The Manual for “Native or Not”

A Project for CS385

JIAMING SHEN
ZHAOWEI TAN
ZHICHEN WU

Shanghai Jiao Tong University
IEEE Honor Class

Contents

1 Introduction

Hello there! Thank you for voting for us and welcome to our project! We are ***Team JeeQee O'ye***, constituted by three people: ***Jiaming Shen***, ***Zhaowei Tan*** and ***Zhichen Wu***.

In this section, we will give a brief introduction on our project, including why we came up with the idea, the expected goal of this project and the practical significance of the work.

1.1 Motivation

Many of us will apply for graduate schools in the near future. TOEFL score is indespensible for this process, and a higher TOEFL score may increase your chances of being admitted. However, Chinese students tend to earn a low score in Speaking Task, which prohibits some top students from reaching the minimum requirements of their dream schools.

From this point, we developed our idea. We aim to build a system, via machine learning methods, to help us train the accent in order to achieve a higher TOEFL score. Considering that this goal is way too far from us, we start from a simple task, which will be described in the next subsection.

1.2 The Basic Goal

Our basic goal is a binary classification problem:

> ***Given a recording of a fixed script of English words, we would like to predict whether the speaker is a native English speaker or not.***

There exist some potential advanced tasks, which will be discussed in the later section.

1.3 Significance

Upon the finishing the project, the outcome can be used to:

- assess your accent
- judge the native language

The function will be useful for the following people:

- ESL teachers and students
- People (like actors) who need to learn an accent
- Linguists who do research in this area
- Computers whose services differ due to native language
- Anyone who finds this interesting

2 The Dataset

2.1 Source of the Data

The recordings are scraped from the George Mason University Department of English Speech Accent Archive *(http://accent.gmu.edu/)*. We do not own the data, and all the credits of the data go to that entity.

2.2 The Discription of the Data

In the website, all the recordings are classified according to the speakers' native languages.

amharic
antigua and barbuda creole english
anyin
appolo
arabic
aramaic
armenian
aromanian
ashanti
dutch
eastern farsi
ebira
edo
efik
english
esperanto
estonian
ewe
igbo
ikalanga
ilocano
ilonggo
indonesian
interlingua
irish
irish gaelic
italian

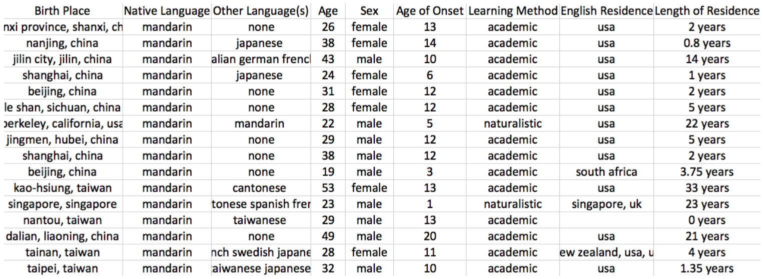
Figure 1: The snapshot of the website

We now have crawled the recordings of two categories, *English* and *Mandarin*, combing 620 clips together. More categories can be crawled by yourself further, which will be elaborated in Section 2.3.

In each recording, every speaker will read the same English script:

> *Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station.*

In addition to the recording, we scrap the basic information of the speaking, including the native language, birth place (pinpointed to the province), other language spoken, age, sex, age of starting learning english, learning method, english residence (if has, the length of the residence). These data are collected into csv files which have been provided together with the recordings. You can read them using various programming languages.



Birth Place	Native Language	Other Language(s)	Age	Sex	Age of Onset	Learning Method	English Residence	Length of Residence
nxi province, shanxi, ch	mandarin	none	26	female	13	academic	usa	2 years
nanjing, china	mandarin	japanese	38	female	14	academic	usa	0.8 years
jilin city, jilin, china	mandarin	alian german frencl	43	male	10	academic	usa	14 years
shanghai, china	mandarin	japanese	24	female	6	academic	usa	1 years
beijing, china	mandarin	none	31	female	12	academic	usa	2 years
le shan, sichuan, china	mandarin	none	28	female	12	academic	usa	5 years
berkeley, california, usa	mandarin	mandarin	22	male	5	naturalistic	usa	22 years
jingmen, hubei, china	mandarin	none	29	male	12	academic	usa	5 years
shanghai, china	mandarin	none	38	male	12	academic	usa	2 years
beijing, china	mandarin	none	19	male	3	academic	south africa	3.75 years
kao-hsiung, taiwan	mandarin	cantonese	53	female	13	academic	usa	33 years
singapore, singapore	mandarin	tonese spanish frer	23	male	1	naturalistic	singapore, uk	23 years
nantou, taiwan	mandarin	taiwanese	29	male	13	academic		0 years
dalian, liaoning, china	mandarin	none	49	male	20	academic	usa	21 years
tainan, taiwan	mandarin	nch swedish japane	28	female	11	academic	ew zealand, usa, u	4 years
taipei, taiwan	mandarin	aiwanese japanese	32	male	10	academic	usa	1.35 years

