# Enhanced Execution of Instructions in Minecraft Dialogue: An Extension with Attention Mechanism and Advanced Embeddings

**Stanley Joel Gona** **Mackwyn Quadras**
University of Potsdam, Germany
{gona, quadras}@uni-potsdam.de

## Abstract

Minecraft, known for its intricate nature, offers an ideal environment to investigate the complex connection between language-driven instructions and agent behavior. Our study builds upon the foundational research presented in ”Learning to Execute Instructions in a Minecraft Dialogue” (Jayannavar et al., ACL 2020). While the original study introduced Long Short-Term Memory networks (LSTMs), our objective was to comprehensively explore their capabilities. By integrating the attention mechanism with LSTMs as well as GRUs, we aimed to enhance the model's ability to understand and respond to complex, multi-turn dialogues. In the realm of word embeddings, we transitioned from GloVed to Google's Word2Vec, expecting a deeper semantic understanding. Furthermore, we incorporated the Leaky ReLU activation function to improve the model's learning dynamics. To provide a more comprehensive assessment of our model's effectiveness, we introduced Jaccard Similarity as an additional metric. Additionally, by implementing data prefetching techniques, we aimed to significantly streamline the training process. This paper offers a thorough evaluation of our enhanced model, highlighting its improvements and establishing a strong foundation for future research in instruction-driven agent interactions.

## 1 Introduction

Minecraft, recognized for its intricate nature, provides an ideal setting to explore how language instructions influence the actions of virtual agents. This study builds upon the fundamental research conducted in ”Learning to Execute Instructions in a Minecraft Dialogue” by (Jayannavar et al., ACL 2020). While the original research introduced Long Short-Term Memory networks (LSTMs), it didn't delve deeply into their capabilities.

Our research is a follow-up and enlargement of a groundbreaking study titled ”Learning to Execute Instructions in a Minecraft Dialogue” conducted by (Jayannavar et al., ACL 2020). Their pioneering research introduced a model called Seq2Seq, utilizing Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs) to decipher and respond to intricate dialogues within the Minecraft environment. However, the original study predominantly employed GRUs, leaving the possibilities of LSTMs relatively untapped.

In our research, we haven't just employed LSTMs; we've tried to improve the Seq2Seq model by incorporating the attention mechanism. This enhancement is vital as it empowers the model to focus on the critical aspects of a conversation, resulting in a more profound comprehension and precise responses during intricate multi-turn interactions in the challenging Minecraft environment.

Moreover, we've shifted from using GloVe to Google's Word2Vec for linguistic embeddings. This transition isn't just a technical adjustment; it's a deliberate improvement aimed at providing the model with a deeper and more nuanced understanding of language semantics. This enhancement enables the model to better comprehend and execute instructions with increased precision.

We've also implemented the Leaky ReLU activation function, a modification intended to enhance the model's learning process. This addition plays a vital role in enabling the model to effectively learn from the training data and apply this knowledge to unfamiliar situations. Consequently, it bolsters the model's robustness and predictive accuracy across the various scenarios encountered in Minecraft.

To guarantee a thorough and dependable assessment of our model's performance, we integrated Jaccard Similarity as an extra metric. This inclusion offers a quantifiable means to evaluate the model's precision in aligning predicted outcomes with real ones. It furnishes valuable insights into

http://juliahmr.cs.illinois.edu/Minecraft/

the model's practical usability and dependability.

Furthermore, we've incorporated advanced data prefetching techniques to streamline the training process, enhancing its efficiency and resource management. This enhancement is essential for addressing logistical hurdles and ensuring a more seamless development process.

This study isn't just a minor enhancement; it's a thorough investigation in the fields of AI and natural language processing. By studying AI within the diverse and lively world of Minecraft, we aim to discover fresh understanding and insights that can be applied to real-world situations, with the goal of making interactions with AI more user-friendly, flexible and centered around human needs and experiences.

