sentence and tagged nutrient-related data are passed onto the mapping unit. Nutrition-related data (FoodNames, PortionSizes, Units) are separately identified in the input sentence; therefore, the mapping unit is responsible for determining association among the data items. This mapping unit is composed of two modules including *value extractor* and *tuple extractor*. Fig 5 shows a block diagram of the unit.

3.3.1 Value Extractor module. The tagged nutrient-related data contains words with different nutrient-specific tags (i.e., food-name, portion-size, and unit). The tagged data might have words that cannot be used by the framework for extracting nutrient-data from the nutrition database. For example 'a', 'half', and 'tablespoon' are not recognizable by the nutrition database; therefore, value extractor module utilizes a data repository to map unrecognized data to an understandable word or number by the nutrition database. The output of the value extractor module for the above-mentioned example is '1', '0.5', and 'tbsp', associated with 'a', 'half', and 'tablespoon', respectively.

3.3.2 Tuple Extractor module. We recognize that the end-user may report food intake without specifying all the food quantities corresponding to the expressed food names. Consider, as an example, the input sentence "I had cheeseburger (380 cal) with 5 chicken nuggets (44.4 cal) and ice cream (259 cal) and 10 almonds (1 cal).". Mapping food names to the quantities based on the order in which they appear in the input sentence may result in inaccurately calculating nutrient intake values. For the considered example, a mapping approach based on the ordering of the nutrient-related data will result in the following tuples, <'cheeseburger','5', 'defaultValue'>, <'chicken nuggets','10', 'defaultValue'>, <'ice cream','defaultValue', 'defaultValue'>, and <'almond', 'defaultValue', 'defaultValue'>, which leads to an estimated calorie intake of 2604 kCal while the actual value is only 871 kCal.

We formulate the problem of tuple extraction as an integer linear programming problem and propose a heuristic to solve the mapping problem.

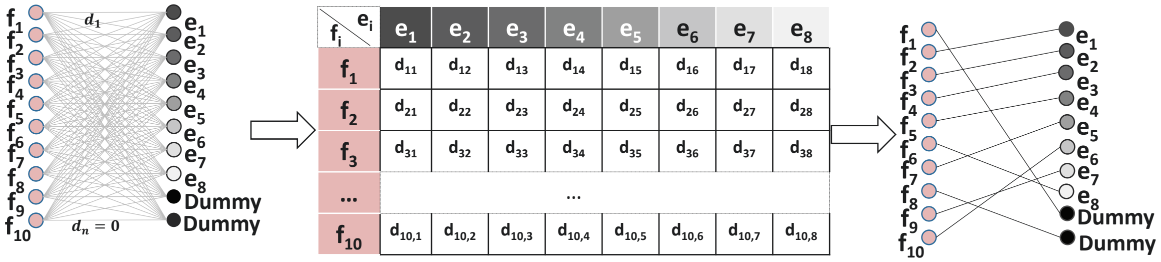
Problem Formulation. EZNutriPal performs the mapping between food names and their associated food quantities based on the relative distance between each food name and food quantity in the sentence. It is quite likely that the number of food quantities is less than the number of food names due to incomplete information provided by the end-user or because of error the processing pipeline in the system; therefore, we enhance the efficiency of our approach by breaking the solution into two steps: *nutrient data partitioning* and *food quantity labeling*. The partitioning step aims to find the relation between food names and their associated food quantities. If the constraints of the problem meet, we assign exactly one tagged word from each food quantity (entity) to each food name to minimize the total distance between the mapped food name and the food quantity.

There can be k food quantities associated with each food item. Assume that the partitioning algorithm groups the nutrient data gathered from the user's input into two partitions. The first partition is a list of n food items tagged in the sentence. The second partition has m tagged food quantities corresponding to the k_{th} $entity$ (food quantity) (e.g. m portion sizes in the sentence) such that $n \geq m$. We calculate the distance between the i_{th} food item f_i and j_{th} element of the k_{th} $entity$, e_{kj}. We find an optimal entity for each food name such that the sum of distances between food items and their associated entities are minimized. An Integer Linear Programming (ILP) formulation of this optimization problem is as follows.

$$\begin{aligned}
&\textit{Minimize} && \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}x_{ij} && (3)\\
&\text{Subject to:} && d_{ij} = L(e_{kj}) - L(f_i) && (4)\\
& && d_{(i-1)j}x_{ij} \geq 0 \qquad , \quad \forall i,j && (5)\\
& && \sum_{i=1}^{n} x_{ij} \leq 1 \qquad , \quad \forall j && (6)\\
& && x_{ij} \in \{0,1\} \qquad , \quad \forall i,j && (7)
\end{aligned}$$

where $L(e_{kj})$ and $L(f_i)$ denote the locations of e_{kj} and f_i in the sentence, respectively. The measure d_{ij} represents the distance between e_{kj} and f_i. We note that it is more probable for the food quantities to precede the food name in the English language. Constraint 4 guarantees that food names appear after their corresponding food quantities. Constraint 5 is employed to guarantee that among all food quantities of each type at most one is assigned to each food item.

Graph Modeling. The nature of the problem is an assignment problem; therefore, it can be modeled using a bipartite graph and solved as a matching problem. The bipartite graph is constructed based on the partitioning of the food names and food quantities to represent the two shores of the graph. We note that this graph will not necessarily be a complete bipartite graph.



f_i \ e_i	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8
f_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}
f_2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	d_{27}	d_{28}
f_3	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{36}	d_{37}	d_{38}
...	...							
f_{10}	$d_{10,1}$	$d_{10,2}$	$d_{10,3}$	$d_{10,4}$	$d_{10,5}$	$d_{10,6}$	$d_{10,7}$	$d_{10,8}$

Fig. 6 An example of a weighted bipartite graph.

Fig 6 shows the complete bipartite graph, the table, and final bipartite graph for a sentence with 10 food items and 8 entities. Assume that we have an n×m table X filled based on the problem constraints discussed previously. If e_{kj} is located after the previous food name f_{i-1} ($L(e_{kj}) > L(f_{i-1})$), the variable x_{ij} is set to '1'. It is set to '0' otherwise. For each x_{ij}=1 in the table, we draw an edge in the graph. Each entity can be assigned to only one food name because food names are expected to have one associated tagged nutrient data of each specific entity. This simple procedure will give us the bipartite graph that will be used for food-to-quantity mapping.

