Speech-based Interactive Location Guide for Liivi 2

Imre Purret **Päivi Kristiina Jokinen**
Imre.Purret@gmail.com kristiina.jokinen@ut.ee

Abstract for demonstration

1 Introduction

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. Many computer operating systems have included speech synthesizers since the early 1990s. [1]

2 Required programs

Current program is written in Python 2.7 (32bits) and used on Windows operation system and also tested on Mac OS X. It also needs couple of extra modules for Python, one is pyttsx, which generates speech from text. Running pyttsx also needs Mark Hammond's win32com package on Windows.

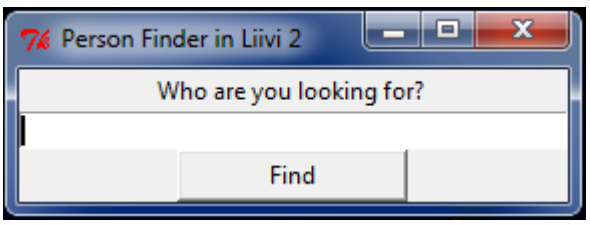
3 Knowledge base

For getting information into the program we created a couple of text files: PersonRoom.txt which contains information with person name and their room number and RoomLocation.txt which contains information with room number and short information how to get into that room.

We also use the third file RegularExpression.txt, what will be generated by program. This file contains information with person name and regular expression how to ascertain that it is right person.

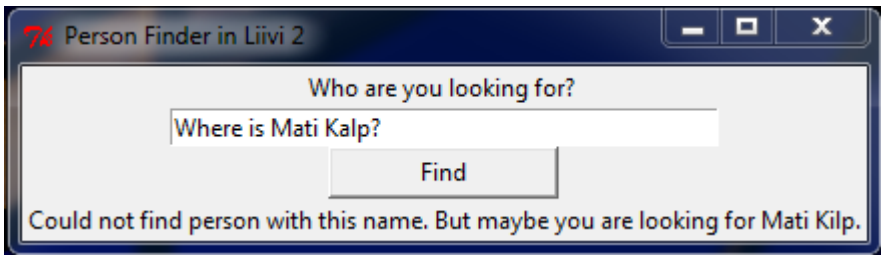
4 Program

Program checks if file RegularExpression.txt exists, if not then creates it. For creating it uses information from file PersonRoom.txt, mainly information about names.
If RegularExpression.txt exists then checks if it's up to date, if not then updates it. If yes then sets up the pyttsx engine and reads in files with knowledge. After that it generates a Tkinter window and starts waiting for the user input.
Program sets up the pyttsx engine, reads in the text file about knowledge and generates a Tkinter window. After that it will be waiting for the user text input.



After getting an input it starts checking if the input matches to any of the regular expressions about persons given in file . If no match was found then it calculates all Levenshtein distances[2] from all ordered word pairs given in input to all person names given in PersonRoom.txt file. It finds smallest distance and selects person who had smallest distance with two words in input and program says: "Could not find person with this name. But maybe you are looking for " and adds name what has closest distance. Same information is also shown on the Tkinter window as well. Then program moves to

second state what first recognize if user confirms that it's correct person if not then starts for looking again. If confirms then it generates the route to the room using the knowledge from the text files and gives the route as a speech input and also as a text input to Tkinter window.



If the input string matches to a regular expression then it generates the route to the room using the knowledge from the text file and gives the route as a speech input and also as a text input to Tkinter window.

Before checking if input string matches to a regular expression I replaced Unicode characters with ASCII code characters, because in English there are no such letters like "Ü", "Õ", "Ö" and "Ä". So for regular expression check I replaced "Ü" with "Y", "Ö" and "Õ" with "O" and "Ä" with "A". Same with pronunciation when giving a speech output from program.

5 Conclusion

In general speech recognition is popular topic which is improving fast and there are many well known programs using it. One of the most famous ones is Siri an intelligent personal assistant and knowledge navigator which works as an application for Apple Inc.'s iOS.[3] But it still seems to be a hard topic to work on for getting good results.
We can see that creating synthesized speech is applicable without a lot of trouble. With creating knowledge base and using some algorithms it is possible to get considerable results with it.

Reference

1. Allen, Jonathan; Hunnicutt, M. Sharon; Klatt, Dennis (1987). From Text to Speech: The MITalk system. Cambridge University Press.
2. http://www.levenshtein.net/ 2014-05-08.
3. http://en.wikipedia.org/wiki/Siri_(software) Retrieved 2014-05-08.