A Big Data Approach to BIM Models

CHRISTOPHER BEORKREM
University of North Carolina at Charlotte

FERNANDO J. CLAUDIO RODRÍGUEZ
University of North Carolina at Charlotte

ALIREZA KARDUNI
University of North Carolina at Charlotte

ASHKAN RADNIA
University of North Carolina at Charlotte

Keywords: Architecture,BIM,Data,Computation, Visualizations.

As technology develops at an astounding rate, data has become an intrinsic by-product on major fields, including architecture. Providing designers, contractors and other stakeholders with additional information in order to make informed real time decisions. This paper describes the creation of a series of prototypes which are intended to convert an archive of Building Information Models (BIM) from Little Diversified Architectural Consulting, an early adopter of Autodesk Revit, into a database which can be analyzed using Big Data approaches to data analysis. Using python, as opposed to Revit's native C# language, provided a more dynamic and simpler pipeline for connecting to the underlying database behind Revit. Use of non-relational databases proved a more efficient system for managing all the information produced by the models. Throughout the use of these tools, we were able to generate our prototypes, which can be divided into three main sections. Starting with altering data inside a Revit model, allowing us to manipulate the databases and add new information as needed. Secondly we collected data between multiple models and stored all the information in a single database. Lastly is visualizing the obtained data from models in order to represent, compare and evaluate the data. After running multiple tests on the models, results proved to reduce repetitive tasks on traditional workflows, while allowing for better analysis and understanding of the implications and processes we use as designers in the 21st century.

INTRODUCTION

Data has quickly become one of the largest and most valuable by-products in history and has revolutionized the way the world collects, develops and uses information. Industries such as automotive and motorsport analyze all the data retrieved from sensors in their vehicles to maximize the efficiency of their products and racers to gain an advantage from the competition. Therefore, "increasingly, the value of a business is tied to its ability to mine data"[1] and the AEC industry is quickly becoming one of the largest data producers in the world, as building design and production becomes more automated and more building elements are sensored. This information is becoming essential for clients to better understand how their buildings are performing, ranging from simple things like temperature statistics on thermostats, to counting the number of times doors are being opened and closed, to complex computer vision monitoring of behaviors. Alternatively, regulating the controls of HVAC systems, while monitoring heat gain and loss through this process and understanding the circulation and use in the building across the day, only helps to fix issues once the building is occupied. Data created during the design process can help to back up design estimates even after construction is completed, potentially providing live comparisons from environmental analysis to actual environmental data.

From an economic perspective, data can help manage construction costs and budgets, during and after the design process, providing cost and time estimates, grounded in real time data, which can be compared as the project progresses. During the construction phase, data produced on site can provide up-to-date information that can help contractors, construction companies and stakeholders make informed on-time decisions [2]. As technology continues to progress at an astounding rate, it is time for the architecture field to catch up to these advances and take advantage of them. Building Information Modeling (BIM) tools such as Autodesk Revit have revolutionized the industry becoming the standard for architectural production. At the same time Revit is generating an astounding amount of valuable data. This paper will describe the creation of a series of prototypes which are intended to convert an archive of BIM models from a large Architectural Design firm (Little Diversified Architectural Consulting) into a database which can be analyzed using Big Data approaches to data analysis.

Given the current intrinsic value of data, the goals of the project consist of creating a standardization of information and automatizing large repetitive tasks. For example, an analytical benchmarking task that normally could take multiple employees, weeks to complete, can now be performed in a matter of minutes. This helps demonstrate the power and benefits of using this information. Knowing this, the value and use of this data is greatly expanded, especially when analyzed across hundreds, or even thousands of models. When we can quickly access and compare a huge range of models we can begin to demonstrate some of the capabilities of harnessing the power of big data for Architecture.