Figure 2: The snapshot of the csv file

2.3 The Crawler

We have provided our own crawler. Given that the number of recordings is lopsided between Mandarin and Native English, you can modify the crawler to get more recordings from other mother languages and use that as your non-native English dataset.

Two crawlers play different roles. ***music.py*** crawls the recording file, while ***metadata.py*** crawls the metadata of the speaker. You can modify them to get your dataset.

You can browse different languages in *http://accent.gmu.edu/browse_language.php*. After your choosing one, you can paste the language name (e.g. arabic) to replace ***the first "english"*** in the 5th line of ***music.py*** and the ***"english"*** in 15th line of ***metadata.py***.

After designating the source of the data, you can change the file name by altering ***the third "english"*** in the 5th line of ***music.py*** and the ***"english"*** in the 89th line of ***metadata.py***.

Finally, do not forget to change the target path where you store the data. The csv file will be store in the same directory by default and you can as well alter ***the second "english"*** in the 5th line of ***music.py*** to change you folder to store the recordings.

3 How to Start

In this section, we present a simple documentation of our startup codes, including the methods of importing data, compressing audioes(optional), and using mfcc to extract features.

3.1 Directionary Structure

– Start Codes
— English
—— english1.wav
—— english2.wav
—— ...
— Mandarin
—— mandarin1.wav
—— mandarin2.wav
—— ...
— main.m
— readData.m
— compressData.m
— featureExtract.m
— readme.md

3.2 Components

- ***readData.m***

  First, you should put all your data in a separate foler and name them in a structural and meaningful way. You can import and store all data in a cell array by giving the name of that folder, the specific type of your audioes ('mandarin', 'english', etc), as well as their suffix ('.mp3' or '.wav').

- ***compressData.m***

  This part converts auidoes into sound tracks of approximately the same length. We do this for the sake of later feature extraction part, however, this part is totally optional. We are not sure whether this step will alter some important features of original data, and thus if you are afraid of its possible side effects and want to skip this part, be my guest.

- ***featureExtract.m***

  This part use 'melfcc.m' in 'rastamat' toolbox, so You MUST add 'rastamat' directionary into the working path. The input of 'melfcc.m' is waveform 'x', its sampling rate 'fs', and a lot of optional parameters. For more detailed information of 'rastamat' toolbox, see resource-1. Besides, we provide a basic tutorial of mfcc, see resource-2.

  P.S. For the baseline system, we use Matlab neural-network toolbox, and for its tutorial please refer to Section 4.1.

4 Some Tips

4.1 Additional Resources

You can harness the following information to finish this projects:

- Rastamat toolbox documentation

  http://labrosa.ee.columbia.edu/matlab/rastamat/

- MFCC tutorial

  http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

- Matlab neural-network toolbox tutorial

  http://cn.mathworks.com/help/nnet/gs/fit-data-with-a-neural-network.html

- The Munich Automatic Segmentation System MAUS

  http://www.bas.uni-muenchen.de/Bas/BasMAUS.html#downloads

4.2 Advanced Tasks

Here are some advanced version of this project:

- Instead of judging native or not, we can judge which the native language of the speaker is, thus presenting a multi-classification problem.
- We can using the recordings to predict the onset learning year of the speaker, which stands a regression problem.
- Moreover, we can extract the vowels and the consonants in the recordings and use that to predict native or not in any given recordings, breaking the shackles of a given paragraph.

4.3 Contact

If you have any problems or suggestions, you are more than welcomed to contact us. You can either find us in class, or send us e-mails. Please mail to tanzw94@gmail.com.

Thank you again. Enjoy machine learning!