The inclusion of the attention mechanism and the transition to Google's Word2Vec represent strategic improvements aimed at enhancing the model's understanding of language semantics and its ability to grasp context effectively. Additionally, the incorporation of the Leaky ReLU activation function and the utilization of Jaccard Similarity as a metric play vital roles in optimizing the model's learning behavior and offering a measurable gauge of its performance.

Our work is a comprehensive exploration, providing fresh perspectives, methodologies, and laying a robust foundation for future research in instruction-driven agent interactions within complex and varied environments like Minecraft. It's a synthesis of innovative thought and technical proficiency, pushing the boundaries of AI within gaming environments and contributing meaningfully to the evolving discourse in AI and natural language processing.

In conclusion, this paper is a detailed exploration and appraisal of an advanced model developed for navigating the intricacies of language-driven instructions within the diverse world of Minecraft. It's a journey of innovation and discovery, aimed at enhancing our understanding and capabilities in AI and showcasing the endless possibilities when insightful research is combined with advanced technology.

## 2 Related Work

Minecraft is not just a game; it's a complex world where every interaction and every block placed or removed can teach us something about how computers can understand and work with human language. Our exploration in this realm is guided by the footsteps of many who have ventured into the fields of AI and Natural Language Processing before us.

Researchers have delved deep into understanding interactions in collaborative settings, exploring how instruction givers and followers can communicate effectively. The studies by Hu et al. (2019), Suhr et al. (2019), Kim et al. (2019), and Ilinykh et al. (2019) have shed light on the dynamics of communication and cooperation, opening new perspectives on how we can enhance interactions between humans and computers.

Deciphering instructions is a complex task, and semantic parsing has been the key to unlocking this mystery in the domain of human-robot interactions. Pioneers like Chen and Mooney (2011), Artzi and Zettlemoyer (2013), and Andreas and Klein (2015) have bridged the gap between human instructions and robotic actions, paving the way for more intuitive and natural interactions between humans and machines.

The SCONE Corpus, developed by Long et al. (2016), has been a beacon for researchers, providing a structured framework for understanding context-dependent sequential instructions. It has served as fertile ground for advancing our knowledge in the domain of action prediction and object dynamics in predefined worlds.

The innovations in Vision-and-Language Navigation (VLN) and Cooperative Vision-and-Dialog Navigation (CVDN) by Anderson et al. (2018) and Thomason et al. (2019), respectively, have ushered in a new era of research focused on visually rich environments. These advancements have shifted the paradigm towards exploring intricate interactions within visually dense and information-rich settings.

Minecraft, with its dynamic and multifaceted environment, stands out as a unique challenge for studying computer-agent behavior and language understanding. It demands a rethinking of how we approach instruction-based interactions, requiring a nuanced approach to navigate its complexities.

We are voyagers in the vast sea of knowledge created by the collective wisdom of many researchers. By navigating the intricate world of Minecraft, we are building upon the rich legacy of prior research, aiming to translate our findings into real-world applications and contribute to the evolving dialogue in AI and Natural Language Processing. Our journey is enriched by the contributions of each researcher, providing diverse perspectives and insights that

enhance our understanding of the intricate interplay between language and action.

# 3 Task Formalization

In this comprehensive exploration of language execution in Minecraft dialogues, our primary goal is to enhance the AI model's ability to interpret and act upon instructions within the complex and varied Minecraft environment. We specifically concentrate on handling detailed multi-turn dialogues. The challenge at hand involves developing a model capable of not only comprehending the nuances of human language but also translating this comprehension into precise in-game actions.

## 3.1 Goal

Our main goal is to build upon and enhance the initial Seq2Seq model, developed by (Jayannavar et al., ACL 2020), by incorporating an attention mechanism and fully utilizing LSTMs. We aim to amplify the model's focus and understanding in dialogues, allowing for more precise responses in the multi-layered interactions found within Minecraft.

## 3.2 Model Description

Our approach involves the utilization of a Seq2Seq model, incorporating both GRUs and LSTMs, for decoding and participating in complex Minecraft conversations. While the foundational study by (Jayannavar et al., ACL 2020) primarily utilized GRUs, we have expanded upon this research to explore the potential of LSTMs in greater depth.

