A Speech-Based Mobile App for Restaurant Order Recognition and Low-Burden Diet Tracking

Xiaochen Huang and Emmanuel Agu(✉)

Computer Science Department, Worcester Polytechnic Institute,
Worcester, MA, USA
emmanuel@cs.wpi.edu

Abstract. Obesity is a public health problem in the US. Diet tracking helps control obesity but manual entry is tedious. Proposed solutions such as food recognition from photographs and scanning barcodes have limitations. We investigate the use of speech recognition for diet recording. We improved the accuracy of food order recognition at restaurants by (1) limiting words in the speech recognizer's corpus to only items on the menus of nearby restaurants (2) implementing an acoustic model to recognize the speaking style of the smartphone owner. Building on these mechanisms, we propose DietRecord, a smartphone application that automatically recognizes and records restaurant orders. Results of our user studies to evaluate DietRecord were encouraging.

1 Introduction

Obesity rates have more than doubled since the 1970s [1]. Obesity increases the risk of many health conditions including hypertension and diabetes. Diet tracking helps control obesity, and to manage ailments requiring controlled diets. Manually recording food is tedious, which reduces compliance. Several proposed solutions have limitations. Recognizing foods from photographs [2] cannot recognize certain foods. Scanning food QR codes [9] or barcodes [10] works but not all foods have barcodes. Improved user entry interfaces have also been proposed but are also manual [8].

Speech is one of the most natural methods of interaction. However, in practice, factors such as environmental noise and individual speaking styles make speech recognition not accurate enough for food entry. Our work focuses on recognizing spoken food orders at restaurants since 25 % of Americans consume fast food daily [3]. We improve speech recognition accuracy and speed using two concepts:

1. *Location-dependent speech recognizer vocabularies:* Just before a user orders food at a restaurant, we limit the words the speech recognizer can guess was spoken (recognition range) by pre-populating the speech recognizer's corpus with only menu items from nearby restaurants.
2. *Speaker Personalization:* Since the speech recognizer may pick up the orders of other customers and restaurant staff, we train the recognizer on the speaking style and accent of the smartphone owner.

X. Zheng et al. (Eds.): ICSH 2015, LNCS 9545, pp. 333–339, 2016.
DOI: 10.1007/978-3-319-29175-8_31

Leveraging these mechanisms, we propose *DietRecord*, a smartphone app that automatically recognizes and records foods ordered by its owner at restaurants. DietRecord users can browse order history and nutrition information using its interfaces.

2 Investigating Speech Recognition Accuracy

CMU Sphinx [4] and Google Voice Service [5] are currently the most widely used speech recognition systems. Using Google Voice, a user can initiate a Google search by speaking into their smartphones. CMU Sphinx is a non-commercial speech recognizer that has features useful for our work. Developers can modify the set of words that can be recognized by building a special corpus (vocabulary). They can also adapt the voice recognition system to individual accents and speaking styles by building an acoustic model. Since we wanted to leverage these customizations, we adopted Pocketsphinx [6], a mobile implementation of CMU Sphinx for our work.

To establish a baseline against which to compare our novel ideas, we initially evaluated unmodified PocketSphinx and Google Voice Speech Recognition demos and their error rates under various conditions. We define the error rate as the percentage of incorrect words in a test sentence. We investigated two factors that affect error rates: (1) the length of spoken sentences and (2) the background noise (in decibels) since restaurants may be noisy. Our test sentences consisted of typical food orders such as "I want a big mac" of different lengths. PocketRTA, a software-based spectrum analyzer was run to generate various noise levels.

Without our proposed enhancements, Pocketsphinx had low speech recognition accuracy, with an error rate of about 65.5 % (Fig. 1) and performs worse than Google Voice. While recognition accuracy is inversely proportional to noise levels, we found no relationship between speech recognizer accuracy and sentence length.

