Towards the use of artificial neural networks in the early stages of architectural design processes: integrating architect's declarative domain knowledge into ANN

Daniela Martuffi[1], Pedro Soza[2], Rodrigo Garcia[3]

[1] Universidad de Chile, Santiago, Chile
daniela.martuffi@ug.uchile.cl
[2] Universidad de Chile, Santiago, Chile
psoza@uchilefau.cl
[3] Universidad del Bio-Bio, Concepción, Chile
rgarcia@ubiobio.cl

Abstract. This research seeks and proposes the integration of architects' common stock of knowledge to leverage artificial neural networks, given the lack of active tools that allow for human-computational interaction and that can assist in the preliminary stage of design in an informed manner. The research was conducted as experimental and qualitative research, and its methodology design was divided in two stages, working first in a qualitative approach, capturing extractable implicit knowledge with semantic analysis, and secondly, by delivering the knowledge to an ANN through the processing of semantic information by a NLP model to create matrix and node graphs. Results demonstrate the potential of ANNs to generate representations from declarative database adjacency relations and show that the leveraging of AI with architects' declarative knowledge could provide a way to improve the initial phases of architectural design and decision-making through an iterative and knowledge-guided approach.

Keywords: Parametric Design, Artificial Neural Networks, Declarative domain knowledge, Graph representations, Design Process.

1 Introduction

This research seeks and proposes the integration of architect's common stock of knowledge to leverage artificial neural networks, given the lack of active tools that allow for human-computational interaction that can assist in the preliminary stage of design in an informed manner. Given that design problems do not have a definitive formulation or an established rule of thumb (Rittel &

Webber, 1971), a tool of this nature would allow for a deeper understanding of how architects formulate design problems, and new possibilities when making decisions. Design processes as unstructured (Simon, 1973), abstract and incomplete problems allow for further exploration, carried out by producing semantic and procedural representations (Soza, 2018) through movements across and along the design episode (Goel, 1999) with iterations of moving and deepening of ideas (Goel, 1995). Although architects' design processes are different in everyone, they are based on exploration using a common stock of knowledge, with shared languages and representations such as plans, isometrics and sketches (Schon, 1992) and searches for similar solutions and strategies. These processes involved in the preliminary design require unstructured mental representations of the problem that can support transformations (Goel, 1995) to structure the problem through the process of formalization. Hence, the design problem is transformed into a representation of the problem and thus modified by internal and external changes that act on it. Based on the problem representation, two ways are defined to address it (Kelly & Gero, 2021), one through design thinking and the other through computational thinking, where both function as shifting ways of reasoning between particular and general. Since design is defined as an activity of exploration and learning that operates within a context-dependent on perception, goals and constrains (Gero,1990), design thinking can be understood as the knowledge that has been developed in relation to the reasoning involved in the decision making while dealing with this type of problems. Alternatively, computational thinking can be described under two definitions (Wing, 2006), one based on the type of reasoning that is used, involving problem-solving, designing systems and understanding human behavior from the fundamentals of computer science, and one based on the type of solutions produced, defining it as the process involved in formulating problems and their solutions to be effectively represented by an information processor.

Computational design, on the other side, allows us to go further than our stock of knowledge, memory or imagination by opening up new possibilities as solutions or strategies when designing. Most known generative design is able to generate geometries using algorithms based on an objective, limits and rules, while artificial intelligence, specifically artificial neural networks allow us to work with more complex tasks, generating alternative solutions and design strategies from the latent space, in addition to a learning in the agent from the dataset. This type of design currently allows for increased efficiency around formal development, however, most of this kind of design is a product from the work of a passive tool handled by humans and not as an active tool that enables the evolution of design. This is due to the non-integration of the architect's stock of knowledge with the tool, but rather an implementation, taking a secondary role according to its operational nature within the design process. One generalized problem of artificial intelligence, specifically machine learning, involves the amount of data required to obtain higher accuracy, reason why it is suggested to integrate human knowledge to reduce the need for large amounts of data while increasing reliability to build up systems that can be understood by humans (Deng et al, 2019). This integration supposes two main advantages: first, by being explained by experts from each field it can be validated with greater rigor, generating a statistically more reliable mass, and secondly, by being subject to interpretation, it helps to deepen the theoretical mechanisms of machine learning models. For this integration to be successful, a crucial requirement is for the input and output to be both human and computational interpretable.