DEVELOPING REVIT ANALYTICS

Over the last 18 months we have worked with Little, an early adopter of Autodesk Revit to develop an understanding of

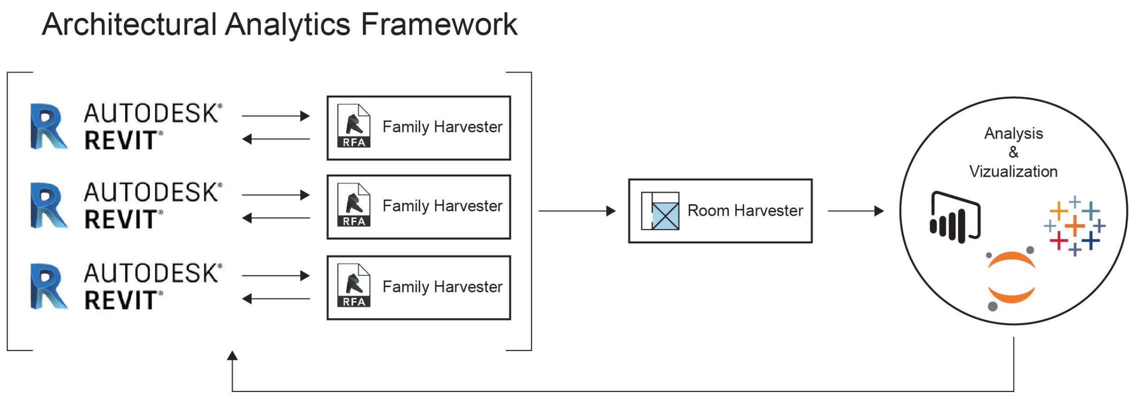

Figure 1. Architectural Analytics Framework of the multiple prototypes.

what information might be available within their archive of approximately 300 design models (an initial data set). This archive (as those with most firms have) were modeled using an evolving subset of standards which have been loosely adhered to within the firm structure. Each model has a widely varying number of backups which were archived during the modeling process. This results in a large, potentially rich, yet unstructured data set.

We have been working to develop a process by which we can harvest particular data points within these models and add parameters to their definitions to be able to subsequently use them for creating benchmarking standards for internal office processes as well as programming and design of future projects. These tools will allow for each model to be standardized and then easily mapped using typical database automation tools to better understand past programming standards for a program type (office, retail, etc.).

Secondly, we have worked to develop a tool which will strip each of these models of their unnecessary data and create models which will have a uniformly searchable set of standards regardless of how, what version, or when they were modeled. This tool can be used to create a uniform data set which can represent both past and future work of the firm. These uniform inter-project data sets can be used for analysis using simple data analytics and interactive visualization tools. With more complex tasks and inquiries, these datasets can be analyzed using advanced machine learning and artificial intelligence algorithms.

One of the main benefits of BIM is streamlining the architectural design process towards efficiency, but these models also contain a wealth of information that could be harvested and analyzed by AEC firms. However, there are various reasons for architectural firms to need to go beyond data collection and analysis to actually programmatically alter and update data on BIM models. The first reason is the fact that in architectural firms, data is created and named inconsistently. Family parameters and naming conventions vary widely even with the most rigorous of firm standards. This inconsistency is problematic, as this would mean that all of these similar rooms would be treated differently by the analytical tools. Moreover, it would be desirable to provide a method for architects to update their Revit model data, based upon the outcomes of the analytics. To this end, in order to have a complete pipeline, we propose a framework that includes an iterative process of data collection, analysis, and alteration. Fig.1 shows a diagram of the logic of our proposed framework, this section will describe the development of a set of tools that would fit within this framework.

Revit has a built-in database exporter- Open Database Connectivity (ODBC), which creates a simple self-referenced database of all the elements in a given model. At their core the databases allow for more typical approaches to data analytics, but they are quite large and have object IDs which are often duplicitous. The default exports are then rendered more complex by the often-changing standards used to tag objects within the model by users or manufacturers.

As an alternative, the method we employed to create these prototype databases used PyRevit, which as the name implies is a collection of tools for developing plugins for Revit using an implementation of the Python language into the Revit ecosystem by Ehsan Iran-Nejad. As opposed to Revit's native C# language; python, which is currently the leading programming language for data science and analytics, provides more dynamic and simpler pipelines for connecting to the underlying database behind Revit. PyRevit streamlines the process of creating UI elements within Revit, extracting data, and altering and updating data within Revit projects.

In order to allow for a wide range of analytics using a wide range of data and visualization methods, our Revit add-ons extract data from multiple models at once and organizes them within a single database. This is made possible by utilizing a non-relational database as the backbone of the system.