The addition of the attention mechanism is a significant improvement, allowing the model to focus on critical segments of a conversation. This enhancement is essential for effectively maneuvering the diverse scenarios in Minecraft and ensuring more precise and contextually aware responses.

## 3.3 Linguistic Representation

To attain a deeper grasp of language nuances, we transitioned from GloVe to Google's Word2Vec for linguistic representation. This strategic shift aims to equip the model with a finer understanding of linguistic details, thereby enhancing its ability to accurately comprehend and execute instructions.

## 3.4 Measurement Metrics

To thoroughly evaluate the model's performance, we introduced Jaccard Similarity as an additional metric. This inclusion offers a quantifiable means to measure the accuracy of the model's predicted outputs compared to the actual results.

## 3.5 Assumptions and Limitations

We believe that improving the model's capacity to understand and respond effectively in Minecraft's diverse environment will have broader implications for enhancing real-world applications in natural language processing.

# 4 Model Architecture

Embarking on the complex and ever-changing world of Minecraft, we've constructed a sophisticated and cohesive model. Building upon the groundwork laid by (Jayannavar et al., ACL 2020), our architecture is carefully crafted to bring together various elements, with the goal of effectively understanding and executing complex, multi-turn conversations in the diverse Minecraft environment.

## 4.1 Core Framework

At the core of our innovative architecture lies the enhanced Seq2Seq model, well-regarded for its proficiency in handling sequential data, making it the ideal choice for navigating the intricate dialogues inherent in our domain. It serves as the foundation, guaranteeing organized and logical dialogue understanding and response generation within the diverse Minecraft environment.

## 4.2 Encoder

Within our Seq2Seq framework, the Long Short-Term Memory (LSTM) networks are integrated as the fundamental component of our encoder. LSTMs, renowned for their ability to capture extended dependencies in sequences, play a pivotal role, ensuring our model maintains contextual consistency during lengthy dialogues. This is essential for precise interactions within the multifaceted and intricate settings of Minecraft.

## 4.3 Decoder

In our decoding process, we unite the attention mechanism with an Actions Decoder. This pairing optimizes the model's concentration on pertinent parts of the dialogue, leading to contextually enriched and precise interactions. Concurrently, the

Repository https://github.com/GrimGamer1999/BAP-unipotsdam$_{imgs}$

Actions Decoder plays a vital role in interpreting and effectively executing the derived instructions within the game environment.

### 4.4 Transition to Word2Vec

To enhance the semantic comprehension of our model, we have transitioned from GloVe to Google's Word2Vec for linguistic embeddings. This strategic shift is designed to augment the model's grasp of linguistic subtleties, facilitating a more nuanced and accurate interpretation and execution of instructions within Minecraft's elaborate scenarios.

### 4.5 Integration of Leaky ReLU

In our pursuit of optimized learning, the Leaky ReLU activation function has been integrated. This activation function mitigates the problem of inactive neurons and optimizes the learning trajectory, playing a pivotal role in enhancing the overall reliability and efficacy of the model.

### 4.6 Incorporation of Jaccard Similarity

To ensure a thorough and comprehensive assessment of the model's performance, we've added Jaccard Similarity as an extra metric. This addition provides a measurable aspect to gauge how well the model's predictions align with the actual outcomes, giving us valuable insights into its practical utility and trustworthiness in real-world situations.

### 4.7 Advanced Data Prefetching Techniques

To streamline the training process, advanced data prefetching techniques have been implemented. This enhancement is critical for overcoming logistical constraints and ensuring a smoother, more efficient developmental process, contributing to the model's successful training and refinement.

## 5 Data: Minecraft Dialogue Corpus

The Minecraft Dialogue Corpus, which was introduced by Narayan-Chen et al. 2019, forms the foundation of our project. It consists of 509 in-game conversations between two human players, centering around the Minecraft Collaborative Building Task. In this task, one player assumes the role of the architect, while the other takes on the role of the builder. The objective of the game is for the architect to examine a designated target structure, which the builder cannot see. Subsequently, the architect must provide verbal instructions to guide the builder in constructing the target structure. In this dataset, there's a variety of details neatly arranged into these categories.