2 Methodology

The research was conducted as experimental and qualitative research, and its methodology design was divided into two stages since it's necessary, in the first place, to understand which implicit knowledge can be extracted from the architect's stock of knowledge and, in second place, how this knowledge can be provided to and implemented by a machine.

2.1 Stage 1: Knowledge capture

The first stage of knowledge extraction was defined by understanding the problem, identifying the typology or case to be analyzed and it was decided to utilize space distribution schemes in house floorplans. There were two main reasons for selecting schematics, first to analyze the design process in the preliminary stage, i.e., the first approaches to the resolution of a floor plan, and second, as they use conventional symbols to represent the components in a spatial sense (Yi-Luen,1995). Within the semantic knowledge found in the diagrams, it is possible to distinguish the information from nodes as well as connections, corresponding to spaces and their attributes, and type of relationship existing between them respectively.

The second step in knowledge extraction was data identification, since in the context of artificial intelligence and neural networks, it is essential to know required and expected data, the dataset is determinant for its implementation. In this research, it was defined that the relevant data to be extracted were topological relations of connection between rooms. To define these relationships, spatial reasoning based on RCC8 calculus connection relationship was used. Given the qualitative nature of these relationships, they must be processed by a natural language processing algorithm (Wang, 2018).

For data collection, given the experimental nature of the research and its specific targeting on a particular stage of the architectural design process, no public data was available, therefore, a self-made dataset was elaborated from sketches collected in architecture schools by last year's students and

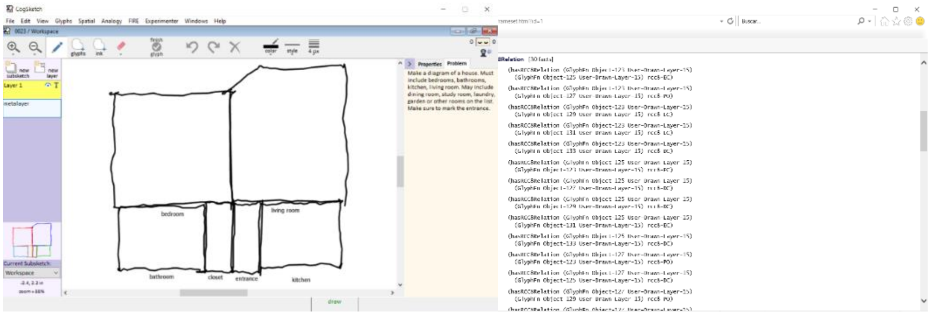

Figure 1. Diagram in Cogsketch and its global knowledge. Source: Cogsketch. 2022

professional architects. Drawings were collected using Cogsketch, a sketch comprehension software that provides an interface for assigning meaning to the drawings. This software integrates a set of visual and spatial processing routines to encode the properties of sketches in a human-like format, plus a large world knowledge library to provide conceptual understanding of the drawing (Forbus & Wetzel, 2020). The program can compute topological relations from RCC8 relations such as intersections, containments or superpositions based on disconnected, tangent, or edge-connected relations.

After diagrams were collected, data was extracted to be converted into a data structure, Cogsketch displays knowledge about drawings in a HTML format, which is exportable to a spreadsheet where data cleanup was carried out, outlining knowledge into datasets of labels and relations. Concluding the sorting, two files were generated, a chart of relationships with “room 1”, “room 2” and “relationship type” in .cvs format, and a text string with the same data written in prose of the form - “Room 1” has a relationship of the type “relationship type” with “room 2” in .txt format. The two files were used respectively as output and input to start training the neural network.

2.2 Stage 2: Knowledge transfer

The second stage involves the implementation of natural language processing model with artificial neural networks according to computational methodology. For this implementation, an artificial neural network and language must be chosen to use in a visual development environment. To process extracted data the natural language processing model used was Word2Vec, which converts data to vectors using Secuential keras artificial neural network, to learn word associations from text blocks. To begin the processing two definitions were made: the initial input library with the previously extracted .cvs files, and the relationships to be found with a list defined as:

```
labels[i]= ["rcc8-DC", "rcc8-EC", "rcc8-PO", "rcc8-TPP",
"rcc8-EQ", "rcc8-NTPP"]
```

```
['Kitchen', 'Bedroom', 'Closet', 'Entrance', 'Bathroom', 'LivingRoom']
El archivo es el numero 023 y su cantidad de relaciones es 30
[[0, 16], [1, 16], [2, 4], [3, 0], [4, 0], [5, 0]]
```

```
[[0. 0. 0. 1. 0. 1.]
 [0. 0. 2. 1. 1. 2.]
 [0. 2. 0. 1. 1. 1.]
 [1. 1. 1. 0. 0. 1.]
 [0. 1. 1. 0. 0. 0.]
 [1. 2. 1. 1. 0. 0.]]
```

