Project Proposal

Team Members: Reynerio Sarmiento, Gursimar Singh, Sajana Wijesinghe, Bryon Ferrer, Igli Ajdini

Verbotics is an up and coming medical analysis business composed of several people developing an engine used to analyze medical reports from a doctor or nurse to find out how much the insurance company has to pay for the patient’s treatment using billing codes. Our duty is to implement an efficient and easily accessible search and edit part of the engine. It may seem like a very difficult task, but our approach to such a daunting issue is nothing short of revolutionary. First, we would implement a search tool on that site to let the user search for billing codes without having to open up a new tab on his or her browser and look for what billing codes are associated with which parts of the report. This is done by having the search tool link to an official billing code database and have the user type in either a specific code or keywords that relate to that code, similar to what the Google search bar does when a user types in something in its search bar. The user can then add it to the billing code list by clicking on the code and the engine would then search for keywords in the report that relate to that particular code, if any. That way, we would eliminate the hassle of reporting incorrect and inaccurate billing codes due to some codes having insufficient evidence of its existence in the list, adding more to the medical costs for treatment than usual. Second, we would add new editing tools to let the user edit the billing codes reported by the engine’s underlying artificial intelligence in a proficient manner. This is done by letting the user extend vague billing codes to make them more specific, as more specific billing codes would let the patient know more about why he or she is charged as much. We will let the user type in more specific information to that vague billing code and the code will change depending on what the user puts in. Third, we would let the user assess the performance of the engine in analysing and encoding using a debug mode. With this mode, the user can input billing codes resulting from their analysis of the report and compare it with what the engine has. The mode would highlight those that are similar and
italicize those that are kind of similar, but the engine doesn’t have the capability to extend its analysis to make the code more specific. That way, the user can make sure the back-end of the engine itself is doing its job properly as a helper in diagnosing the billing code problem plaguing patients and hospitals alike.

A lot of principles from Computer Science are very much applicable here. For instance, the billing code “search bar” is an example of an autocomplete problem, which would have to require certain algorithms to make sure the autocomplete is as efficient as possible. This can also be an example of using search queries from database to make sure the results equal the user inputs. Otherwise, the billing code can end up disastrously inaccurate in specificity. Editing the billing code results to make them more specific would be an example of using trees as data structures, since the user is being given a base and adding keywords allows the user to go further and further into the tree to find the code that is closely associated to what he or she is typing. As a team, we would want to make sure that the search and editing tools are easily accessible to the user when it comes to look and feel so that he or she has no trouble when it comes to correctly identifying and analysing medical reports with the help of artificial intelligence. We would also want to make sure that search algorithms and queries would take such a short time to execute, which would be a dive into runtime analysis and polynomial and non-polynomial complete problems. We hope that using these concepts to create these additions would make the engine more versatile and user-oriented than ever before and, as a result, be a more well-rounded engine to be used by hospitals worldwide.