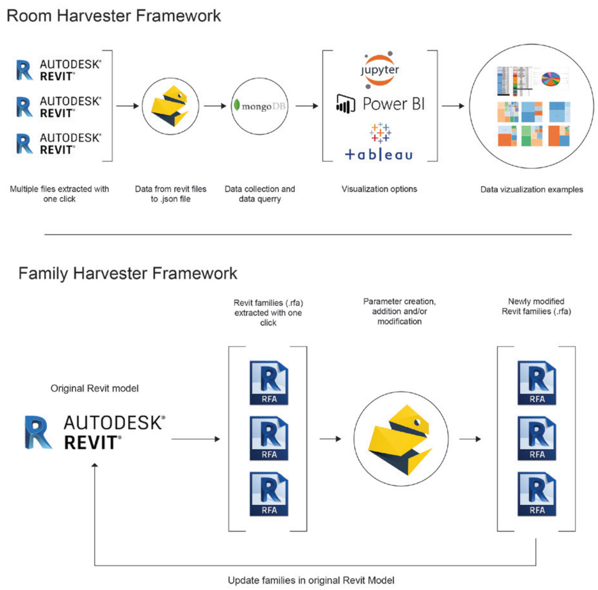

Figure 2. Framework for the Family and Room Harvester

Non-relational (NoSQL) databases are best suited for datasets that are not structured like a table. NoSQL databases are structured in terms of documents and collections as opposed to tables and rows in relational datasets. As an example, within Revit families, each type of room may have different data types, names, attributes and values, these types of databases are most suitable for NoSQL. In our case, we utilized MongoDB which is an open source database designed specifically for non-relational data and has one of the most useful set of read, write and delete operations[3]. In our system, each room is a document.

In this example, we worked to create a test bed for analyzing the data we were able to harvest. We gathered some simple information from five example Revit models- Rooms and Room Tags. Rooms are physical elements inside the Revit environment, like walls and doors, they have perimeter dimensions and areas. Room Tags are annotation elements that are added after rooms are created and define the area by its boundary elements, such as walls or room separation lines. The reason behind choosing these elements is because they contain different parameters, some which are dynamic and computed based on the room's spatial properties, such as area or volume. While others are defined by the operator, for instance, occupancy type and department. Extracting room data is traditionally done using room schedules and adding the specific parameters to produce the data table. Our system allows us to gather both the physical and associated spatial qualities and quantities within a single database. We intend to add in a method for using natural language processing tools to more fully automate the renaming process in the future.

Our goal for developing this tool was to make the process of collecting, extracting and visualizing data fast and easy. All with the push of a button within the Revit interface. Given how some documents can have very large file sizes, manually browsing multiple files can be a very slow process. The power of this tool lies is the ability to process multiple Revit projects automatically, extracting data from all selected data points within each project and saving the resulting data into a database. Moreover, removing the need of going through the repetitive process of creating schedules inside each Revit document. After extracting all existing parameters for all rooms in multiple projects, these results are saved into a single database. This database becomes the powerful tool that we can use to analyze the historical value of a large quantity of Revit models at once.

However, room tags are not limited to Revit's default parameters as they can be compounded with other features of Revit's API to store the data of other elements that exist within the room or identify the walls and windows which are attached to the room boundaries. This feature allows us to expand the range of data we can extract from a Revit document and make room harvester aware of all the existing elements within the room and maintain parent/child relations between a room, and its respective elements and therefore with each of those elements other relations. The standardization of parameters can be mapped to a bigger spectrum of Big Data within a firm, allowing for benchmarking standards and design alternatives for future projects.

ALTERING DATA ON REVIT MODELS:

Architecture firms, very often have multiple people collaborating on large projects inside a single Revit model. Although most firms maintain a template standardization for their naming conventions and parameters, this is not always accurate. Employees do not always follow these standards and tend to use similar elements and naming conventions for various purposes. This causes inaccuracies and inconsistencies across models and data. This tool provides a streamlined process for any firm, to add and/or modifying a parameter to multiple elements to maintain or create consistency across all models and databases.