Problem Solution. The optimization problem for mapping is the problem of finding d_{ij} values that minimize the total distance from f_i to all the e_{kj} while meeting the problem constraints. This problem is in the form of a Linear Assignment Problem and can be solved using *Hungarian* method [29].

We propose a simple polynomial time algorithm, based on the *Hungarian* method [29], to solve the mapping problem. The algorithm is shown in Algorithm 2. Our goal is to assign at most one entity to each food name such that the total distances of assignments are minimized. This problem can be viewed as a weighted matching problem. Since G(V,E,W) is a bipartite graph where |F| ≥ |E|, we find a perfect matching in G to minimize the assignment costs. The Hungarian approach utilized in this work runs in O(n^3) where $n = max(|F|\ , |E|) = |F|$.

Algorithm 2 Hungarian algorithm used for simple mapping in EZNutriPal

```
procedure HUNGARIAN-ALGORITHM
Input: food names $f = \{f_1, f_2, ..., f_n\}$, $k$ entities $e = \{[e_{11},..., e_{1m}],...,[e_{k1},..., e_{km}]\}$
Output: MappedTupple $<f_i, e_{1j}, ..., e_{kj}>$
    for each $e_k$ in $e$ do
        if len($f$)==len($e_k$) then
            $f$ and $e_k$ are mapped based on the order
        else
            Build an $n \times m$ matrix $X$ with elements $d_{ij}$ as the distance between $f_i$ and $e_{kj}$
            while not optimal do
                for each row of the matrix do
                    Subtract the smallest element of the row from each element of that row
                for each col of the matrix do
                    Subtract the smallest element of the col from each element of that col
                Utilizing the minimum number of lines cover the zeroes in the matrix
                if minimum number of lines is n then                    ▷ test optimality
                    break
                Determine the smallest uncovered entry. Subtract this entry from each uncovered row, and
    then add it to each covered column.
```

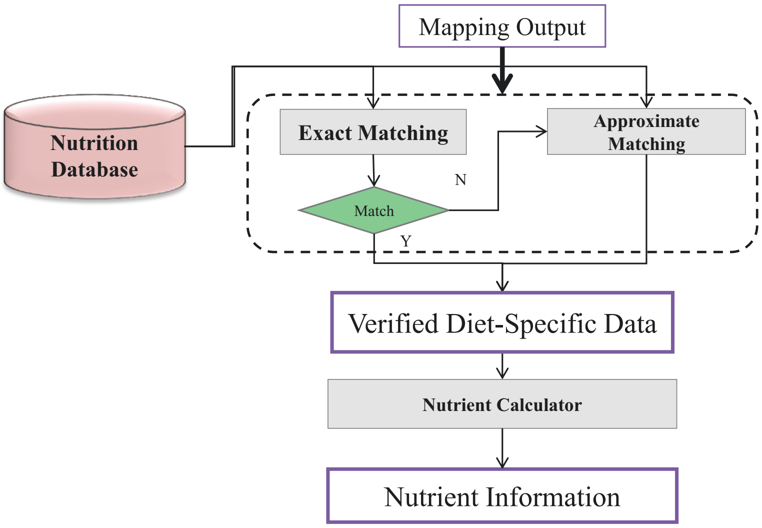


Fig. 7 The string matching unit searches for the expressed food names and their associated values in the nutrition database.

A list of mapped nutrient-specific data produced by the mapping unit is passed to the string matching unit. This unit contains two modules including *exact matching* and *approximate matching*. A schematic of this unit is shown in Fig 7.

The system aims to search for each food name in the nutrition database. In case the system was not able to locate the exact food name in the first step, an approximate matching approach is utilized. This is a technique for finding a match between two strings approximately. At first, the algorithm aims to narrow down the search space in the database and generate a list of most probable matches for the desired food name. In order to accomplish this goal, the likelihood of each string in the database matching the detected food name in the sentence is calculated. This probability is later compared with a predefined threshold. Thereafter, the strings with the calculated matching probabilities less than the threshold are eliminated from the match list. In the second step, the most similar food name is determined by running edit distance algorithm on the match list obtained from the first step.

4 VALIDATION STRATEGY

In this section, we describe our approach for validating EZNutriPal including implementation details and experimental settings. To evaluate the system, each unit was experimented in a real-world scenario. The natural language processing unit, as well as the prompting unit need to be evaluated for efficient learning. To this end, two types of experiments were conducted involving 13 participants with ages in the range of 18-30, 4 females and 9 males, at Washington State University. The participants were recruited in three cohorts to allow us to assess convergence of the learning algorithms both as more data are collected with one participant and as more participants are added to the system. Therefore, the rate of convergence of the error to zero in the classification models was measured in each phase of the experiment each of which involved one cohort of the participants. The experiments also allowed us to monitor how the user's diet behavior stabilizes over time. The data collected in these experiments were also used to evaluate the impact of the training unit and prompting unit on the overall performance of the system as measured by the accuracy of the algorithms and estimated calorie intake. We obtained Institutional Review Board (IRB) approval for performing these experiments. The experimental design is further discussed in Section 4.2.

4.1 System Setup

We utilized Google's open source speech recognition API (Application Programming Interface) on Android platform for speech-to-text conversion. After the end-user taps the speak button, the app streams the audio to the server and receives the response from the API using an intent called RecognizerIntent. This intent is for supporting speech-to-text conversion through an intent in Android called "intent.Android.speech.RecognizerIntent". The preferred language and speech model is set using a variable called "Android.speech.RecognizerIntent.EXTRA_LANGUAGE_MODE".

