CIS131 Final Project

Christopher Nowacki

May 16, 2023

1 Project Overview

TotLingo is a vocabulary analysis tool designed to track, analyze, and provide insights into the vocabulary development of toddlers as they are learning to speak. The program will support a profile system to allow parents/caregivers to track multiple vocabularies. Once logged into a profile, the user has the option to enter speech data or enter an analysis menu, which will present them with several different vocabulary reports for determining the size and other characteristics of the toddlers vocabulary.

2 Features

1. Profile Management: Users can create, remove, or log into individual toddler profiles, allowing for personalized tracking and analysis.

2. Speech Data Entry: Users can input speech data in the form of text strings, which are then processed and stored in an easy-to-read text file. Each word entry includes the word itself, the number of times the word has been used, and the dates of first and last usage.

3. 3. Vocabuluary Analysis: The program offers several options to view analysis of the vocabulary data, including word frequency, most recent words added, most used words, and the option to input a specific word and view statistics on that words usage.

2.1 Required Features

1. Speech Input: The application must provide users with a simple and intuitive interface to input speech data in the form of text strings which are then processed and stored.

2. Users should be able to get analysis of vocabulary data and statistics.

3. Information should be presented in a user-friendly and readable format.

4. Profile Management: The program must allow users to create, remove, and log-in to individual profiles. There should be safeguards to make sure profiles aren't accidentally deleted.

2.2 Python Programming Features

1. Object-Oriented Programming (OOP): The application will use classes to keep related functions grouped together and for modularity.

2. String Methods and Formatting: This program will take advantage of Python's string methods and formatting – such as split() to break speech phrases into tokens.

3. Menu systems will be loop-based and use conditional statements.

4. The program will use Python's built in file functions for reading and writing data to .txt files.

2.3 Technology Used

The primary technology used in Python. The program will use the **os** module to interact with the operating system when creating and managing profiles. It will also use the **datetime** module for tracking when a word was first added and last used. The program will use classes for modularity and reusability.

3 Implementation

1. 1. Text formatting and tokenizing class: A Python class, named 'VocabProcessor', will be created to handle the formatting and tokenization of text strings. This class will contain methods for cleaning and

transofrming the input text, such as converting text to lowercase, removing punctuation, and splitting the text into individual words. The 'VocabProcessor' class will be responsible for ensuring that the speech data is in a suitable format for further analysis and storage.

2. Loop-based Menus: Menus will use while loops and conditional statements to display available options, accept user input, and execute the corresponding modules.

3. Data storage in trackedvocab.txt: The program will store vocabulary data in a plain text file in an easy-to-read format. Each line in the file will represent a unique word entry, with data fields separated by a delimiter (e.g. ','). The fields will include the word itself, the date the word was first added, the date of the last entry for that word, and the number of times that word has been accessed.