### 5.1 Data Collection Details

Data collection for this project was carried out over a period of roughly three weeks, amounting to approximately 62 hours of data. In order to aid this undertaking 40 volunteers chimed in, comprising a mix of undergraduate and graduate students, each possessing varying degrees of familiarity with Minecraft. These participants were teamed up for 1.5-hour sessions, where they collaborated to construct predetermined structures within an 11x11x9-sized virtual realm. The Builders began with an inventory containing six different colored blocks, each set consisting of 20 blocks. On average, each gaming session lasted approximately 8.55 minutes.

### 5.2 Gaming Guidelines

Architects were prompted to give instructions without burdening the Builder and to allow space for responses. Builders received clear instructions not to place blocks outside the assigned building area and to follow the Architect's guidance. Natural communication was encouraged, discouraging unnecessary casual conversation. Participants had the flexibility to engage in multiple sessions, ensuring they didn't encounter the same target structure more than once.

### 5.3 Data Logging

The game's progress was carefully tracked by taking pictures at specific time intervals, like when blocks were put or removed, and when players conversed. These pictures, or 'snapshots,' included details such as the time, the chat history, the Builder's current location, the available blocks in the inventory, the current placement of every block, and images from both the Architect's and the Builder's point of views.

### 5.4 Dataset Structure

The Minecraft Dialogue Corpus is a collection of 509 dialogues held between human players, resulting in a total of 15,926 sentences and 113,116 words. This dataset encapsulates a diverse range of 150 target structures, each varying in complexity. There are a minimum of three dialogues available for every single one of these structures. To

Dataset: https://uofi.app.box.com/s/pwm7gr71hncqbtyscy9quyd446j19ydx support research and assessment, the dataset has been carefully divided into training, testing, and development subsets. This division ensures that the structures used for training purposes remain unseen during testing phases, contributing to a fair and unbiased assessment. The dialogues themselves, on average, contain approximately 30.7 sentences, although the length naturally varies based on the complexity of the target structure being discussed.

### 5.5 Mechanics and Player roles

In Minecraft, you can put blocks wherever they can touch an existing block or the ground, even if the supporting block is taken away later. This allows you to create structures that seem to float in the air. Surprisingly, about 53.6% of the target structures in the game are designed this way, with blocks that appear to float. In giving directions, Architects frequently used the Builder's point of view and recent actions to figure out how to describe things, making the game even more interesting and challenging. In the game, a few of the target structures resembled everyday objects, and Architects employed either clear or creative explanations to steer Builders in the right direction. At times, Architects even assigned names to parts of the structure to make the instructions more straightforward. Builders played an active role in the dialogues by seeking confirmation, asking for clarification, giving progress updates, offering suggestions, and sometimes even extending upon the instructions. These elements offer a thorough glimpse into the Minecraft Dialogue Corpus, highlighting its significance and complexity in examining how people communicate and work together within the world of Minecraft's construction challenges.

## 6 Experiments

In this section, we describe the models used and the experiments conducted to evaluate the performance of the proposed models for Builder Action Prediction.

### 6.1 Models Details

In the original paper (Jayannavar et al., ACL 2020), a range of neural network models were employed to efficiently handle structured data. At the core of the approach lies the EncoderRNN, a fundamental model proficient in processing sequences, including textual instructions. It provides the flexibility to utilize GRU or LSTM cells and can capture context from both ends of a sequence. To generate actions based on this contextual understanding, the ActionsDecoder is introduced, which can seamlessly collaborate with the WorldStateEncoderCNN. This convolutional neural network (CNN) model excels at comprehending spatial and structural information within an environment, often denoted as the 'world state.' Collectively, these models establish a potent framework for interpreting sequences, comprehending contextual subtleties, and generating responsive actions. This approach represents a valuable asset for the efficient management of structured data, making it an indispensable tool across various applications.