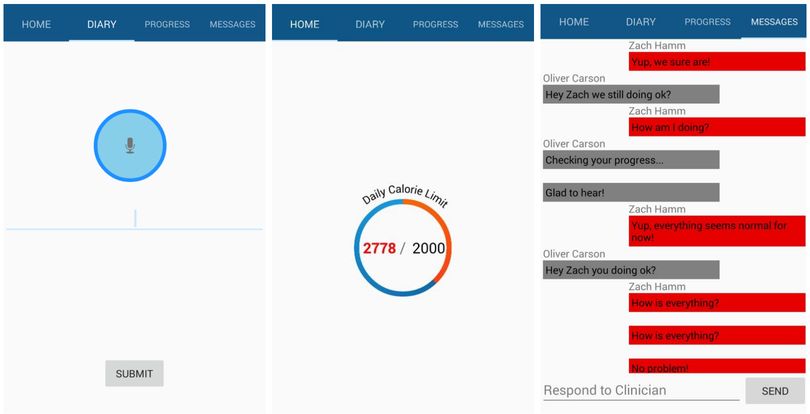
To train the natural language processing algorithm, we utilized an existing rule-based model for designing the conversational user interface. We gathered intent training data to train the rule-based model, understand the intent, and meaning of an unstructured input sentence. After understanding the input, this model outputs a proper response. The intent training data initially contained one simple sentence expressing the nutrition intake (I had *PortionSize Unit* of *FoodName*). The frequent use of EZNutriPal by the end-user resulted in feeding more data to the training set. The newly added intent training data were annotated continuously (e.g., on a daily basis) by a human experimenter and a new model was automatically trained using the data. Similarly, the entity training data were gathered and prepared to learn an entity classifier. The newly tagged words were added to their associated entity in the training data by a human experimenter. The data fields associated with food names were added to the nutrition database manually. The annotations given by the models were checked and revised by the human experimenter daily.

The mapping, string matching, and nutrient calculator units were implemented in Python. In order to perform approximate matching, we utilized the Levenshtein algorithm embedded in the pylev library. The nutrition dataset was extracted from the USDA database and embedded in python dictionary to allow fast lookup using hash tables with time complexity of $O(1)$.

4.2 Experimental Data

Our goal was to evaluate the performance of EZNutriPal and to assess the impacts of utilizing named entity recognition, training models, and prompting unit on the accuracy of the system. We implemented a mobile app using Android Studio and conducted the experiments using the implemented software. A screen-shot of the mobile app is shown in Fig 8. In the diary page, the end-user could report nutrition intake and transfer it to the server. In the home page, the amount of calorie consumption in a day is shown and a red line indicates that the end-user has exceeded the calorie limit. The last page shows the interface for communication between clinician and end-user.

We evaluated the system with 13 participants including 5 female and 8 male subjects with ages between 18 and 35. The study was divided into three phases where each phase lasted for two weeks. One purpose of these experiments was to assess how fast the error of the learning algorithms converged to 'zero' as new participants are added to the system; therefore, our data collection initially involved only one participant in Phase (I). In the next phase, Phase (II), we recruited two new participants to use the system simultaneously. Finally, we included 10 participants in Phase (III). The participants were asked to record their dietary intake in the form of unstructured input on a daily basis during the two-week experiment. The participants were asked to express their dietary intake as text or speech for the duration of the data collection. The collected data revealed that the participants mostly expressed their eating in three ways including complete sentences (e.g., *I ate a piece of apple for breakfast.*), describing a scenario (e.g., *I went to McDonald and ordered a large meal of big mac.*), and a brief expression (e.g., *1 apple*).



Zach Hamm
Yup, we sure are!
Oliver Carson
Hey Zach we still doing ok?
Zach Hamm
How am I doing?
Oliver Carson
Checking your progress...
Glad to hear!
Zach Hamm
Yup, everything seems normal for now!
Oliver Carson
Hey Zach you doing ok?
Zach Hamm
How is everything?
How is everything?
No problem!
Respond to Clinician

Fig. 8 An sample screen shot of the mobile app.

The data sent to the server were used for further examination and analysis that took place daily. For ground-truth labeling, the participants were asked to specify the word labels in each sentence. The participants labeled their sentences to nutrient-specific tags (i.e., food name, size, and unit). The data were fed into the system and tagged using the learned model. The number of correctly tagged data items enumerated by comparing the tags to the ground truth labels. The mistakes were corrected, and unrecognized intents and entities were added to the *intent training dataset* and *entity training dataset*. This allowed the system to retrain new intent and entity classification models using the newly annotated data. Moreover, the number of incorrect prompts was enumerated. The ground-truth labeling for enumerating incorrect prompts were based on the labels provided by the end-users for each word. For instance, if the end-user tagged e_1 in the sentence and the prompting unit still asked for e_1 to be labeled, this entry was considered as error. As another example, if the end-user did not specify a tag for e_1 and the prompting unit failed to initiate a prompt e_1, we counted this as an error.

To compute a nutrient intake, the system searches for the food name in the nutrition database. For each possible unit of measurement, a column exists in the nutrition database (e.g. lb, oz, gr, tbsp, tsp, etc.). The system calculated the nutrient intake based on the obtained data (nutrient value) from the proper entry (foodName) and column (unit) in the nutrition database.

5 RESULTS

In this section, we present results for each experiment conducted in our study.

5.1 Accuracy of EZNutriPal

As stated previously, we conducted a six-week long study in three phases involving one, two, and ten participants for a total of 13 participants. Each phase started with a basic model utilizing one intent in the *intent training data*. The data were examined and annotated, new sentences were added to the intent training dataset, and a new model was trained on the entire dataset, in a daily basis. Therefore, by feeding new data and correcting the prompts and tags, the accuracy of the model improved over time. The number of input instances (i.e., number of expressed sentences) during the experiments are shown in Table 1. The average number of sentences in phase I, II, and III was 14.5, 24.5, and 70.8, respectively.

After entering the data into the system, some of the sentences were not recognized by the intent classification model. Moreover, some of the words were not correctly tagged (e.g., for input sentence “I had a cup of black

Table 1 Number of instances, each row corresponds to a day, and each column is associated with one phase of the experiment

Instances			
Day ♯	**Phase I**	**Phase II**	**Phase III**
day 1	13	24	73
day 2	17	28	80
day 3	12	33	77
day 4	19	21	78
day 5	11	33	83
day 6	14	20	83
day 7	8	23	65
day 8	19	25	69
day 9	14	20	72
day 10	18	21	60
day 11	16	26	57
day 12	16	23	61
day 13	11	22	62
AVG.	14.5	24.5	70.8

bean for breakfast.", the system generated these tags: 'black':'sys.color', 'bean':'foodName'). The total number of unrecognized and mis-tagged words, shown in Fig 9, can be viewed as the expected error of the combined intent and entity classification models. In phase I of the experiment, the expected error was initially 76.9%. After 5 days of data collection and algorithm training, the error converged to zero. At the beginning of phase II, the error of the system was 29.2% and it reached to zero after 3 days. In phase III of the experiment, the error was initially 16.4% and converged to zero on the second day of the experiment.