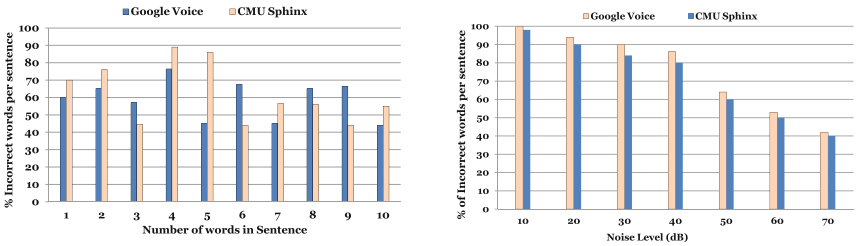

Fig. 1. Accuracy of PocketSphinx unmodified vs Google Voice Speech Recognition for different sentence lengths (left) and background noise (right)

3 Improving Speech Recognition Accuracy

We then explored location-dependent vobularies and speaker personalization to improve the accuracy of spoken order recognition.

(1) *Location-Dependent Vocabularies to limit Speech Recognizer Range:* The corpus of PocketSphinx was pre-populated with only 121 food items from the menus of Dunkin Donuts and McDonalds, two fastfood restaurants. After rebuilding the recognizer's corpus, a subject read out various orders from each of these menus. Figure 2 shows the recognition rate for various text lengths and environmental noise levels (in decibels). Clearly, limiting the recognizer's range significantly improved its accuracy.

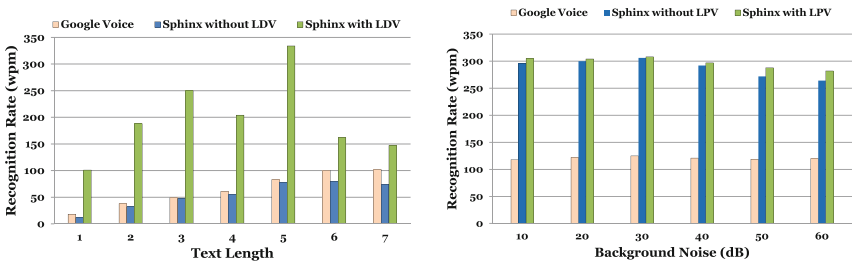

Fig. 2. Speech recognition speeds of CMU Sphinx vs Google Voice for different text lengths (left) and background noise (right)

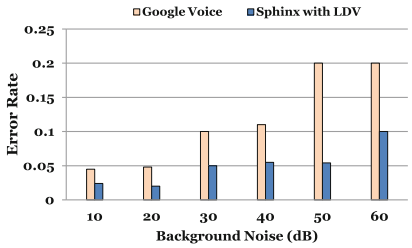

Fig. 3. Error rates of Sphinx with Location-Dependent Vocabularies (LDV) vs Google Voice, different background noise

The error rate was calculated as the number of errors in recognized words divided by the total number of words spoken. Figure 3 shows that Pocketsphinx with the limited recognition corpus has a lower error rate than Google Voice.

(2) *Personalization based on Speaker Identification:* CMU Sphinx has several high-quality acoustic models we used to adapt to the user's speaking style. To train the acoustic models, we generated an adaptation corpus by recording the smartphone owner speaking menu items. This adaption data consisted of a list of order sentences, a dictionary describing the pronunciation of all words in that list of sentences, and a recording of users speaking each of those sentences.

4 The DietRecord Restaurant Order Recognition App

Overview: Based on our findings, we created the DietRecord Android app that automatically recognizes and records food orders spoken by a smartphone user. DietRecord detects the user's arrival at a restaurant and retrieves the menus of nearby restaurants using the Foursquare API. These menu items are used as a corpus into DietRecord's PocketSphinx-based speech recognition module. Distances to all restaurants in the user's vicinity are computed and the user is assumed to be in the restaurant that is closest to their current location. An acoustic model of the smartphone owner's speech is also used to customize DietRecord for the smartphone owner's speaking style. DietRecord inserts recognized foods into appropriate categories (breakfast, lunch, dinner, snack), based on the time the order is placed. Figure 4 shows the DietRecord system architecture.