Figure 2. Matrix from drawing 23. Source: Martuffi, D. 2022

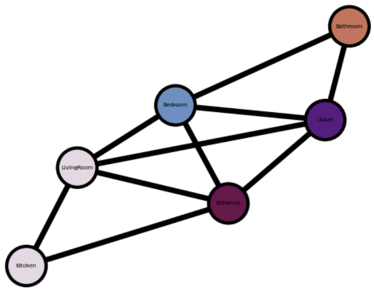

Figure 3. Example node graph generated from matrix 23. Source: Martuffi, D. 2022

with values from 0 to 5 respectively. Subsequently, a definition was created to run through the data to sort a list of spaces and a list of labels, which must fill each other. To do this, the .cvs file is converted into a dataframe using Pandas, and then run through to fill in the spaces, setting the numbers of spaces as `M = len(list_habit)`. From dataframes we made matrices using Numpy zeros, creating an MxM sized matrix with the relationship list and a "for" loop ran through the dataframe and matrix. With `list_habit.index` the relations found in the dataframe are inserted into the matrix as it searches for the name of the relation and converts it to the position of the vector within the matrix. After this process, the file number, list of labels, and relationship matrix are printed.

From matrices, graph representations were generated as a reinterpretation of the neural network processing. For this, a NetworkX model was implemented as it analyzes structure, dynamics, and function of complex networks. This model was chosen since it determines convenient node positions avoiding knots and offers spring form generation that simulates connecting forces keeping connectors close while trying to repel nodes. At the same time, values for each space were separated by color, and a position was assigned to them in x,y values. For its creation, a main function was generated to produce the graphs from the dataframe, which runs through the entire matrix directory and generates iterations of each file using `relation_matrix=df_to_node(filename)`.

3 Results

Under this procedure, a distinction must be made between two types of research results based on quantitative and qualitative data. Quantitative data was defined as the disaggregated data derived from the methodology of the study, in this case, numerical data about the number of labels and relationships between spaces. They have discrete values, i.e., a limited number of possible values based on the number of spaces in a dwelling.

On the other hand, qualitative data were defined as descriptive data, properties and characteristics that were obtained, such as type of relationships, methodology of data collection and thematic data together with their attributes. Thematic data are understood as data related to a particular topic or subject, in this research, the thematic data are representations of graphical reinterpretation of semantic information.

3.1 Quantitative Results

Quantitative data were extracted through analysis from 150 diagrams, both individually with 150 analyses, one per diagram, and collectively as a data set within a dataframe.

When analyzing data globally we can make observations about the number of spaces (labels) where the most frequent space was bedroom with 298 occurrences and the least frequent was garage with 22 occurrences. Is also possible to identify that layouts with the fewest spaces were layouts that only proposed the required spaces specified in the given problem: entrance, bedroom, bathroom, kitchen and living room. The layout that proposed more spaces had 20 spaces, the average per layout was 9.1 and the total number in the 150 drawings was 1347.

About relationships it is possible to determine that the minimum number of relationships in a layout is 20 since the required spaces are 5, each one having a relation with the remaining 4. From this way of extracting the result, it is possible to determine that the number of relationships has a function of (N^2)-N, however, since most of the relationships were 0 or disconnected this function is insufficient. The total number of relationships are

[0, 8449], [1, 3576], [2, 1204], [3, 41], [4, 0], [5, 9]

from which it is possible to extract the percentage of relations with 63.5% of disconnected, 26.9% of connected by an edge, 9% of overlapping, 0.3% of inside tangential and 0.06% of inside non-tangential.