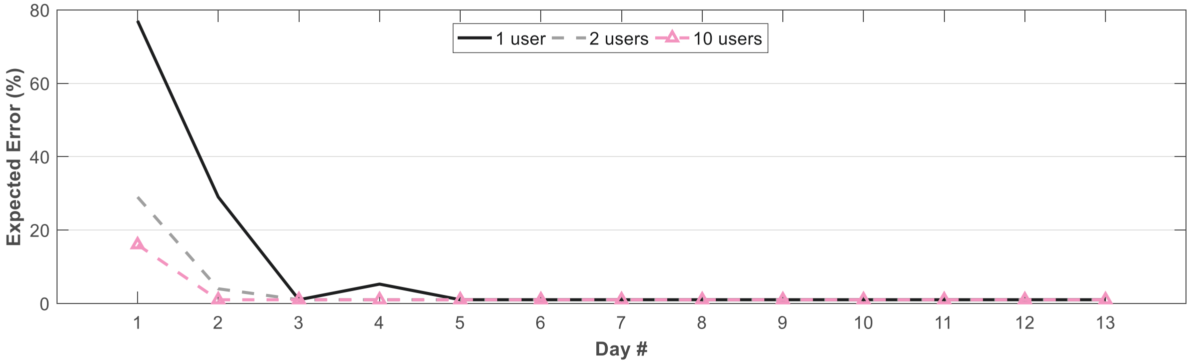
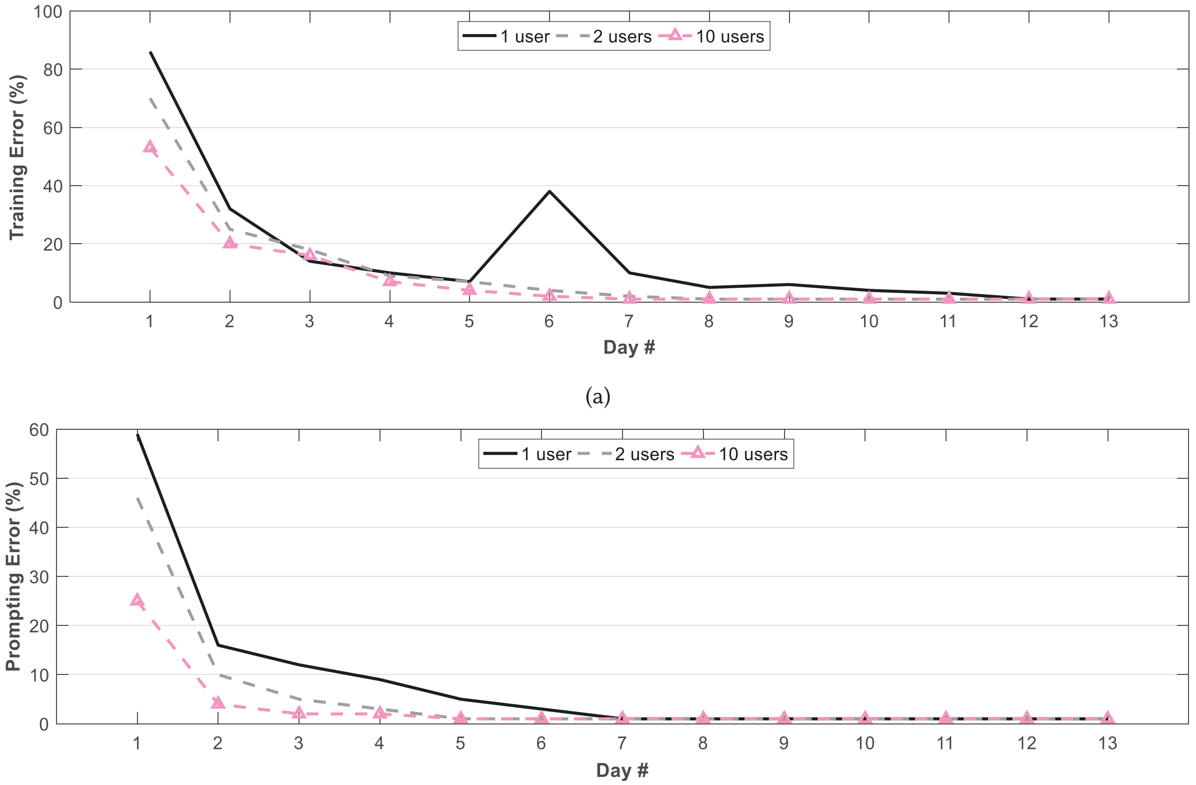


Fig. 9 Expected error of the system for each phase of the experiment.

We were also interested in examining the amount of time that it takes for the system to learn dietary habit of the end-user. We hypothesized that the dietary habit of an individual can be entirely learned within few days. We observed changes in combined error of the intent and entity classification models calculated using Equation 8, for each phase of the experiment. Moreover, the changes in error of the prompting unit for each phase were monitored. This error rate was calculated using Equation 9.



(a)

(b)

Fig. 10 Errors of (a) trained model; and (b) prompting unit.

$$E_{Training} = (\frac{| \overline{NS_{tags}} |}{| NS_{tags} |}) \times 100 \tag{8}$$

$$E_{Prompting} = (\frac{| \overline{NS_{prompts}} |}{| NS_{tags} |}) \times 100 \tag{9}$$

The term $| NS_{tags} |$ represents the total number of nutrient-related tags in the input data. The terms $|\overline{NS}_{tags}|$ and $|\overline{NS}_{prompts}|$ denote the total number of incorrectly detected nutrient tags and prompts using the natural language processing, respectively .

The results of this experiment are shown in Fig 10. The results demonstrated that the accuracy of the classification models and prompting unit improves as more participants are involved in using the system. The error of the classification models was initially 86%, 70%, and 53% for phase I, phase II, and phase III of the experiment, respectively. The error converged to zero after 12, 8, and 6 days for phase I, phase II, and phase III, respectively. On day 6 of phase I, the participant consumed only one food item which did not exist in the database; therefore, the model only tagged the size correctly. This explains the sudden change in accuracy of the models. Moreover, the error of the prompting unit started at 59%, 46%, and 29% for phase I, phase II, and phase III of the experiment, respectively.

