

Voice Control in Calorie Tracker Application using Levenshtein Distance Algorithm

Daniel Halim¹, Seng Hansun¹

¹Department of Computer Science, Universitas Multimedia Nusantara, Tangerang, 15811, Indonesia

*Corresponding author email: hansun@umn.ac.id

Received : August 24, 2017

Accepted : January, 18, 2018

Online : April, 30, 2018

Abstract– Each food consumed by people contains a number of calories needed by the body to perform an activity. Calories can be described as fuel of engine to move and carry out tasks. However, ignorance on the food consumption can cause some negative effects on health like, too skinny, obesity, and emergence of various diseases. Therefore, it is necessary to have an application which can provide some information about one's calorie needs, so that he/she can control the calorie intake. This paper describes the development of an application to assist consumers in controlling calorie intake according to the calorie needs. The application supports voice control feature to improve user comfort in performing certain commands and inputting food consumed. In addition, the application also uses Levenshtein Distance algorithm to correct food recognition errors which is spoken by user. The application is developed using Java programming language for Android with SQLite database.

Key words: Android, Calorie Tracker, Levenshtein Distance, Voice Control.

Introduction

Currently, most people are busy with their own activities while ignoring their health, either the children, teenagers, or adults. Most of them are not aware of their health, such as eating instant food, doing too many activities that cause the lack of rest, and never doing sports. Though eating nutritious food, taking adequate rest, and doing some exercises are the key for maintaining health (Sasando, 2014).

The most important factor that is often overlooked by people nowadays is their food. They tend to eat the potluck food (instant or fast food) or some even skip their lunch or dinner just because one of their activities has not been completed. These things can cause trouble for people's health either the posture is too thin, obesity, or the emergence of other various diseases.

In Indonesia, the number of obese people has increased every year (National Geographic, 2014). Unhealthy lifestyle (related to the food consumption) is one of the main causes that cause increasing population of obese people. According to the Minister of Health, Tjandra Yoga Aditama, based on the existing data, the number of obese people in Indonesia are not too high compared to other Asian-Oceania countries. However, the number rises with the prevalence (incidence cases) up to 20% every year (Pikiran Rakyat, 2013).

Based on the problems that have been mentioned above, it is needed an application that can calculate the ideal calorie needed by each person in one day. With varying levels of age, weight, height, gender, and activity, whether students or employers who have light activity, or athletes who practice all the time, will have different calorie needs in a day. Each person can generate optimum energy to perform daily activities with adequate amount of calories (Stephanie, 2012). Therefore, the goal of this study is to design an Android application that can count calorie needed and control the amount of calories they have eaten every day.

This application was made for mobile smartphones with the Android operating system. Almost everyone have a mobile device and their daily activities are also related to their mobile devices. They always bring and use their mobile devices anywhere and anytime. For some reasons, such as simply, flexibility, and can be accessed offline, then this application will be made on mobile devices.

Today, modern portable devices, such as smartphones and tablets, have been equipped with a

microphone and touch screen display. Selecting a menu and typing with the onscreen keyboard is the main way for user to enter the input and perform the commands on portable devices. However, in many situations, user can only type using one hand, while the other hand holding the device. In addition, typing on a small-size device would also be very difficult, so the speed will be slower than typing on a computer (Sim et al., 2012). Meanwhile, this application provides a variety menu of foods (approximately 80 menus) for users to select and enter the variety of foods that has been consumed, which indirectly would complicate users if they have to type and select the menu from the number of options available. Based on this problem, this application will provide the voice input feature, which will allow the user to enter the menu of food consumed and also to do some voice commands (Sim et al., 2012).

Automatic Speech Recognition (ASR) technology has become a popular way to communicate with mobile phones. ASR technology provides verbal communication between user and mobile device, as well as the experience of natural interaction with user. It converts speech from a recorded audio signal to text (Gruhn *et al.*, 2011). However, the recognition performance of ASR technology often does not meet the accuracy or precision of the mobile devices, especially caused by the noisy environment (Moon and Sim, 2012).

Therefore, in this application, Levenshtein Distance algorithm will be implemented to improve the accuracy of user's voice input. This algorithm will fix the voice input errors, either because of noise or pronunciation errors, by checking the voice input with the words in database.

