Natural Language Processing and Collaborative AI in Minecraft

Elizabeth Kaplan **Prashant Jayannavar** **Julia Hockenmaier**
University of Illinois at Urbana-Champaign & North Carolina State University
(eakaplan@ncsu.edu) **(paj3@illinois.edu)** **(juliahmr@illinois.edu)**

Abstract

The goal of our project is to create a Builder AI and an Architect AI that can communicate between each other to create block structures in Minecraft. The architect communicates instructions to the Builder of how to create a final build structure. The Architect can observe the Builder but cannot place blocks. The Builder uses the Architect's instructions to place blocks to build the structure. The Builder can also communicate and ask questions to the Architect in return.

1 Introduction

The purpose of the project is to produce collaborative Architect and Builder AI that can communicate through natural language processing. Our hope is that the instructions and communications between the AI mirror the standard human communication that was collected through our human trails. (e.g. Narayan-Chen,2019). A sought-after goal in the field of AI is creating agents that can communicate their surroundings with humans and cooperatively solve tasks in the given environment. (e.g. Winograd,1971). Our work attempts to further the research and understanding behind this goal.

2 Seq2Seq Builder Utterance Model

CNN We have implemented and trained a convolutional neural network (CNN) to recognize simple shapes in a build configuration. The CNN was trained to recognize seven primary shapes: row, diagonal, plane, rectangular prism, L-shape, U-shape, and T-shape. There are different orientations to these shapes (i.e. L_vertical_up) for a total of eighteen shapes. We also include sub-shapes as sub-labels for these figure (i.e. a plane is made up of rows, so a plane has a sub-shape of rows and is represented in the labeling of our figures.) These shapes were generated randomly along with composite shapes; composite shapes had two to four shapes in a given build area.

The build area is represented as a grid of one-hot vectors, meaning there is a "1" where a block is present and a "0" where there is the absence of a block. The grids represent the 11x9x11 Minecraft world, so the grids array is length 1089. Each grid only represents a single-color channel. The CNN is given six color channels (red, orange, yellow, green, blue, and purple), one at a time in order to produce predictions in the Seq2Seq model. We are in the progress of creating a CNN that can take in a vector array as an input. Our current CNN model only takes in a single vector (processing a single color at a time). This model has performed exceptionally. A table below illustrates the accuracies of the shape predictions.

	Row	Plane	Rectangular Prism	Diagonal	L-Shape	U-shape	T-shape
Precision	100	100	100	99	100	98	99
Recall	100	100	100	98	100	99	100

Figure 1: Precision and Recall Data of Shape Predictions from CNN

Dialogue Acts To introduce another metric for measuring the accuracy of Builder responses, we have introduced text categorization. We have sorted the Builder responses into eight mutually exclusive categories, called dialogue acts. These acts summarize the Builder response types. These categories are: Greeting, Verification Questions, Clarification Questions, Suggestions, Extrapolation, Display of Understanding, Build Materials Update, and Chit-Chat/Other. The Greeting is a welcome or recognition statement, for example, “Hello. What are we building this time?” Verification is a statement the Builder asking the Architect to verify the action was correct, such as “Is this good?” Clarification is a question to aid accuracy and understanding of the Architect’s instruction, for example asking where to place a block, “On top the 8th block?” Suggestions, help propose an idea for the Architect to consider, “Can you give me an instruction for a single purple block first?” Extrapolation is when the Builder makes an educated guess as to how the build structure should/will be produced, for example saying, “let me try this” and proceeding to build without explicit instruction. Display Understanding is expressing understanding, “Roger that,” or lack of understanding of the given instruction, “I think I messed up.” Build Materials Update is a response about the current quantity of build materials, “I only have 20 blocks of each color. I used them up. Okay I will remove some.” Finally, there is Chit-Chat/Other which is any other statement that is usually not necessary to the completion of the build structure. Our hope is that the text categorization will aid in the accuracy of the Builder’s generated responses.

The model is designed with an RNN (recurrent neural network). The model takes in the previous Builder and Architect utterances and makes a prediction about the type of utterance that should be generated. This model serves as metric in determining the accuracy of the generated responses. We can calculate the precision and recall of whether the proper dialogue-act class is generated by our models. This text classification helps in aiding the generation of responses so the meaning behind responses can be more relevant and accurate, even if the content is slightly different from our trials.

Block Weights We have implemented a convention to label the block weights of the figure as the builder is building. The weights are an integer (2-6) to keep track the five most recently placed blocks. The most recently placed block is the largest and the following blocks are each decremented by one. As before, an absent block is weighted 0 and a present block is weighted 1, or greater if it was most recently placed. The inclusion of this convention when loading the data into our Seq2Seq model will aid our Builder in the accuracy of its action prediction. For example, the block weights allow for the Builder to understand which block has been most recently placed, so it predicts that the next Builder action will likely build off those most recently placed blocks.

3 Future Works

In the future, we hope to be able to be able to produce a Builder Action Prediction Model. This model would allow the builder to predict what action to take next (place a block, remove a block, speak, or do nothing. The features we have implemented previously will hopefully aid the Builder in making intelligible action choices. For example, is the understanding of basic shapes so if the Architect instructs “Build a row of length 3 in blue.” With the aid of the CNN, The Builder will understand the concept of a row, and hopefully predict to place blocks side-by-side to form a row.

In the future, we also hope to include some of these same features into the Architect AI. The CNN model is also being re-used for the Architect to understand the build structure and give more accurate and complex instructions to the builder. The Dialogue Acts Classification would also be useful in aiding the accuracy of the Architect’s responses. The Architect has more utterances throughout each build mission, and the response have more complexity, so it is thought that we will implement a multi-label classification system. The Architect's quality of instructions could also be aided by understanding which block have been most recently placed. For example, "put another block next to the last one" or "remove that last one that was put down." Hopefully, with future progress the Architect and Builder AI's can produce competent, collaborative conversation, that successfully create the block structures.

References

Narayan-Chen, Anjali, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative Dialogue in Minecraft. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

Terry Winograd. 1971. Procedures as a representation for data in a computer program for understanding natural language. Technical report, MIT. Cent. Space Res.