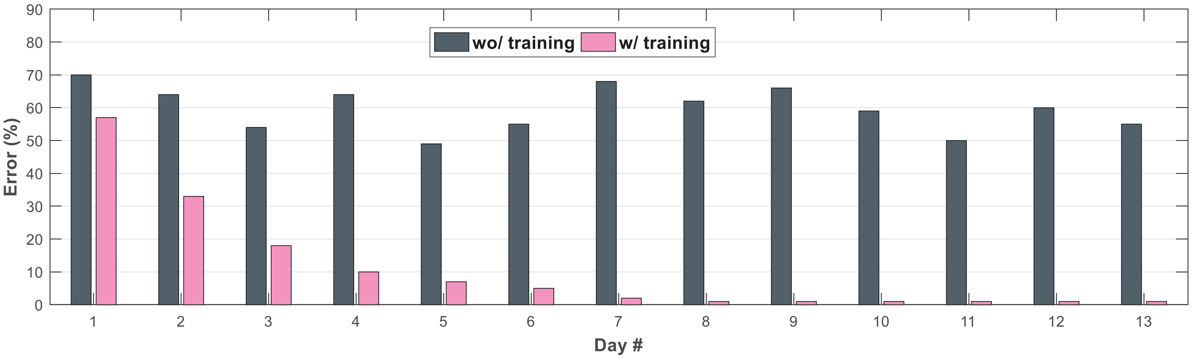


Fig. 11 The error of the system with and without training block.

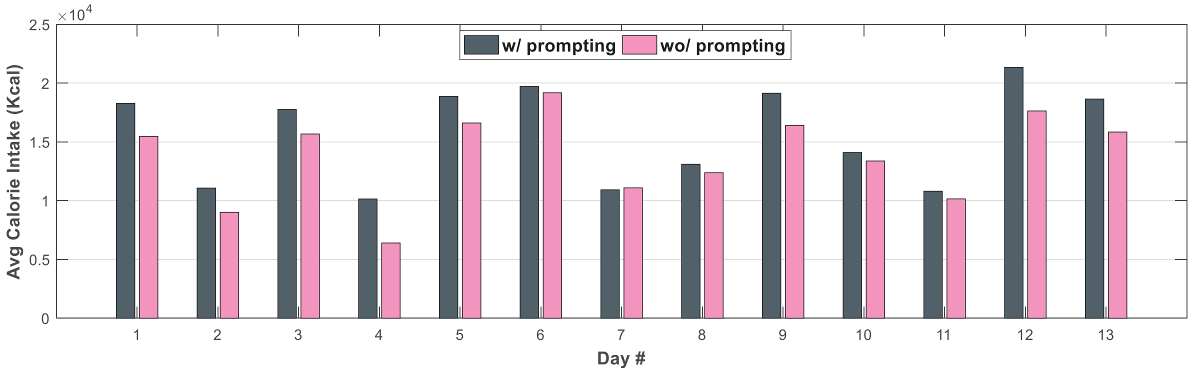


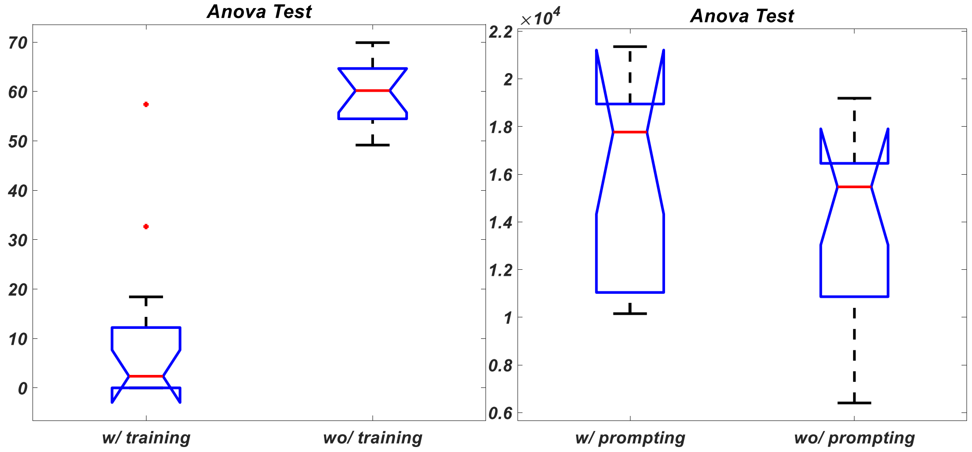
Fig. 12 The average calorie intake with and without prompting unit.

5.2 Impacts of Incremental Learning

We evaluated the impacts of the classification models on the performance of the system. Two instantiations of the system were considered for this experiment: (1) the system with training module; and (2) the system without training module. The data were entered into both systems on a daily-basis. The system was evaluated in 13 days with all the 10 participants. The results are shown in Fig 11. The average error of the system with and without use of the training models was 10.3% and 59.7%, respectively.

We evaluated the impacts of the prompting unit on the performance of EZNutriPal as well. After classifying the user's input using intent classification and tagging the nutrition-specific data utilizing entity classification, the system examines the existence of required nutrition-specific data in the input sentence. If the participant did not provide the required information, a prompt was generated.

In order to evaluate the impacts of the prompting unit on the performance of the system, data were collected from 10 participants in 13 days. It was assumed that the food quantities were not expressed by the participants



(a)

(b)

Fig. 13 The ANOVA test results for (a) impacts of training unit; and (b) impacts of prompting unit.

and default values were assigned to food quantities. The missing data results in errors in nutrient calculation. The nutrient intake was calculated with and without the prompting unit. The nutrient value associated with the user's input using the prompting unit was considered as the actual value. The error of the system without using the prompting unit was 29.1%. The results are shown in Fig 12.

5.3 Statistical Analysis

We utilized several statistical tests to assess if the impacts of the prompting and training units on the accuracy of the system are statistically significant. First, we performed a t-test to assess whether there is a statistically significant difference between the framework with and that without these units. Second, systematic difference between two instances of the framework was assessed using a one-way Analysis of Variance (ANOVA) test. The goal of this ANOVA test was to identify any significant differences between the two instances of the framework.