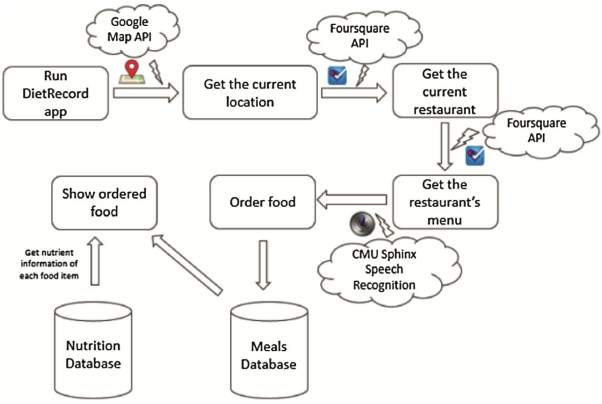

Fig. 4. DietRecord system architecture

User Interaction Modes Supported by DietRecord: Recognized food orders are stored in a database on the smartphone. Over time, meal entries become a diet diary, which also includes nutrition information such as calories, carbohydrates, protein. Users may interact with DietRecord either by speaking their food orders or using DietRecord's interfaces to browse their order history.

Nearby Restaurant Detection: The Google Map API was used to detect the user's current location (longitude and latitude), which is used by the Foursquare API to obtain a list of restaurants within 500 m and their menus. Figure 5 shows (a) DietRecord's restaurant detection screen. Figure 5(b) shows some recognized foods.

Extraction of Nutrition Information of Ordered Food: DietRecord presents users with nutrition information of their recorded diet (See Fig. 5(c)), which was retrieved from the USDA national nutrition database.

DietRecord Implementation: DietRecord is implemented in the Android operating system. DietRecord's interfaces were created by extending Android's activity class. A Service class detected the user's location and performed background processing such

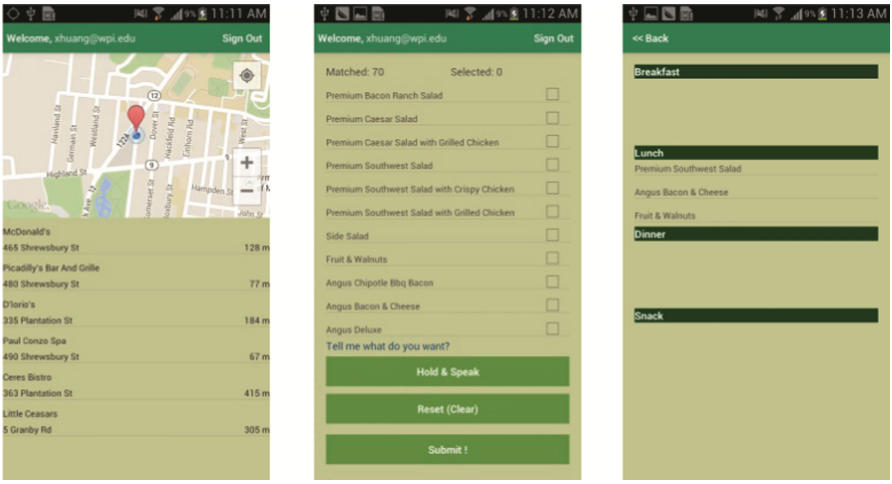

Fig. 5. Screens (a) Restaurant detection (b) Recognized foods (c) Nutrition information.

as running the PocketSphinx speech recognition engine. Content Providers were used to store data in DietRecord. An intent was used to start an activity, service or broadcast from another activity. The diet data was stored in SQLite, a lightweight database.

5 DietRecord Evaluation

Study 1: We recruited 23 Worcester Polytechnic Institute students aged 23 to 29 (15 male, 8 female). Participants were given a questionnaire to gauge their general interest

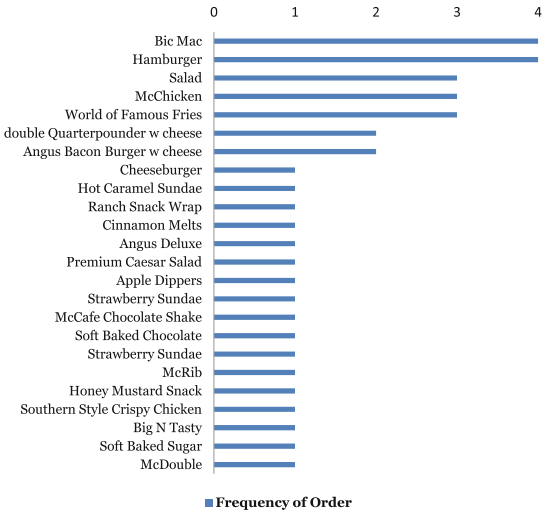

Fig. 6. Distribution of foods ordered by subjects.