Theories and Design

Calorie

Calorie is a unit to calculate the amount of energy. To know someone's calorie needs, it can be calculated based on the Basal Metabolic Rate (BMR). BMR is the minimum calorie needed by someone just to sustain his/her life, assuming that the person is in a state of complete rest, no activity at all (Hulbert and Else, 2004). The following equation is used to calculate BMR.

$$\text{Male BMR} = 66 + (13.7 * \text{Weight}) + (5 * \text{Height}) - (6.8 * \text{Age}) \quad (1)$$

$$\text{Female BMR} = 655 + (9.6 * \text{Weight}) + (1.8 * \text{Height}) - (4.7 * \text{Age}) \quad (2)$$

Where the following provisions are:

- Weight in Kg
- Height in cm
- Age in years

In addition, the level of user activity also has considerable influence in managing human energy. The amount of energy used for activities will vary according to the intensity and the level of activity. Each level of activity has its own constant in the calculation of daily calorie needed. One way to measure the intensity of human activity is the Metabolic Equivalent Task (MET). The activity levels were measured by MET can be divided into three levels, which is light, moderate, and heavy. Table 1 is a table of the constant level of activity for male and female at each level (Stephanie, 2012).

Table 1. Constant Level of Activity

Activity	Male	Female
Light	1.3	1.3
Moderate	1.65	1.55
Heavy	1.76	1.7

After knowing the constant level of activity, then used the Harris-Benedict equation to calculate daily calorie needs as follows.

$$\text{Calorie Needs} = \text{BMR} \times \text{Constant Level of Activity} \quad (3)$$

Furthermore, the proportion of the body, Body Mass Index (BMI) or the ideal body weight will be checked using the following equation.

$$BMI = Weight / (Height * Height) \tag{4}$$

with the following provisions:

- Weight in Kg
- Height in m

Based on the results from BMI calculations, someone can be classified into a specific category, which will also affect their calorie needs. Table below is a table of category and its effect to calorie needs.

Table 2. Category and Its Effect

BMI	Category	Calorie
< 18,5	Underweight	Calorie+100
18,5<=x<= 24,9	Normal	Calorie
> 24,9	Overweight	Calorie-700

Voice recognition

Voice recognition is a process of recognizing the spoken words and then converting it into the text form automatically. In other words, voice recognition is a process that allows the computer to recognize what someone said and then turn it into written form. The input data is received as human speech or voice, then the system will identify the words or sentences spoken and generate output as text according to what were said. The spoken words are converted into digital signals by converting sound waves into a bunch of numbers and then adapted to the specific codes and matched with a pattern stored in a device. Each different utterance will produce different characteristic patterns (Nurcahyadi, 2014).

There are two types of voice recognition viewed from the speaker dependence, which are:

- (1) Independent Speech Recognition, which is the speech recognition system that is not affected by who is speaking, but has limitations in the amount of vocabulary. This model will match each utterance with a recognizable word and select the "seems to" match.
- (2) Dependent Speech Recognition, which is the speech recognition system that requires the special training of speaker, where the results of the training of each speaker will be stored in a profile. This profile will be used to interact with a speech recognition system and the system will work depend on who's talking.

Levenshtein Distance

Levenshtein Distance algorithm is a matrix that is used to measure the difference of distance between two sequences. Levenshtein Distance is created by Vladimir Levenshtein in 1965. The calculation of edit distance is obtained from the matrix that is used to calculate the amount of difference string between two strings. The calculation of distance between two strings is determined from the minimum number of alteration operation to make string A into string B. There are three kinds of main operations that can be performed by this algorithm, which are (Adriyani *et al.*, 2012; Winarsono *et al.*, 2009):

- (1) Operation of Character Alteration

Operation of character alteration is an operation to swap a character with another character, for example the author writes the string 'yang' to 'yng'. In this case the character 'a' is replaced with the letter 'n'.

- (2) Operation of Character Addition

Operation of character addition means adding characters into a string. For example, the string 'kepad' to string 'kepada', is performed the addition of the character 'a' at the end of the string. The addition of the character is not only performed at the end of the word, but could be added at the beginning or in the middle of the string.

- (3) Operation of Character Deletion

Operation of character deletion is performed to remove characters from a string. For example, in the string 'barur', the last character is removed so that it becomes 'baru'. In this operation, it is performed the removal of character 'r'.

The greater the number generated by the Levenshtein Distance operation, the greater the difference between the two strings.

For more details, see the illustration matrix of Levenshtein Distance between two strings as

follows, 'penjara' and 'jarak'. Both strings, 'penjara' and 'jarak' have a range of 4. That means to convert the string 'jarak' to 'penjara' required 4 operations as follows (Nafik et al., 2014).

- (1) Insert character 'p' jarak -> pjarak
- (2) Insert character 'e' pjarak -> pejarak
- (3) Insert character 'n' pejarak -> penjarak
- (4) Delete character 'k' penjarak -> penjara

The matrix representation can be shown by the following figures.