Given this outcome, it is possible to observe that the most common relationship is DC, since all labeled spaces present in the sketch that are not adjacent to a room will be disconnected from it. The second most common relation is EC, connected by an edge, since rooms usually have 4 adjacency faces and they connect at least two of them. An important result on the

Table 1. Total relations chart. Source: Martuffi, D. 2022

	BD	BT	CL	DR	KT	EN	GR	LR	HW	SD	LD	ST	GA
BD	214	207	49	27	42	42	74	101	227	24	2	6	5
BT	207	44	46	16	61	32	24	45	146	15	4	7	5
CL	49	46	0	2	2	5	3	9	22	1	3	1	1
DR	27	16	2	0	64	20	22	80	53	5	5	55	4
KT	42	60	2	64	2	51	27	94	93	12	18	9	6
EN	42	32	5	20	50	0	9	55	54	10	2	9	4
GR	72	24	3	22	27	10	4	50	34	4	6	7	6
LR	101	45	9	80	94	56	50	0	105	14	11	10	3
HW	227	146	22	53	93	55	34	105	30	24	8	11	4
SD	24	15	1	5	12	9	4	14	24	2	1	1	0
LD	2	4	3	4	17	2	6	11	8	1	0	4	0
ST	6	7	1	5	8	8	7	10	10	1	4	0	1
GA	5	5	1	4	6	4	7	3	4	0	0	1	2

frequency of relationships is the number given by space pairing, which varies depending on whether the "disconnected" relationship is considered. For a better analysis, the "disconnected" relationship is ignored since it provides information that is not relevant at global level and misrepresents the data.

From the table of total relationships between spaces (Table 1), it is possible to observe that the most frequent relationship is between room-hallway, room-room and room-bathroom. It can also be seen that the spaces closet, dining room, entrance, living room, loggia and cellars are usually unique or do not have a relationship of adjacency with a space of the same type.

3.2 Qualitative results

It is possible to determine the method for data collection and codification of the relationships as a result from the extraction of the implicit semantic knowledge used in the space distribution diagrams of a house. This generates two reinterpretations by the neural network from the NLP model, which are presented as thematic data, adjacency matrices and node graphs.

The first results of this research are diagrams drawn in Cogsketch, which reflect the knowledge from the common stock that we architect possess and recall when making decisions, allowing us to assemble geometric compositions based on our implicit knowledge about the relationship of these spaces.

From this we obtained the semantic information of the adjacency relations based on the RCC8 calculus connection relations in *.csv format, forming the database we used to feed the artificial neural network. From its processing, it was possible to establish content and process respectively as thematic data and knowledge extraction strategies.

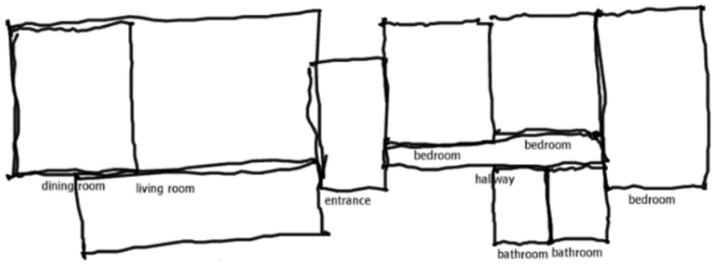

Figure 4. House plan distribution scheme 24. Source: Cogsketch. 2022

```
[[0. 1. 0. 0. 1. 0. 0. 1. 0. 2.]
 [1. 0. 1. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 2. 1. 0. 0. 0. 0. 0.]
 [0. 0. 2. 0. 2. 0. 1. 0. 0. 0.]
 [1. 1. 1. 2. 0. 1. 1. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 1. 0. 0. 0.]
 [0. 0. 0. 1. 1. 1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 3.]
 [2. 0. 0. 0. 0. 0. 0. 1. 3. 0.]]
```

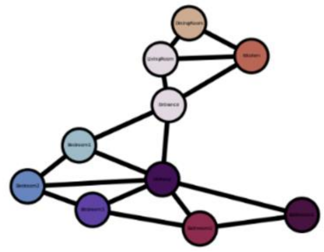

Figure 5. Thematic data form drawing 24. Source: Martuffi, D. 2022

Adjacency matrices shown in Figure 3 represent sorted relationship data in the form of a NxN sized matrix, where N represents the number of rooms. The rows and columns represent each room and their intersection is determined by the type of relationship. The relationships were set as 0 for "disconnected", 1 for "connected by an edge", 2 for "overlapping", 3 for "within tangentially" and 4 for "within non-tangentially". To clarify the matrix it was assumed that all elements are related to themselves by creating a diagonal of 0 in the matrix from the upper left corner to the lower right corner, and, that the relationships are bidirectional, i.e., if A is adjacent to B, then B is adjacent to A, so the diagonal corresponds to the symmetry axis of the matrix.

