Title: Exploration of Spoken Programming Languages

Intro
Currently dictation can be, and is, used by people to take notes and write essays. However, it does not work very well for writing code. This project will investigate how best to change this. We know from studies conducted in the past that speed of speech is far higher than typing for the majority of people (I will find and add a reference here). It has also been shown that being sedentary and typing for long periods of time is not only an RSI risk but has also been linked to reduced life expectancy (another reference).

There does exist software that can be used to dictate code. However as far as I have been able to determine, this software forces the user to speak in an unnatural way. For example, the user still has to specify punctuation characters in the code, such as brackets and commas. When we are reading out regular text we do not directly specify the punctuation that is written in front of us, but instead it is inferred by the listener. Why should dictating a computer program be any different?

During the course of this project, I will define a pseudocode like language (which is both Turing complete and unambiguous). Alongside this, I will create an application which takes dictated English as an input. The semantics of the input would be that of a programmer explaining a function in the most natural way to them. It will function as if one person was explaining to another person how to code up the function in a pair programming session.

Work to be done:
- Further exploration and evaluation of current solutions to see if there is any common themes which I can integrate into my project.
- Deciding how the spoken examples will be gathered, the main issue being that, in the preliminary pilot study the prompt presented to the respondents influences the way in which they choose to specify the algorithm/function.
- Running the surveys with participants to gather the data for the project to use in both training models and during evaluation.
- Taking the dictations from the surveys and turning them into textual transcripts so that they are easier to process by the system.
- Identify the libraries which will be required and familiarising myself with them and also confirming the syntax of the pseudocode which I will use (which must be an unambiguous context free grammar).
- Write the module which generates the Abstract Syntax Tree (AST) of the code from the spoken transcripts using Traditional NLP methods.
- Write the module which will take the AST and then from this, produce the pseudocode representation of the AST.
- Write the module which will use Statistical Machine Translation(SMT) techniques to create the pseudocode.
- Evaluating the project.
- Writing up the project.

Starting Point.
I have already undertaken a pilot study to explore methods for collecting an appropriate training corpus. I have briefly looked at current solutions such as https://voicecode.io/. I did this to determine firstly if they solve the same problem and secondly if I could learn anything from them.

Structure of the Project
The project will be made up of 2 main modules:
- The first will take in text transcripts from the survey and process them using statistical machine translations to map the input to a AST. From this AST a pseudocode will be generated.
- The second will use Statistical Machine Translation to map the transcripts to the pseudocode.

(I personally feel that the most natural way to do this would be with a labelled flow diagram, is that permitted in the the proposal or not?)

Success criteria
- The success of the project will be judged based on how many edits are required to make the code work as the dictator intended.

- We can baseline this against a simple bag of words approach. This will involve:
  - Filtering the transcript files to extract variable names ect
  - Creating the most probable re-ordering of the words in the transcript based on an n-gram model of the pseudocode.

Plan of work
The Project will be split into 10 two-week work-packages:
1. Evaluation of the currently available solutions.
2. Confirming exactly how the data collection will be carried out, including the prompts which will be presented to the respondents.
3. Running the surveys to gather data for the project and turning the spoken words into typed transcripts. As well as, finding and familiarising myself with the libraries that will be required to process the data as well as defining the pseudocode language that I will use for the output.
4. Writing the first of the modules listed in the Structure of the Project section.
5. Writing the second module specified in the Structure of the Project section.
6. This period will be used to continue you work on which of the 2 previous modules need more time.
7. Improving the module written in the previous 2 weeks to improve performance.
8. Evaluating the system including creating a baseline to compare the project too.
9. Writing the dissertation.
10. Continue to write the dissertation.

Resource declaration
I plan to use my own computer, (2.9 GHz CPU, 16 GB RAM, 1TB Flash Storage, macOS Sierra).