		P	E	N	J	A	R	A
	0	1	2	3	4	5	6	7
J	1	1	2	3	3	4	5	6
A	2	2	2	3	4	3	4	5
R	3	3	3	3	4	4	3	4
A	4	4	4	4	4	4	4	3
K	5	5	5	5	5	5	5	4

Figure 1. Levenshtein Distance Matrix for “Jarak” and “Penjara” (Source: Nafik *et al.*, 2014)

This algorithm starts from the top left corner of a two-dimensional array that has been filled with some initial numbers. The cost value on the bottom right corner is the edit distance value that describes the amount of the difference between two strings.

From the Figure 1, it is obtained that the minimum number of modification that have to be performed to convert the string 'jarak' into string 'penjara' is four operations, which is: the insertion of character 'p', the insertion of character 'e', the insertion of character 'n', and the deletion of the character 'k'.

Application design

In Figure 2, it can be seen that when the user run the application, then the application will check the database. When the application is run for the first time, which means the user is a new user, then the application will display the registration page and ask the user to enter some data, such as name, age, sex, weight, height, and activity level that was performed by user. These data are needed by the application to calculate the calorie needs. However, if the user has ever done the registration before, the user will go directly to the main page every time the user runs the application. On the main page, there are four main tabs, which are profile, menu, tracker, and about. Each tab has an interface and their respective functions.

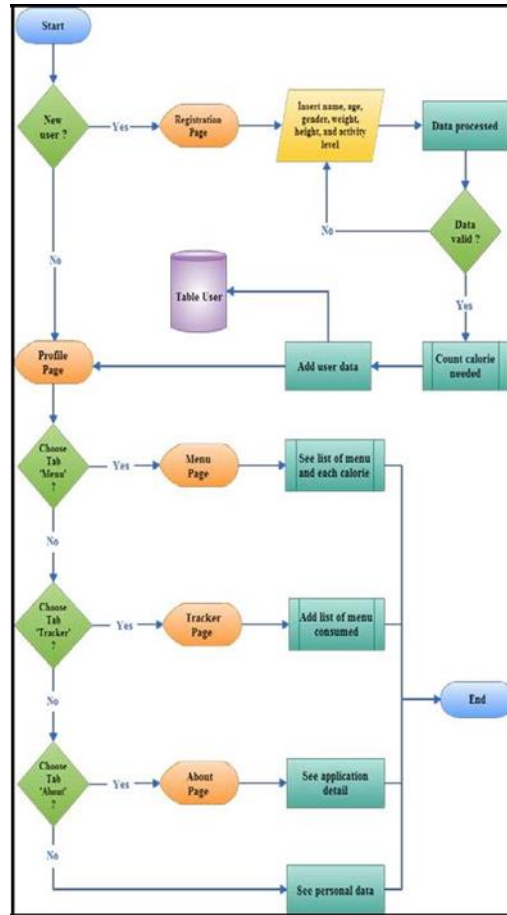


Figure 2. General Flowchart

Results and Discussion

The application design that has been completely implemented is represented in the pages which are called activities in Android Programming. On the first time usage of calorie tracker application, user is asked to fill in some data to be used by the application in calculating the calorie needs.



Figure 3. Registration Page

Then on the profile page, user can see and edit the personal data that has been entered on the registration page. In addition, user can also see the calculation results from the application, such as BMR, BMI, and calorie needs.



Figure 4. Profile Page

On the menu page, the application will display a list of menu for each category and its amount of calories. If some menu is not available on the list that has been provided, then user can add the new menu with its amount of calories by himself to the list of menu according to what he wants to add.



Figure 5. Menu Page

Next on the tracker page, the application will display the calorie needed on a certain day and the list of food that has been consumed on the same day. Furthermore, user can also see the history of food consumed since last week to a month ago.