### 6.2 Expansion Details

**Attention Mechanism:** Our research incorporates advanced techniques hoping to enhance the performance and adaptability of neural network models. A significant improvement in the decoder model involves the integration of the attention mechanism. This mechanism, Implemented through the Attention class, dynamically calculates attention weights. This enables the model to focus selectively on specific segments of the input sequence. By doing so, it ensures that the decoder captures both local and long-range dependencies, resulting in a deeper contextual comprehension and more accurate outputs.

**Word2Vec Embeddings:** The original authors used a smaller Glove word embeddings file to create a vocabulary. In our approach, we opted to use a distinct type of word embedding file, specifically the Word2Vec word embedding file. Additionally, we conducted experiments with a slightly larger embedding file compared to the original Glove embeddings. This decision allowed us to work with a more extensive vocabulary generated from a Word2Vec embedding file. Our objective was to explore the effects of utilizing a different and larger vocabulary on our task.

**Activation Function:** Additionally, to introduce non-linearity into the models, the LeakyReLU activation function has been adopted across all three model files. In contrast to the conventional ReLU, LeakyReLU allows for a slight gradient even when the unit is inactive, mitigating potential issues such as the 'dying ReLU' problem. We Included it in our models to ensure a strong gradient flow during training, which might help the model converge faster and improve its performance.

**Prefetching:** We incorporated prefetching into our code, a technique that, while demanding on GPU resources, significantly boosts training speed. This optimization was particularly advantageous since we conducted our work in Google Colab, which offers access to GPUs on a daily basis. It's important to note that Google Colab imposes a daily limit on GPU availability, but this limit isn't determined by GPU usage itself; rather, it's based on the amount of time you utilize the GPUs. As a result, utilizing prefetching was an ideal choice for our situation.

**Metrics:** The original paper (Jayannavar et al., ACL 2020) employs various metrics such as f1-score, accuracy, recall, and precision. We made the choice to include Jaccard similarity in our analysis to introduce new metrics and facilitate comparisons among them.

### 6.3 Evaluation

**Baseline Results:**Before we dive into the enhancements and alterations introduced in our study, it's vital to fully comprehend the fundamental performance benchmarks established by the original models. The original research (Jayannavar et al., ACL 2020) opted for GRUs as the foundation of their models, deeming it the most suitable solution for their investigation. In our project, we initially chose to incorporate LSTMs to examine and contrast the two approaches, thereby establishing a baseline for the various modifications we intended to implement.

Table 1 showcases the extensive outcomes derived from the initial model using GRU units:

| GRU Results | | | |
|---|---|---|---|
| | H1 | H2 | H3 |
| BAP-base | 11.8 | 12.4 | 14.6 |
| + action history | 14.6 | 18.2 | 19.7 |
| + perspective | 15.7 | 18.7 | 18.8 |

Table 1: Performance with GRU models

On the other hand, Table 2 presents the performance metrics observed from the model utilizing LSTM units:

| LSTM Results | | | |
|---|---|---|---|
| | H1 | H2 | H3 |
| BAP-base | 12.1 | 11.1 | 17.3 |
| + action history | 12.6 | 12.1 | 15.8 |
| + perspective | 13.6 | 11.6 | 16.7 |

Table 2: Performance with LSTM models

These baseline results, characterized by the choice of recurrent units, play a crucial role in contextualizing the advancements made in our study. They not only provide a comparative framework to assess the effectiveness of our modifications but also offer insights into the inherent strengths and limitations of the original architectures, thus guiding our pursuit of model refinement. The experiment offers three approaches for providing game history to the models: H1 includes A's last utterance and any subsequent B utterances. H2 contains all utterances following B's second-to-last action sequence. H3 encompasses all utterances after B's second-to-last action sequence, alternated with a token representation of B's last action sequence. While H1 may suffice when A's last utterance is a standalone instruction, often, prior conversation is essential. Additionally, B's next action sequence is closely linked to their previous actions, justifying the inclusion of H2 and H3.