In order to understand in depth the use of this tool and its capabilities, it is necessary to describe key components in the use of the data embedded in the Revit environment. Starting with the use of "families" which are 2D or 3D elements inside a project which houses a series of information that serves to represent a discrete building or documentation. Each one is saved as a separate file(.rfa) which can be loaded into a project. These range from furniture, doors, and windows to plumbing and lighting fixtures. Based on the properties housed by each family, these are organized into categories that Revit uses to perform certain calculations.

To test our process and classification system, we have developed a tool that consists of adding a standardized parameter across all furniture family files inside a Revit document. Currently occupant loads are determined based on square footage, however these are not entirely accurate. A precise number would be better determined by calculating the maximum

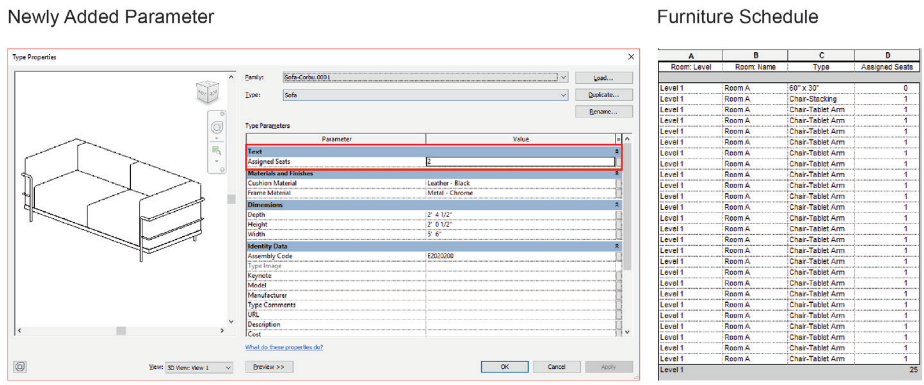

Figure 3. Revit interface displaying family with newly added parameter and furniture schedule

capacity of people which can use a given space. Using the family Harvester tool, we were able to collect all the furniture across the project and create a standard parameter, that would assign the amount of people which could sit on the given piece of furniture. These furniture families are later updated in the original Revit document. Once loaded a new database can be created that identifies all the furniture, (see Fig.3) inside each room and shows the total sum of occupants for each level and room.

By calculating the number of seats designed within the space we can more accurately know the number of people the space is designed for. The process for using the tool, is divided into three simple steps; first the tool 'harvests' or collects all the furniture families available in the file, and creates a new parameter, while adding it and uploading it to each one of the families. Secondly, we assigned a value to that parameter which is done manually (the user will be prompted to do so for each element). Thirdly, a database in which all the furniture is sorted by level and later, by rooms to count the amount of occupancy of each spaced based on the total amount of furniture available in the space. The first two steps would only have to be performed once, as once this newly parameter is created once the family file is loaded into another document it will contain this information.

The outcome of this process has multiple benefits; one is the calculation of actual occupancy as opposed to occupancy load which gives a more realistic understanding of the space required for a particular program. These calculations can be used to help determine better egress systems strategies and circulation cores, allowing buildings to perform more efficiently based on the amount of people that use each area, not the amount of people allowed by law.

CASE STUDIES:

For architects, program distribution is one of the most important components to the schematic design of a building. Relationships such as scale and proportion are intrinsic data points that help guide a project, these have very often if not always been based on intuition and precedent. However, most of this information consists of numeric data, which in large numbers can be hard to interpret. Generating visualizations for these can be a tedious and time consuming process. When presented with this problem, our solution was to provide a simple and quick alternative to the traditional spreadsheets using TreeMaps. TreeMaps provides a hierarchical structure which creates an intuitive method for navigating spatial relationships. Fig.4(Top) shows an example of the resulting TreeMaps from multiple projects. The data collected from the Room Harvester contains information by department, room type and area. Using this information we are able to create clear and simple visuals that display the relationships of spaces in and across projects. This allows for the firm to quickly and visually compare program size and distribution, and to create relations from current and previous projects. Conceptually, we imagine being able to use this tool to compare schematic designs in realtime to alert a user to anomalies in their design when compared with the firm's historical metrics for similar programs.