The t-test analysis revealed that there is a significant difference between the framework with and without the training unit (p=9.06 × 10^{-9}). However, t-test showed that the superiority of the framework with prompting over the one without prompting is not statistically significant (p = 0.2276), thus suggesting the need for a larger-scale study. Fig 13 illustrates the results of the ANOVA tests (95% CI) on the data. The results suggest that the framework with learning capabilities had a significant difference from the one without the training unit at a 0.95 significance level (p=9.06 × 10^{-9}). However, the test showed no significant difference between the framework with and that without prompting (p=0.2276). Yet, the system with the prompting unit performed better.

5.4 Comparison

In our previous work [24], Speech2Health utilized a pattern matching technique for extracting nutrient-related data from input sentence. In this article, we compare the pattern matching technique with the proposed named entity recognition approach in EZNutriPal. For evaluating Speech2Health framework in our prior work, a script containing sentences about 36 food names was provided to the user. We utilized this script to enumerate the number of queries made to the database by both frameworks, Speech2Health and EZNutriPal. Table 2 shows the number of queries for each framework versus the actual number of food items in each sentence. The results demonstrate that Speech2Health queried the database 3.4 times more than the actual number of food names, while EZNutriPal queried the database 0.8 times less than the actual number of food items. We also utilized the data from phase III of the experiment and enumerated the number of queries to the database by Speech2Health and EZNutriPak. For the total of 728 food items, Speech2Health and EZNutriPal queried the database 2824 times and 655 times, respectively.

Table 2 Number of queries to the nutrition database using pattern matching.

sentence♯	Speech2Health	EZNutriPal	NumOfFoods
1	14	3	3
2	14	4	5
3	12	1	2
4	15	5	6
5	8	1	2
6	8	2	2
7	10	3	3
8	10	4	4
9	15	3	4
10	10	2	3
11	6	2	2
$total$	122	30	36

We compared the mapping approach utilized in Speech2Health [24] with the mapping approach in EZNutriPal. The mapping approach in Speech2Health works based on the order of the food names and food quantities in the input sentence. In EZNutriPal, we hypothesized that the mapping problem can be modeled as a linear assignment problem and tried to solve this problem using Hungarian method as discussed previously in this article. Table 3 shows the difference in the calorie calculations between the two mapping methods utilized in Speech2Health and EZNutriPal using three examples.

Table 3 Comparison of calorie-intake computation based on the techniques used in Speech2Health and EZNutriPal.

sentence	Speech2Health(kCal)	EZNutriPal(kCal)
I had a cup of mac and cheese, cheeseburger, and 10 almonds for lunch	2604	871
I had whole grain bread with 2 tbsp of cream cheese and 3 tbsp of honey	434	302
I ate egg, 10 mushrooms, 0.5 naan, and 3 tomatoes	1377	292

6 DISCUSSION AND FUTURE WORK

Including human factors in modeling may lead to inaccurate outcomes. A deterministic model, thus, is likely to be less effective for a stochastic event. In the case of nutrition monitoring, several events can be considered probabilistic rather than deterministic. For instance, a slice of pizza comes with different sizes and calories. The participant may forget or over/under-estimate the food portion and other related data. A probabilistic model will provide a distribution of possible outcomes. In our future work, a probabilistic model will be developed for predicting missing information in the input data and mapping the data to their associated food items.

Part of our future work involves development of a knowledge-base for nutrition monitoring and diet planning. Specifically, we plan on devising a nutrition ontology that leads to the construction of a hierarchical knowledge-base for our system. Such a knowledge-base can be utilized to optimize the prompting approach on-the-fly based on the user's expressed information about their food intake. As a result, this will not only improve the accuracy of the prompting unit but also minimize the amount of burden on the user. We recognize that nutrition databases often include food items with different levels of complexities (e.g., pepperoni pizza, veggie pizza, etc.). Using solely the user's expressed sentence, details about dietary intake will likely be overlooked (e.g., "I had a piece of pizza" can potentially follow prompts such as "what toppings?", "what kind of crust?", "what kind of sauce?", etc.). With a knowledge-base approach, we can obtain additional details about user's food intake where the amount of details can be application-dependent. One approach to develop a hierarchical knowledge-base is to construct a tree-like structure in which each level of the tree maintains details about a food category (e.g., first level: pizza, second level: crust types, third level: sauces, etc.). Additionally, a hierarchical knowledge-base results in more accurate nutrient-intake calculations because the system interacts with the user to understand dietary intake details with a higher precision. The knowledge-base can be also used to disease-specific or population-specific interventions. Furthermore, a hierarchical knowledge-base can be used for diet planning. In diet planning, the goal will be to provide diet recommendations based on user's diet history and a targeted diet goal. A hierarchical knowledge-base can be potentially used to focus on a subset of the entire knowledge-base (e.g., a sub-tree) to make diet recommendations that are specific to the user and potentially improve adherence to diet recommendations.

Our ongoing evaluation of EZNutriPal involves conducting user studies using our implemented system. Our goals are to conduct a rigorous evaluation of the system, measure the impacts of the technology on health outcomes, and assess usability and acceptability of the technology. We are also working on developing a diet planner for guiding the users toward their healthy-eating goals. Moreover, we are planning to use our system in conjunction with other technologies and develop a hybrid nutrition monitoring platform with multi-modalality inputs.

Our current design in this article requires the users to record dietary data consistently. In reality, however, the potential disadvantages of this approach can be overcome by designing intelligent algorithms that eliminate the need for consistent recording of the food intake. In our future work, we aim to predict user's input based on diet history and contextual data such as activity and location captured using embedded sensors such as accelerometer, gyroscope, magnetometer, and GPS sensors. Our working hypothesis is that diet patterns change gradually because humans tend to eat similar foods frequently. We assume that diet patterns can be predicted based on contextual data such as activity and location. By accurately predicting the user's input, our system can begin to monitor nutrition and provide diet recommendations without (or with minimal) explicit input from the user.

7 CONCLUSIONS

The major contribution in this work is the introduction and validation of a novel approach for nutrition monitoring based on unstructured data such as spoken data and free-format text. This research provides a pervasive approach for recording and understanding natural language for diet assessment by integrating advances in speech recognition, named entity recognition, incremental learning, text analysis, and mobile health. The goal of using unstructured data is to provide a more convenient nutrition monitoring technology for users in different situations. We utilized named entity recognition algorithms to identify nutrition-specific information within the input data. A prompting unit was proposed to ask the end-user for any unclear or missing information. A mapping unit was proposed utilizing integer linear programming for mapping food names to their associated food quantities. A 2-tier approach was devised for analyzing the text for calorie computation. The performances of the incremental training models and the prompting unit were evaluated based on the number of users during 13 days in three phases. The error of incremental learning models with 10 participants in phase III was 17% and 33% less than that of phase II with two participants and phase I with one participant, respectively. The impacts of using incremental learning models and prompting unit were evaluated as well. The performance of the system with incremental training was 89.7%, which is 49.3% higher than the performance of the system without the use of these models.