**Expansion Results:**In our effort to gauge how well the suggested alterations work, we methodically assessed the models in two main setups. We chose to exclusively assess H3, as it consistently delivered the most promising outcomes for both GRUs and LSTMs. Additionally, the limitations on time and GPU resources influenced our decision to concentrate solely on H3. Initially, we incorporated all the adjustments except for the activation functions and measured the performance using the H3 metrics. Afterward, we tested a complete model, including the activation function changes. This twofold evaluation approach was essential in understanding the separate and combined effects of the changes we proposed.

Table 3 presents the detailed results of the H3 metrics for the model without the activation function modifications:

| ReLu | | | | |
|---|---|---|---|---|
| | F1 Score | | Jaccard | |
| | GRU | LSTM | GRU | LSTM |
| BAP-base | 14.8 | 13.5 | 8.0 | 7.8 |
| + action history | 16.2 | 17.1 | 8.8 | 9.3 |
| + perspective | 15.7 | 14.1 | 8.1 | 7.6 |

Table 3: Performance with LSTM models

On the other hand, Table 4 encapsulates the outcomes derived from the model with all the proposed changes, including the activation function adjustments:

| LeakyReLu | | | | |
|---|---|---|---|---|
| | F1 Score | | Jaccard | |
| | GRU | LSTM | GRU | LSTM |
| BAP-base | 11.0 | 10.0 | 5.8 | 5.2 |
| + action history | 15.3 | 16.3 | 8.3 | 8.8 |
| + perspective | 16.6 | 14.9 | 9.1 | 8.0 |

Table 4: Performance with LSTM models

A comparative analysis of the results from both configurations provided us with valuable insights. While the metrics from the first configuration set a baseline, the improvements (or potential trade-offs) in the second configuration helped in determining the tangible benefits of the activation function modifications. Our findings, detailed in the subsequent sections, shed light on the efficacy of our modifications and pave the way for future refinements in this domain.

## 7 Analysis and Interpretation

### 7.1 Baseline Analysis

In the baseline analysis (Table 1 and Table 2), the model's performance was evaluated using GRU and LSTM units, respectively, as starting points. These tables established a reference for assessing the effectiveness of the proposed enhancements. The results in these tables were presented for different scenarios, considering various levels of conversation history (H1, H2, and H3).

**Crucial observations:** The original model (Jayannavar et al., ACL 2020) primarily utilized GRUs, with the LSTM results serving as a comparison. The performance metrics included the F1 Score, measuring the balance between precision and recall, and Jaccard Similarity, a metric for evaluating set similarity. The performance varied across different levels of conversation history, indicating that the context of the conversation played a significant role in model performance.

### 7.2 Expansion Analysis

**Table 3** - Outcomes with Suggested Refinements (Activation Function Alterations Excluded)
Table 3 outlines the results achieved by the model after integrating the proposed refinements, with the exception of the activation function modifications (ReLu). These enhancements encompass the attention mechanism, Word2Vec embeddings, Jaccard Similarity, and advanced data prefetching methods.

**Crucial observations:** The inclusion of the suggested refinements generally maintains performance levels compared to the baseline (Table 1 and Table 2). Both the GRU and LSTM models do not exhibit substantial improvements in the F1 Score and Jaccard Similarity. The introduction of Jaccard Similarity as an additional metric enables a more comprehensive assessment of model effectiveness.

**Table 4** - Outcomes with the Entire Set of Suggested Refinements (Incorporating Activation Function Modifications) Table 4 showcases the outcomes achieved by the model, encompassing all the recommended refinements, inclusive of the activation function alterations (LeakyReLu).

**Crucial observations:** Clearly, the activation function modifications (LeakyReLu) have a discernible impact on the model's performance. In a broader context, it becomes evident that while LeakyReLu tends to yield lower overall values in F1 Score and Jaccard Similarity compared to the outcomes in Table 3, it shows some promise in specific scenarios, particularly when considering action history and perspective coordinates.