Using ODBC exports or schedules created by Revit can be incredibly time consuming and nearly impossible to analyze across multiple models. The data is often organized in

Undefined ways and the IDs tagged to each object usually don't have any consistent convention. Using the Room Harvester tool we were able to solve the problem of comparing multiple projects side to side. For a firm this can be a labor intensive process that requires a full time employee working for a period of two weeks to gather the required information, and graphically represent it. Using Tableau we are able to use the data collected from the Revit documents and create an interactive dashboard for analyzing multiple projects simultaneously. Reducing the labor of weeks into a matter of seconds. The interface is dynamic, allowing data from a collection of projects, (Fig.4(Bottom)) displaying more complex relationship between models. This tool enables projects to be easily compared, analyzed and

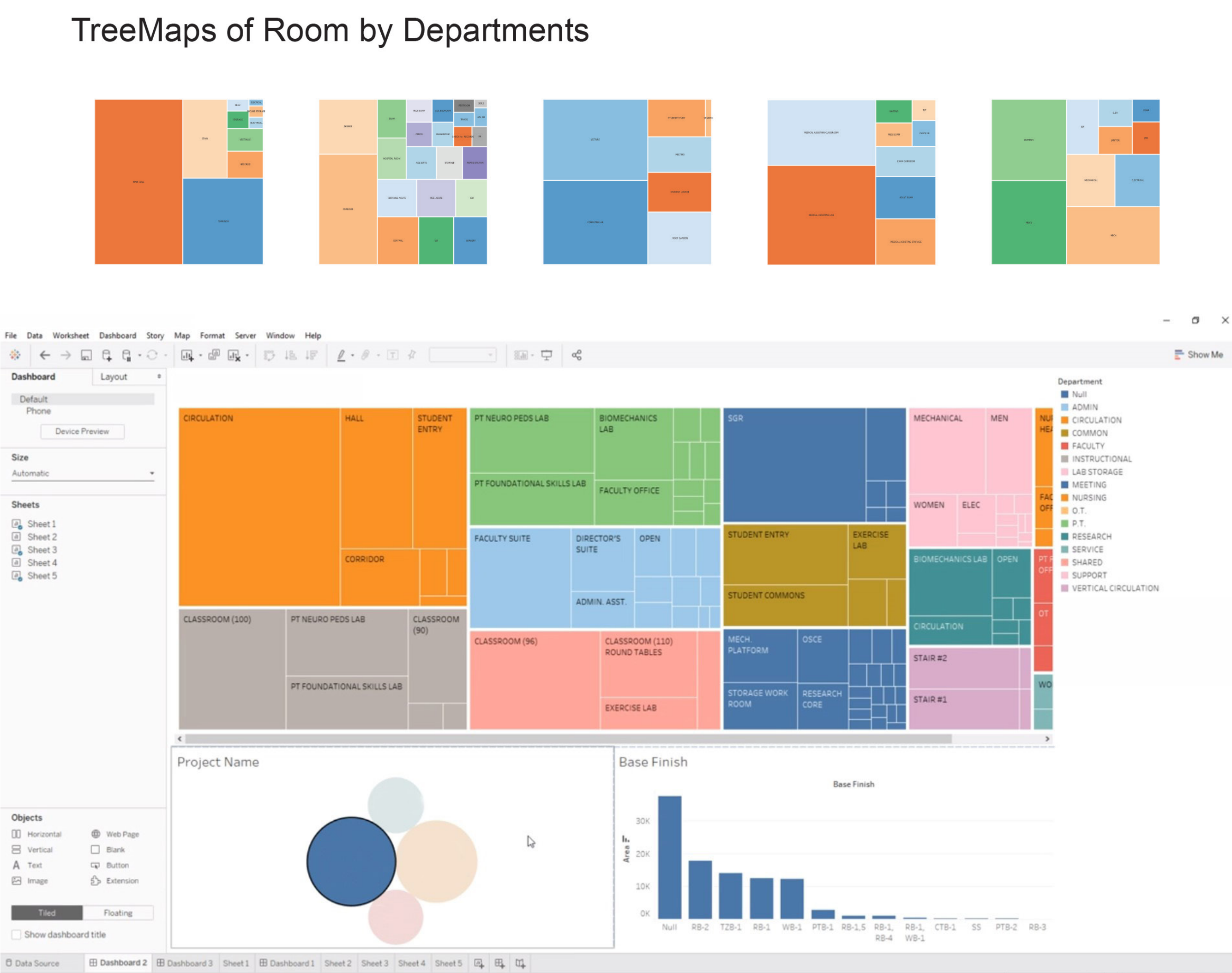

Figure 4. (Top) Resulting TreeMaps from multiple Revit Documents (Bottom) Tableau interface, displaying data from multiple Revit projects