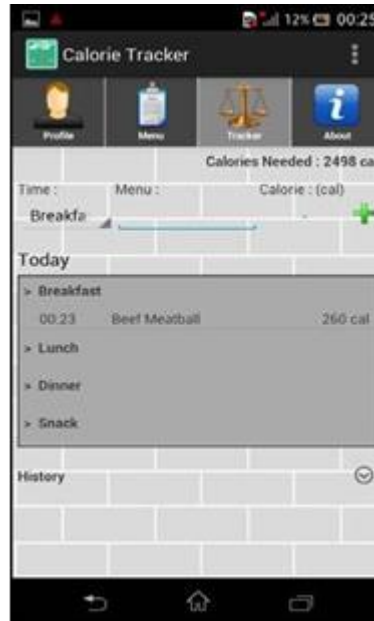


Figure 6. Tracker Page

One of the tests that had been performed in this study was viatesting the accuracy of voice recognition. Testing the accuracy of voice recognition was done by mentioning some of the same words at the microphone on the mobile phone repeatedly. The test was performed to determine the level of accuracy and consistency of voice recognition on the identical pronunciation of some words, as well as to test the performance of the Levenshtein Distance algorithm which had been applied in the application to correct errors of the spoken word. The words were tested by using English as the voice recognition feature that has been made only be able to recognize words in English. The tests were carried out by using 20 data samples spoken five times repeatedly.

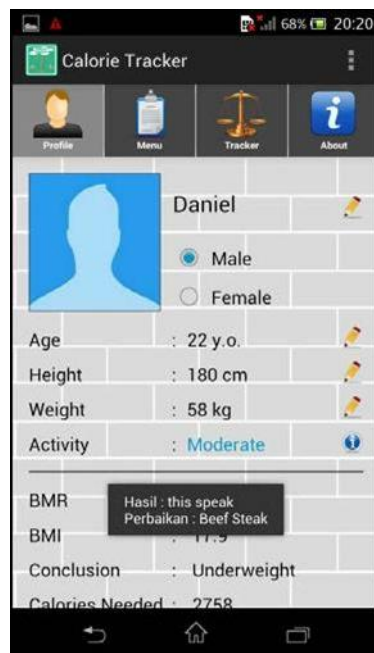


Figure7. Voice Recognition Test for the Word “Beef Steak”

Based on these test results, the Levenshtein Distance algorithm has been implemented

successfully in improving the accuracy of voice recognition to recognize the spoken words, from the accuracy percentage of 51% to 86%. Levenshtein Distance algorithm has improved some of the words' error detected by voice recognition into a new word which is correct and match with the words found in the database. The testing data can be seen on Table 3.

Table 3. Testing data and results

No	Word	Experiment No.	Result	Correction Result
1	Beef steak	1	Bee stick	Beef Steak
		2	Beef steak	Beef Steak
		3	This speak	Beef Steak
		4	Beef steak	Beef Steak
		5	Be speak	Beef Steak
2	Corn chip	1	Corn dip	Corn chip
		2	Corn chips	Corn chip
		3	Corn chips	Corn chip
		4	Corn chips	Corn chip
		5	Corn chips	Corn chip
3	Celery	1	Salary	Celery
		2	Celery	Celery
		3	Salary	Celery
		4	Celery	Celery
		5	Salary	Celery
4	Pear	1	Beer	Beef
		2	Pier	Pear
		3	Pr	Pear
		4	Pier	Pear
		5	Pear	Pear
5	Eggplant	1	Explain	Eggplant
		2	Explain	Eggplant
		3	Explain	Eggplant
		4	Explain	Eggplant
		5	Explain	Eggplant
6	Sunkist	1	Sun keys	Sunkist
		2	Sunkist	Sunkist
		3	Sunkist	Sunkist
		4	Sun keys	Sunkist
		5	Sunkist	Sunkist
7	Grilled beef	1	Your beef	Beef
		2	Deal beef	Beef
		3	Give you a beef	Grilled beef
		4	Give you a beef	Grilled beef
		5	Guilty	Guava
8	Roll cake	1	Whole cake	Roll cake
		2	Whole cake	Roll cake
		3	Call cake	Roll cake
		4	Call cake	Roll cake
		5	Call cake	Roll cake
9	Rice	1	Rice	Rice
		2	Rice	Rice
		3	Ice	Rice
		4	Raise	Rice
		5	Rice	Rice