The node graph shown in Figure 4 shows the same spatial relationship data reinterpreted by the neural network in the form of a topological graph, where the spaces are defined as nodes and their adjacencies by lines. It is important to clarify that this is not a program graph since neither partitions nor spans were considered, but rather the complete distribution of the spaces within the house organized in a cluster-like arrangement.

4 Discussion and final thoughts

Results show that it is possible to extract declarative knowledge in the form of patterns and room relationships. This demonstrates that the integration of human knowledge and computational capabilities, by adding new procedures that contribute to the skills in particular task domains and improve existing procedures (Simon,1996), reflects the learning of the artificial neural network based on architectural strategies used to represent the input in different outputs. Thus, thematic data generated from the artificial neural network acts not only as a representation of architectural knowledge but also offers representations that highlight the importance of integrating architects' common stock of knowledge, although this approach based on adjacency relations is just one of the many ways to address the broader scope of the problem formalization, it serves as an input for the artificial neural network to help shape the problem and as a guide to allow new insights while leveraging computational potential.

As Niedereer (2007) proposes, one of the existing problems in the search for design solutions is the prioritization of propositional and explicit knowledge, excluding formats associated with practice. In this respect, this integration tries to bridge the gap by explicitly defining implicit knowledge. By revealing latent expertise embedded in practice, integration aims to transform tacit insights into practical knowledge to empower designers to explore a wider range of potential solutions. The introduction of artificial neural networks into the architectural design process could provide a convincing solution to the challenges that arise from its inherently iterative nature. Iteration in design processes is often limited by time and may hinder the exploration and deepening of ideas within short periods of time, which may lead to the premature abandonment of concepts (Boden, 2019). This research shows that the leveraging of artificial neural networks with architects' declarative knowledge could provide a way to expand and deepen the initial phases of architectural design through an iterative and knowledge-guided approach by making more manageable and narrowing down the problem space.

.

References

Boden, M. A., & Edmonds, E. A. (2009). What is generative art? Digital Creativity, 20(1–2), 21–46. doi:10.1080/14626260902867915

Deng, C., Ji, X., Rainey, C., Zhang, J., & Lu, W. (2020). Integrating Machine Learning with Human Knowledge. IScience, 101656. https://doi.org/10.1016/j.isci.2020.101656

Forbus, K., & Wetzel, J. (2020). CogSketch Manual. 2020.

Gero, J. S. (1990). Design prototypes: a knowledge representation schema for design. AI Magazine 11

Goel, V. (1995). Sketches of Thought. Cambridge, MA: MIT Press 21

Goel, V. (1999) Cognitive role of ill-structured representations in preliminary design. Database and Expert Systems Applications, 131–143. http://papers.cumincad.org/data/works/att/df70.content.pd

Kelly, N., & Gero, J. S. (2021). Design thinking and computational thinking: A dual process model for addressing design problems. Design Science, Ennis 1993, 1–15. https://doi.org/10.1017/dsj.2021.7

Niedderer, K. (2007). Mapping the meaning of knowledge in design research. Nordes 2007: Design Inquiries, 1(April), 1–27. https://doi.org/10.21606/nordes.2007.002

Rittel, H. W., & Webber, M. M. (1973). "Dilemmas in a General Theory of Planning." Policy sciences, 4(2), 155-169.

Schön, D. A., & Wiggins, G. (1992). Kinds of seeing in designing. Creativity and Innovation Management, 1(2), 68–74. doi:10.1111/j.1467-8691. 1992.tb00031.x

Simon, H. A (1996). The Sciences of the Artificial, 3rd Edition, MIT Press Books, doi:10.7551/mitpress/12107.001.0001

Simon, H. A. (1973). The structure of ill structured problems. Artificial Intelligence, 4(3–4), 181–201. https://doi.org/10.1016/0004-3702(73)90011-8

Soza Ruiz, P. A. (2018). On the role that specific domain knowledge and procedural strategies play in defining the episodic nature of architectural design formulation. August, 235Wing, J. M. (2006). Computational thinking. Communications of the ACM 35.

Wang, Y & Qiao, M & Liu, H & Ye, X. (2018). Qualitative spatial reasoning on topological relations by combining the semantic web and constraint satisfaction, Geospatial Information Science. doi: 10.1080/10095020.2018.1430659

Yi-Luen Do, E. (1995). What's in a Diagram That a Computer Should Understand? Proceedings of the Sixth International Conference on Computer-Aided Architectural Design Futures, 469–480