referenced based on any data type, providing tremendous new insight for future projects.

DISCUSSION AND CONCLUSION

As we continue this work, we intend to generate a more holistic perspective on programming strategies, family use amongst various building typologies, and to use our new standards to create a custom, searchable database based off of firm-wide models. We anticipate these tools and their use to help refine and analyze the firm wide design and modeling standards as well as to create templates based on successful past modeling analysis.

The analytical tools which could sit on top of this database would create new ways for architecture firms to track previous organizational strategies across all program types. Additionally, we would work to create other tools which could be used to track redundancies, potential strengths and weaknesses in design processes, communication with consultants, contractors and code officials, which could result in reduced liability through higher quality models, less redundancy in time spent on drawing sets and modeling.

We intend to establish methods for linking key secondary information to the newly created database, potentially including programming strategies, levels of complexity and development, employees tied to the model, location, time, change orders, budgets and consultants. We will use this data set to establish a method for comparing data across the model set, to test for performative criteria. This data could be used to compare actual performance of built projects, between alternative designs for similar programs and locations. This approach could allow for deeper inquiry using more advanced exploratory visual systems or with Machine Learning (ML) and Artificial Intelligence (AI) algorithms to conduct both supervised and unsupervised analysis of the data given various problems architectural firms might be facing. Given that this an under-studied area of practice, there are still multiple unknowns on how this will affect the profession itself. However, some of the next steps for this project include performing more in depth analysis and tests on Revit models. While also understanding how these new practices will be incorporated in architecture firms current workflows.

Finally, This approach will allow for us to better understand the implications of the processes we use to design and make buildings in the 21st century and allow us to see unobservable patterns in the way we design, work and build. We have too often relied upon precedent and a reliance on a 'this is how we've always done it' mentality. These tools could allow us to actually see the patterns which show how we have always done it'. They could also be tremendously valuable in helping to realize the full potential of BIM modeling on the built environment.

ENDNOTES

1. Daniel Davis, "How Big Data Is Transforming Architecture." ARCHITECT, April 23, 2015, https://www.architectmagazine.com/technology/how-big-data-is-transforming-architecture_o.
2. German Aparicio. "Data-Insight-Driven Project Delivery:Approach to Accelerated Project Delivery Using Data Analytics, Data Mining and Data Visualization." ACADIA 2017 DISCIPLINES & DISRUPTION: Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, (November 2017): 102-109. Example of a conference proceedings paper in a book: [Author Name(s), first then last], "[Paper Title]," in [*Proceedings Book Name*], ed. [Editor Name] ([Publisher City: Publisher Name, Year Published]), [Page Number(s)].
3. Yishan Li, and Sathiamoorthy Manoharan. "A Performance Comparison of Sql and Nosql Databases." Paper presented at the 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), 2013.

BIBLIOGRAPHY

1. Belle, Ashwin, Raghuram Thiagarajan, SM Soroushmehr, Fatemeh Navidi, Daniel A Beard, and Kayvan Najarian. 2015. "Big data analytics in healthcare." BioMed research international 2015.
2. Johanson, Mathias, Stanislav Belenki, Jonas Jalminger, Magnus Fant, and Mats Gjertz. 2014. "Big automotive data: Leveraging large volumes of data for knowledge-driven product development." 2014 IEEE International Conference on Big Data (Big Data).
3. Nagpal, Abhinav, and Goldie Gabrani. 2019. "Python for Data Analytics, Scientific and Technical Applications." 2019 Amity International Conference on Artificial Intelligence (AICAI).
4. Talarico, Gui. 2018. "revitpythonwrapper Documentation."