10	Apple juice	1	Apple juice	Apple juice
		2	Apple use	Apple juice
		3	Apple juice	Apple juice
		4	Apple juice	Apple juice
		5	Apple juice	Apple juice
11	Fried rice	1	That's nice	-
		2	Fried rice	Fried rice
		3	Great price	Fried rice
		4	Set price	-
		5	Fried rice	Fried rice
12	Fried chicken	1	Fried chicken	Fried chicken
		2	Fried chicken	Fried chicken
		3	Fried chicken	Fried chicken
		4	Fried chicken	Fried chicken
		5	Fried chicken	Fried chicken
13	Spinach	1	Peanuts	Peanuts
		2	Spin it	Spinach
		3	Spinach	Spinach
		4	Speed this	-
		5	Peanuts	Peanuts
14	Cabbage	1	Cabbage	Cabbage
		2	Cabbage	Cabbage
		3	Cabinet	Cabbage
		4	Cabbage	Cabbage
		5	Cabinet	Cabbage
15	Pancake	1	Pancake	Pancake
		2	Pancake	Pancake
		3	Pancake	Pancake
		4	Pancake	Pancake
		5	Pancake	Pancake
16	Soursop	1	Sour soup	Soursop
		2	Soursop	Soursop
		3	Somerset	-
		4	Soursop	Soursop
		5	Soursop	Soursop
17	Instant coffee	1	Instead coffee	Instant coffee
		2	In stone coffee	Instant coffee
		3	Instead coffee	Instant coffee
		4	Instant coffee	Instant coffee
		5	Instant coffee	Instant coffee
18	Lettuce	1	Lettuce	Lettuce
		2	Lettuce	Lettuce
		3	Lettuce	Lettuce
		4	Lettuce	Lettuce
		5	Lettuce	Lettuce
19	Squash	1	What's	Snaps
		2	What's	Snaps
		3	Chris	Corn
		4	What's	Snaps
		5	Quest	Squash
20	Starfruit	1	Star fruit	Starfruit
		2	Star fruit	Starfruit
		3	Star fruit	Starfruit

	4	Star fruit	Starfruit
	5	Star fruit	Starfruit
Accuracy Percentage		51%	86%

Conclusions

Based on the implementation and experiment results that has been performed in our study, we come out that the Levenshtein Distance algorithm on Calorie Tracker Application has been successfully implemented in fixing words error which caused by voice recognition error, with the percentage of the words that are recognized by voice recognition is 51% and being fixed by the Levenshtein Distance algorithm to 86% accurate with the spoken word. Hopefully with this research's results, more contribution to make a better application not only for normal people but also for people with disabilities could increase.

References

- Adriyani, N.M.M., Santiyasa, I.W., and Muliantara, A. 2012. Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia. *JELIKU – Jurnal Elektronik Ilmu Komputer Universitas Udayana*, 1(1).
- Gruhn, R.E., Minker, W., Nakamura, S. 2011. *Statistical Pronunciation Modeling for Non-Native Speech Processing*, Signals and Communication Technology, Springer-Verlag, Berlin Heidelberg.
- Hulbert, A.J. and Else, P.L. 2004. Basal Metabolic Rate: History, Composition, Regulation, and Usefulness, *Physiological and Biochemical Zoology*, 77(6): 869-876.
- Moon, S. and Sim, K.C. 2012. Design and Implementation of the Note-taking Style Haptic Voice Recognition for Mobile Devices. *ACM Journal* 978-1-4503-1467-1/12/10.
- Nafik, M.Z., Indriati, and Ridok, A. 2014. Sistem Penilaian Otomatis Jawaban Esai Menggunakan Algoritma Levenshtein Distance. *Repositori Jurnal Mahasiswa PTIIK UB*, 3(4).
- National Geographic, MZW. 2014. Jumlah Orang Obesitas di Indonesia terus Meningkat. *National Geographic*, <http://nationalgeographic.co.id/berita/2014/06/jumlah-orang-obesitas-di-indonesia-terus-meningkat>. Accessed on October 12, 2016.
- Nurcahyadi, E. 2014. <http://www.scribd.com/doc/199865622/2012-1-00506-IF-Bab2001>. Accessed on October 7, 2016.
- Pikiran Rakyat, A-201/A-108. 2013. Jumlah Penderita Obesitas di Indonesia Meningkat Setiap Tahun. *Pikiran Rakyat*, <http://www.pikiran-rakyat.com/node/256976>. Accessed on October 6, 2016.
- Sasando. 2014. Masyarakat Indonesia Kurang Peduli Kesehatan. <http://sasandofm.com/news/gaya-dan-kesehatan/107-masyarakat-indonesia-kurang-peduli-kesehatan-.html>. Accessed on March 6, 2014.
- Sim, K.C., Zhao, S., Yu, K., and Liao, H. 2012. ICMP'12 Grand Challenge – Haptic Voice Recognition. *ACM Journal* 978-1-4503-1467-1/12/10: 363-370.
- Stephanie. 2012. Sistem Pakar Berbasis Web untuk Pengaturan Menu Makanan Menggunakan Decision Tree. Thesis, 0811011-00041.
- Winarsono, D., Siahaan, D.O., and Yuhana, U. 2009. Sistem Penilaian Otomatis Kemiripan Kalimat Menggunakan Syntactic-Semantic Similarity pada Sistem E-Learning. *KURSOR*, 5(2): 75-82.