ACKNOWLEDGMENTS

This work was supported in part by the United States National Science Foundation, under grants CNS-1566359 and CNS-1750679. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] P. Alinia, R. Saeedi, R. Fallahzadeh, A. Rokni, and H. Ghasemzadeh. 2016. A Reliable and Reconfigurable Signal Processing Framework for Estimation of Metabolic Equivalent of Task in Wearable Sensors. *IEEE Journal of Selected Topics in Signal Processing* 10, 5 (Aug 2016), 842–853. https://doi.org/10.1109/JSTSP.2016.2569472

[2] Parastoo Alinia, Ramyar Saeedi, Ramin Fallahzadeh, Ali Rokni, and Hassan Ghasemzadeh. 2016. A reliable and reconfigurable signal processing framework for estimation of metabolic equivalent of task in wearable sensors. *IEEE Journal of Selected Topics in Signal Processing* 10, 5 (2016), 842–853.

[3] Nabil Alshurafa, Haik Kalantarian, Mohammad Pourhomayoun, Jason J Liu, Shruti Sarin, Behnam Shahbazi, and Majid Sarrafzadeh. 2015. Recognition of nutrition intake using time-frequency decomposition in a wearable necklace using a piezoelectric sensor. *IEEE sensors journal* 15, 7 (2015), 3909–3916.

[4] J Anderson and L Rainie. 2012. Main findings: Teens, technology and human potential. *Millennials will benefit and suffer due to their hyperconnected lives. Pew Internet and American Life Project. Retrieved from http://pewInternet. org/Reports/2012/Hyperconnected-lives/Main-findings/Skills. aspx* (2012).

[5] Jeannette Beasley, William T Riley, and Jersino Jean-Mary. 2005. Accuracy of a PDA-based dietary assessment program. *Nutrition* 21, 6 (2005), 672–677.

[6] Dimitris Bertsimas and John N Tsitsiklis. 1997. *Introduction to linear optimization*. Vol. 6. Athena Scientific Belmont, MA.

[7] Andrew JM Boulton, Loretta Vileikyte, Gunnel Ragnarson-Tennvall, and Jan Apelqvist. 2005. The global burden of diabetic foot disease. *The Lancet* 366, 9498 (2005), 1719–1724.

[8] Carol Jo Boushey, Amelia J Harray, Deborah Anne Kerr, TusaRebecca E Schap, Stacey Paterson, Tanisha Aflague, Marc Bosch Ruiz, Ziad Ahmad, and Edward J Delp. 2015. How willing are adolescents to record their dietary intake? The mobile food record. *JMIR mHealth and uHealth* 3, 2 (2015).

[9] Carol J Boushey, Deborah A Kerr, Janine Wright, Kyle D Lutes, David S Ebert, and Edward J Delp. 2009. Use of technology in childrenâĂŹs dietary assessment. *European journal of clinical nutrition* 63, S1 (2009), S50.

[10] Robert A Carels, Jessica Harper, and Krista Konrad. 2006. Qualitative perceptions and caloric estimations of healthy and unhealthy foods by behavioral weight loss participants. *Appetite* 46, 2 (2006), 199–206.

[11] Michelle Clare Carter, Victoria Jane Burley, Camilla Nykjaer, and Janet Elizabeth Cade. 2013. Adherence to a smartphone application for weight loss compared to website and paper diary: pilot randomized controlled trial. *Journal of medical Internet research* 15, 4 (2013).

[12] Hsin-Chen Chen, Wenyan Jia, Zhaoxin Li, Yung-Nien Sun, and Mingui Sun. 2012. 3D/2D model-to-image registration for quantitative dietary assessment. In *Bioengineering Conference (NEBEC), 2012 38th Annual Northeast*. IEEE, 95–96.

[13] Sunny Consolvo, David W McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, et al. 2008. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1797–1806.

[14] Nicolas E Day, N McKeown, Man-Yu Wong, Ailsa Welch, and Shelia Bingham. 2001. Epidemiological assessment of diet: a comparison of a 7-day diary with a food frequency questionnaire using urinary markers of nitrogen, potassium and sodium. *International Journal of Epidemiology* 30, 2 (2001), 309–317.

[15] Daniel Evans. 2017. MyFitnessPal. *Br J Sports Med* 51, 14 (2017), 1101–1102.

[16] Ramin Fallahzadeh and Hassan Ghasemzadeh. 2017. Personalization Without User Interruption: Boosting Activity Recognition in New Subjects Using Unlabeled Data. In *Proceedings of the 8th International Conference on Cyber-Physical Systems (ICCPS '17)*. ACM, New York, NY, USA, 293–302. https://doi.org/10.1145/3055004.3055015

[17] Muhammad Farooq and Edward Sazonov. 2016. Segmentation and characterization of chewing bouts by monitoring temporalis muscle using smart glasses with piezoelectric sensor. *IEEE journal of biomedical and health informatics* 21, 6 (2016), 1495–1503.

[18] Pedro Ferreira, Pedro Sanches, Kristina Höök, and Tove Jaensson. 2008. License to chill!: how to empower users to cope with stress. In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*. ACM, 123–132.

[19] USDA Food. 2004. nutrient database for dietary studies, 4.1. Beltsville, MD: Agricultural Research Service. *Food Surveys Research Group* (2004).

[20] H Ghasemzadeh, RAMIN Fallahzadeh, and ROOZBEH Jafari. 2016. A hardware-assisted energy-efficient processing model for activity recognition using wearables. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 21 (2016).

[21] Annelies HC Goris, Margriet S Westerterp-Plantenga, and Klaas R Westerterp. 2000. Undereating and underrecording of habitual food intake in obese men: selective underreporting of fat intake. *The American journal of clinical nutrition* 71, 1 (2000), 130–134.

[22] Steven L Gortmaker, Boyd A Swinburn, David Levy, Rob Carter, Patricia L Mabry, Diane T Finegood, Terry Huang, Tim Marsh, and Marjory L Moodie. 2011. Changing the future of obesity: science, policy, and action. *The Lancet* 378, 9793 (2011), 838–847.