#### 7.2.1 Comparative Analysis

As we dive into a comparative analysis between the outcomes in Table 3 (which excludes activation function changes) and Table 4 (which includes activation function changes), we uncover a nuanced scenario. Contrary to our initial expectations, the alterations made to the activation function, particularly the introduction of LeakyReLu, manifest mixed effects on the model's performance.
Taking a broader perspective, the utilization of LeakyReLu appears to yield lower overall values in terms of F1 Score and Jaccard Similarity when compared to the results in Table 3. However, a closer examination reveals a noteworthy trend: LeakyReLu displays promise when we zoom in on specific facets of the problem, such as action history and perspective coordinates. In these particular contexts, it tends to exhibit more favorable outcomes, suggesting its potential in fine-tuning the model's responsiveness to certain elements of the Minecraft environment.

Conversely, the combination of ReLu with LSTM, particularly when considering action history, emerges as the most favorable configuration in terms of overall value, even though it falls short of the baseline. This underscores the significance of carefully selecting model components to optimize performance in varying aspects of the dialogue context.

Moreover, the incorporation of the attention mechanism, Word2Vec embeddings, and Jaccard Similarity as evaluation metrics represents a significant enhancement in the model's ability to comprehend and respond to complex dialogues within the Minecraft environment. While these improvements do not universally surpass the baseline, they do signify advancements beyond the original research presented in Table 1 and Table 2.

In summary, this comparative analysis reveals that the introduction of activation function changes, particularly LeakyReLu, introduces a level of complexity to the model's performance. While it may not consistently outperform the baseline, it does demonstrate potential in specific dimensions. The combination of ReLu with LSTM, with a focus on action history, appears to offer the best overall value, albeit still falling short of the baseline. These findings underscore the intricate interplay of model components and stress the importance of tailoring them to different aspects of instruction-driven agent interactions within the Minecraft context.

Additionally, it's worth noting that our study encountered certain limitations, primarily linked to time and GPU constraints. We were restricted to running our experiments for just 20 epochs, whereas the baseline model was trained for 40 epochs to generate all necessary outputs. Unfortunately, executing the model for the full 40 epochs proved unfeasible due to resource availability. This limitation opens up the possibility that, with extended training, the results could potentially exhibit further enhancements. In the realm of deep learning, it is widely acknowledged that prolonged training durations can lead to improved model performance and convergence. Hence, while our current findings offer valuable insights, they may represent an initial glimpse of the model's capabilities, and future research with extended training periods could yield even more promising results.

## 8 Ethical Considerations

Our research places a strong emphasis on ethical considerations to ensure responsible and mindful progress in the field. Given our focus on enhancing AI models in the Minecraft gaming environment, ethical concerns revolve around data usage, model fairness, and the broader societal impact of our advancements.

First and foremost, the trustworthiness and dependability of the models were our top priorities. We took great care to thoroughly assess the changes and improvements made to the models to prevent any unforeseen issues or misuse. Every modification, whether it was integrating attention mechanisms, incorporating Leaky ReLU activation functions, or adopting Word2Vec embeddings, underwent meticulous scrutiny to ensure that our models remained transparent, responsible, and trustworthy. Our aim was to prevent the spread of false or biased information that could harm users or reinforce stereotypes and inequalities.

Additionally, we considered the environmental impact of developing and deploying advanced models, striving for efficiency and responsible resource use during training and operation.

Lastly, we were mindful of the real-world applications beyond gaming. By improving AI models' ability to understand and act upon language-based instructions, we recognized the far-reaching influence of our work in areas such as human-computer interactions and natural language processing. We remained dedicated to transparency, accountability, and responsible innovation, aligning our contributions with ethical principles and societal wellbeing.

In summary, our research adheres to strict ethical standards, aiming to contribute responsibly and meaningfully to the field while considering the broader implications and potential consequences of our endeavors.

