SPEAK PSEUDOCODE2C: A framework to convert customized pseudocode to c code

Raghav Aggarwal(PES1201800312) | Rajdeep Sengupta(PES1201800144) | Shaashwat Jain (PES1201802346) | Srishti Sachan (PES1201802126)

Project Guide: Prof. Nitin V Pujari

Department of Computer Science and Engineering, PES University.

Problem Statement

Speak Pseudocode2c :
A framework to convert customized pseudocode to c code

In today's world, there is very less support to help students understand the programming concepts better. The c code given in textbooks is empirically observed to be not very easy for students to comprehend at the school level when compared to natural language pseudocode. The new education policy has mandated the course of basic programming for students in junior classes.
Therefore, our goal is to use voice-based digital assistant to automate the process of writing and editing code by providing our voice input.

Background

• A study done by P. Sirikongtham and W. Paireekreng uses speech recognition using dynamic multi-pipeline API. This technique combines the powers of two of the market's best speech-to-text conversion models being Google Speech to Text and Microsoft's Speech Conversion model. There is a massive 21% increase in accuracy than the Microsoft's API and 4% increase from Google's current engine.
• T. Dirgahayu, S. N. Huda, Z. Zukhri and C. I. Ratnasari, in their paper, try to solve the problem to generate pseudocode from given source code. For this, they built a neural network with the help of Long Short Term Memory architecture. They used Bilingual Evaluation Understudy(BLEU) score to measure quality of generated pseudocode which was 54.78%.
• Imam, Ayad & Alnsour, Ayman introduce Neural Named Entity Recognition technique from subword units. The model learnt from low-dimensional representation from each unit of word which is then concatenated and fed into bidirectional LSTM-CRF model. They took portrayals from every one of three subword units to interpret what they represent in converting pseudocode to source code.
• Another paper talks about the approach of trying to convert and edit pseudocode to source code in multiple languages using XML and features of DOM(document object model) by encapsulating features of each line of code in tags like <var>, <var-name>, <var-type>, <var-value>. The author mentions the use of <for>, <while>, <do> tags for loops where nesting is present.

Design Approach

We have used incremental model methodology for our project.
The design constraints include:
• Google Cloud free tier has a limitation for Speech-to-Text API which we are using in the development phase of the application, it is free for 60 minutes/month after which it will use the credits.
• The hardware requirements for the system are minimal, although there is a requirement for a microphone and speaker with a working internet connection.
The major assumptions are:
• The user is familiar with the general pseudocode format.
• The user has a working internet connection.
• The host machine has gcc compiler installed for the execution of the c programs along with python3 and google cloud python library which are necessary for running the framework.

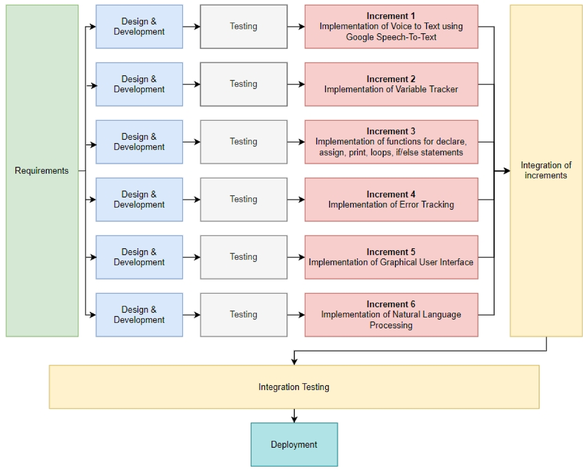
Requirements

Hardware Requirements:
• Microphone with updated drivers.

Software Requirements:
• gcc compiler version 9.3.0 or above should be installed on host system
• Python 3.7 or above installed on the device
• All python packages in requirements.txt file need to be installed (*pip install -r requirements.txt)*
• tkinter must be installed

System Requirements:
• Any Linux Distribution or Windows 32-bit/64-bit
• Good internet connection with speed 1MBps or above

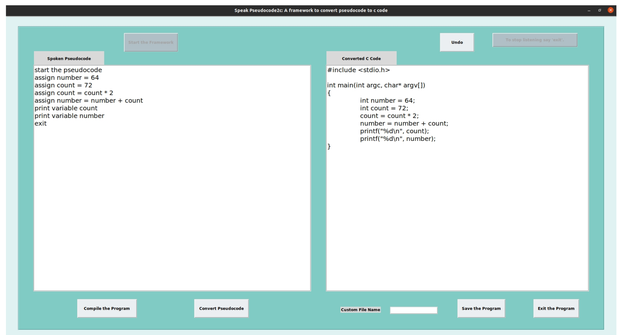
Other Requirements(for better performance):
• No background noise on the user's end for better performance
• Performance requirement: reliability of framework depends on google cloud and user defined libraries for conversion
• Safety requirement: framework provides built-in protection for all user files as it is reliable on google cloud
• Security requirement: confidentiality and integrity provided as per the privacy statement by google cloud services
• User files can't be touched by any other user through this tool



Incremental Model

Results and Discussions

The testing activities include unit testing and integration testing for the framework. A total of 95 test cases were written for the same. The non-functional testing is carried out by third-parties like Google Text-To-Speech and python modules. The test environment was setup using pytest, an open source tool. This being an automated test, makes the future testing easier while deployment.
The results indicate that all the expected deliverables are working successfully.
• Converting speech input to c code through our mapping libraries and their unit and integration tests.
• Building a GUI for the framework using tkinter as a base in python.



Graphical User Interface of the Framework

Summary

The main goal is to achieve line by line conversion of pseudocode to C language along with accuracy. This is achieved by google speech to text models on google cloud platform. The pseudocode in text format is converted to C language using Natural Language Processing and custom libraries. Once the conversion of pseudocode is complete, the framework supports extra features like saving the code, compiling the code and also undo button in case of some error in speech to text conversion.

Conclusions and Future Work

The finalized framework is complete from development and testing point of view. The line by line conversion of pseudocode to c code is successful and features like saving and compilation of code is also implemented. The unit and integration testing is implemented using pytest module. The graphical user interface is designed using tkinter module.
The future scope involves the support of functions and strings and making the framework more robust. Presently, our framework only supports C language but in future, it can be extended for other languages like Java, C++, Python etc.

References

• T. Dirgahayu, S. N. Huda, Z. Zukhri and C. I. Ratnasari, "Automatic translation from pseudocode to source code: A conceptual-metamodel approach," 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), Phuket, Thailand, 2017, pp. 122-128, doi: 10.1109/CYBERNETICSCOM.2017.8311696.
• Imam, Ayad & Alnsour, Ayman. (2019). The Use of Natural Language Processing Approach for Converting Pseudo Code to C# Code. Journal of Intelligent Systems. 29. 1388–1407. 10.1515/jisys-2018-0291.

Project Guide:
Prof. Nitin V. Pujari
Dean IQAC, PES University

Team Members:

Raghav Aggarwal

Rajdeep Sengupta

Shaashwat Jain

Srishti Sachan