[23] Hongsheng He, Fanyu Kong, and Jindong Tan. 2016. DietCam: Multiview Food Recognition Using a MultiKernel SVM. *IEEE journal of biomedical and health informatics* 20, 3 (2016), 848–855.

[24] Niloofar Hezarjaribi, Ramin Fallahzadeh, and Hassan Ghasemzadeh. 2016. A machine learning approach for medication adherence monitoring using body-worn sensors. In *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. EDA Consortium, 842–845.

[25] Niloofar Hezarjaribi, Sepideh Mazrouee, and Hassan Ghasemzadeh. 2017. Speech2Health: A mobile framework for monitoring dietary composition from spoken data. *IEEE journal of biomedical and health informatics* 22, 1 (2017), 252–264.

[26] Niloofar Hezarjaribi, Cody Reynolds, Drew Miller, Naomi Chaytor, and Hassan Ghasemzadeh. 2016. S2NI: A Mobile Platform for Nutrition Monitoring from Spoken Data. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE.

[27] Wenyan Jia, Yaofeng Yue, John D Fernstrom, Zhengnan Zhang, Yongquan Yang, and Mingui Sun. 2012. 3D localization of circular feature in 2D image and application to food volume estimation. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 4545–4548.

[28] Chucri A Kardous and Peter B Shaw. 2014. Evaluation of smartphone sound measurement applications a. *The Journal of the Acoustical Society of America* 135, 4 (2014), EL186–EL192.

[29] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[30] Santosh Kumar, Wendy Nilsen, Misha Pavel, and Mani Srivastava. 2013. Mobile health: Revolutionizing healthcare through transdisciplinary research. *Computer* 46, 1 (2013), 28–35.

[31] Brian Yoshio Laing, Carol M Mangione, Chi-Hong Tseng, Mei Leng, Ekaterina Vaisberg, Megha Mahida, Michelle Bholat, Eve Glazier, Donald E Morisky, and Douglas S Bell. 2014. Effectiveness of a smartphone application for weight loss compared with usual care in overweight primary care patients: a randomized, controlled trial. *Annals of internal medicine* 161, 10_Supplement (2014), S5–S12.

[32] Yuchao Ma and Hassan Ghasemzadeh. 2019. LabelForest: Non-Parametric Semi-Supervised Learning for Activity Recognition. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. 4520–4527. https://aaai.org/ojs/index.php/AAAI/article/view/4369

[33] Stephen DH Malnick and Hilla Knobler. 2006. The medical complications of obesity. *Journal of the Association of Physicians* 99, 9 (2006), 565–579.

[34] Corby K Martin, Sertan Kaya, and Bahadir K Gunturk. 2009. Quantification of food intake using food image analysis. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 6869–6872.

[35] Beverly McCabe-Sellers. 2010. Advancing the art and science of dietary assessment through technology. *Journal of the American Dietetic Association* 110, 1 (2010), 52–54.

[36] J Michael McGinnis and William H Foege. 1993. Actual causes of death in the United States. *Jama* 270, 18 (1993), 2207–2212.

[37] LLC MyFitnessPal. 2015. Myfitnesspal.

[38] Kevin Patrick, Fred Raab, Marc A Adams, Lindsay Dillon, Marian Zabinski, Cheryl L Rock, William G Griswold, and Gregory J Norman. 2009. A text message–based intervention for weight loss: randomized controlled trial. *Journal of medical Internet research* 11, 1 (2009).

[39] Joel JPC Rodrigues, Ivo MC Lopes, Bruno MC Silva, and Isabel de La Torre. 2013. A new mobile ubiquitous computing application to control obesity: SapoFit. *Informatics for Health and Social Care* 38, 1 (2013), 37–53.

[40] Seyed Ali Rokni and Hassan Ghasemzadeh. 2016. Plug-n-learn: Automatic Learning of Computational Algorithms in Human-centered Internet-of-things Applications. In *Proceedings of the 53rd Annual Design Automation Conference (DAC '16)*. ACM, New York, NY, USA, Article 139, 6 pages. https://doi.org/10.1145/2897937.2898066

[41] Ramyar Saeedi, Navid Amini, and Hassan Ghasemzadeh. 2014. Patient-centric on-body sensor localization in smart health systems. In *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2081–2085.

[42] Dale A Schoeller, Diana Thomas, Edward Archer, Steven B Heymsfield, Steven N Blair, Michael I Goran, James O Hill, Richard L Atkinson, Barbara E Corkey, John Foreyt, et al. 2013. Self-report–based estimates of energy intake offer an inadequate basis for scientific conclusions. *The American journal of clinical nutrition* 97, 6 (2013), 1413–1415.

[43] Junqing Shang, Michael Duong, Eric Pepin, Xing Zhang, Kishore Sandara-Rajan, Alexander Mamishev, and Alan Kristal. 2011. A mobile structured light system for food volume estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 100–101.

[44] Yiru Shen, James Salley, Eric Muth, and Adam Hoover. 2017. Assessing the Accuracy of a Wrist Motion Tracking Method for Counting Bites Across Demographic and Food Variables. *IEEE journal of biomedical and health informatics* 21, 3 (2017), 599–606.

[45] Frances E Thompson, Amy F Subar, et al. 2008. Dietary assessment methodology. *Nutrition in the Prevention and Treatment of Disease* 2 (2008), 3–39.

[46] John Torous, Steven Richard Chan, Shih Yee-Marie Tan, Jacob Behrens, Ian Mathew, Erich J Conrad, Ladson Hinton, Peter Yellowlees, and Matcheri Keshavan. 2014. Patient smartphone ownership and interest in mobile apps to monitor symptoms of mental health conditions: a survey in four geographically distinct psychiatric clinics. *JMIR Mental Health* 1, 1 (2014).

[47] Christopher C Tsai, Gunny Lee, Fred Raab, Gregory J Norman, Timothy Sohn, William G Griswold, and Kevin Patrick. 2007. Usability and feasibility of PmEB: a mobile phone application for monitoring real time caloric balance. *Mobile networks and applications* 12, 2-3 (2007), 173–184.

[48] SY Wu and A Green. 2000. Projection of chronic illness prevalence and cost inflation Washington. *DC: RAND Health* (2000).