Research Assessment #11

Date: 7 February, 2020

Subject: Natural Language Programming and Machine Learning

APA citation:

Tomassetti, F. (2019, May 8). How would I go about creating a programming language? Retrieved February 7, 2020, from https://tomassetti.me/how-to-create-programming-language/

Assessment:

This week I researched how to create my own programming language. My recent research into Natural Language Programming and a recent conversation with a friend made me temporarily shift my focus regarding my final product. My original plan was to create a program that would enable the conversion between natural language instructions and code. Although this still remains my end goal, I have decided to first build my own programming language. At this point, I believe this should be an easier task than programming my conversion algorithm since a programming language would have a more rigid syntax and would not require Machine Learning integrated. By first building my own language, I will be able to first understand the intricacies of programming languages, information that will be essential in the development of my end product. By first developing this alternate product, I will also be able to assess the feasibility of my original goal. If building said language takes a long time and proves to be a sufficiently difficult task, it will mean that my original goal was probably not feasible.

From the article, I learned the important steps of making a language and how these steps would differ for me. In general use, a lot of thought goes behind the functionality of the language. Questions such as whether to make it very fast or to make it memory efficient. or whether to make the language imperative, instructions based, or functional, process or algorithm based, must be answered first before the actual language can be designed. For me the answer to these questions was simple. I want to design an imperative language, based on existing languages like Python and HyperTalk. I believe that an

imperative language is the most user friendly, and it is also the type I am more familiar with. I also want to enhance the language for readability as compared to time or space complexity, as the language would be meant for beginners. The one large question that remains is whether I want my language dynamically typed or statically. The difference lies in whether I want variables to have a strictly defined type, inputted by the user, or have a general variable that can store any type of values. The answer to this question lies in the language I decide to use for my interpreter. If I use a static language like Java for the interpreter, I will create a static type language. If I decide to use a dynamic language like Python, I will create a dynamic language.

Another aspect would be the actual development of the interpreter or compiler. This is supposed to convert the syntax into another language. This could technically be written in a language different from the target language, which is to say that the interpreter could be coded in Python, but used to convert the source code to Java. This part of the code would be the most essential towards my original plan, as it is the embodiment of my conversion algorithm, albeit in a non Machine Learning form.