in a diet recording application such as DietRecord. To avoid biasing their responses, we did not present this group with DietRecord or any actual app. Over 82 % of participants were concerned with their daily calorie intake and over 50 % of participants wanted a mobile app that could automatically track their meals.

Study 2: Four other subjects ran DietRecord while ordering food at McDonald's fast food restaurants. They ordered a total of 40 items (26 kinds of food items). Figure 6 shows the distribution of foods ordered by the subjects. When ordering food, subjects spoke with their mouths about 5 inches from the phone's microphone. After testing DietRecord, the subjects were asked questions. For food orders of 1–6 words, the recognition accuracy ranged from 80 to 93.3 % (average of 86.4 %). Recognition accuracy was not influenced by the length of the food name, but by whether the word was a compound word (e.g. Big mac). We computed accuracy as:

$$Accuracy = Number\ of\ correctly\ recognized\ words\ /\ Total\ number\ of\ words\ in\ a\ dish \quad (1)$$

6 Related Work

Other mobile speech entry applications include Parakeet [7], a continuous speech recognition system for mobile touch-screen devices. Users entered text by speaking into their phones. However, Parakeet was less accurate than desk tops and users experienced delays as long as 1 min for some words. Commercial diet tracking apps such as MyFitnessPal have also been proposed but require manual input.

7 Conclusion and Future Work

In this paper, we investigated using speech recognition to record foods ordered by smartphone users at restaurants. We demonstrated higher recognition accuracy and speed by building location-dependent speech vocabularies to limit the recognition range and an acoustic model to adapt to the smartphone owner's speaking style. We proposed DietRecord, a smartphone app that can automatically record users' restaurant orders. Participants in our user studies found DietRecord convenient and useful.

However, DietRecord has some limitations. The Foursquare API we used for retrieving a list of nearby restaurants has some missing food items. The DietRecord app also requires GPS to detect the user's current location, which drains the smartphone's battery. Making DietRecord energy efficient is future work. Finally, the subjects in our user study were mostly from China. In future we will diversify our participants.

References

1. National Center for Health Statistics. http://www.cdc.gov/nchs/
2. Kong, F., Tan, J.: DietCam: regular shape food recognition with a camera phone. In: Proceedings of the BSN. IEEE (2011)

3. Food Research and Action Center (FRAC), Overweight and Obesity in the US. http://frac.org/initiatives/hunger-and-obesity/obesity-in-the-us/
4. Rozzi, W.A., Stern, R.M.: Speaker adaptation in cont. speech recognition via estimation of correlated mean vectors. In: IEEE Proceedings of the ICASSP 1991 (1991)
5. Jyothi, P., et al.: Distributed discriminative language models for Google voice-search. In: Proceedings of the ICASSP. IEEE (2012)
6. Huggins-Daines, D., et al.: Pocketsphinx: a free, real- time cont. speech recognition system for hand-held devices. In: Proceedings of the IEEE ICASSP (2006)
7. Vertanen, K., Kristenssen, P.: Parakeet: a cont. speech recognition system for mobile touch-screen devices. In: Proceedings of the ACM IUI 2009 (2009)
8. Andrew, A.H., Boriello, G., Fogarty, J.: Simplifying mobile phone food diaries. In: Proceedings of the Pervasive Health 2013 (2013)
9. Hsu, H.H., Min-Ho, C., Neil, Y.Y.: A health man- agement application with QR-code input and rule inference. In: Proceedings of the ISIC. IEEE (2012)
10. Ohbuchi, E., Hiroshi, H., Lim, A.H.: Barcode readers using the camera device in mobile phones. In: Proceedings of the International Conference Cyberwords (2004)