## 9 Conclusion

In this research endeavor, we embarked on a comprehensive exploration aimed at elevating the performance of a Seq2Seq model in its ability to comprehend and execute instructions within the context of Minecraft dialogues. Our analytical journey encompassed various stages, commencing with the foundational assessment and progressing towards the introduction of innovative refinements and modifications to the activation functions.

During the initial analysis phase, we employed GRU and LSTM units as our reference points, unveiling the intricate role of conversational history in influencing the model's performance. We meticulously gauged precision and recall through the F1 Score while quantifying set similarity using the Jaccard Similarity metric. These measures shed light on the nuanced contextual factors that support effective interactions between the agent and instructions.

As we delved deeper into the expansion analysis, as presented in Table 3, we introduced a few enhancements. These encompassed the incorporation of the attention mechanism, Word2Vec embeddings, and Jaccard Similarity, with the expectation of witnessing an overall improvement in the model's performance. However, our findings revealed a subtle story. Surprisingly, these enhancements did not yield substantial enhancements in the F1 Score and Jaccard Similarity. Nonetheless, the addition of Jaccard Similarity as an additional evaluation metric broadened the scope of our assessment.

Moving on to Table 4, which included all recommended refinements, along with activation function modifications (LeakyReLu), the complexity of the situation deepened. It was evident that LeakyReLu did not consistently outperform the previous configuration, demonstrating lower overall values in F1 Score and Jaccard Similarity. Nevertheless, a more detailed analysis revealed its potential in specific contexts, particularly when dealing with action history and perspective coordinates.

Of particular note, the combination of ReLu with LSTM, particularly when considering action history, emerged as the most promising configuration, despite not surpassing the baseline. This highlights the intricate balance required among model components to optimize performance across different dimensions of dialogue context.

Furthermore, the incorporation of the attention mechanism, Word2Vec embeddings, and Jaccard Similarity as evaluation metrics represented significant progress. These additions enhanced the model's capacity to navigate intricate Minecraft dialogues.

It is essential to acknowledge the limitations of our study, primarily stemming from time and GPU constraints. Our experiments were confined to a mere 20 epochs, while the baseline model underwent 40 epochs of training to generate essential outputs. This constraint, driven by project deadlines and resource availability, introduces the possibility that extended training could yield further improvements. In the domain of deep learning, protracted training periods are known to result in enhanced model performance and convergence.

In conclusion, our research delves into the intricate dynamics of enhancing Seq2Seq models in the context of Minecraft-based dialogues. While the path to improvement is nuanced, our findings provide a sturdy foundation for future research and development in this field. The interplay among model components, context considerations, and training duration will continue to shape the evolution of language-driven interactions with virtual agents, not only in Minecraft but also beyond.

## Acknowledgement

We would like to extend our sincere gratitude to Kranti Chalamalasetti for the invaluable guidance and support provided throughout the course of this project. Kranti's expertise and insights were indispensable to our work, particularly as we endeavored to replicate the findings of the original paper. The constructive feedback and continuous encouragement we received have been instrumental in the successful realization of our research objectives. We are profoundly thankful for the opportunity to learn under Kranti's guidance.

## References

[1] Jayannavar, P., Narayan-Chen, A., & Hockenmaier, J. (2020). Learning to Execute Instructions in a Minecraft Dialogue. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 2589-2602). Online: Association for Computational Linguistics. D10.18653/v1/2020.acl-main.232

[2] Hu, H., Yarats, D., Gong, Q., Tian, Y., & Lewis, M. (2019). Hierarchical decision making by generating and following natural language instructions. In Advances in Neural Information Processing Systems 32 (pp. 10025–10034).

[3] Suhr, A., Yan, C., Schluger, J., Yu, S., Khader, H., Mouallem, M., Zhang, I., & Artzi, Y. (2019). Executing instructions in situated collaborative interactions. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 2119–2130). Association for Computational Linguistics.

[4] Kim, J.-H., Kitaev, N., Chen, X., Rohrbach, M., Zhang, B.-T., Tian, Y., Parikh, D., & Batra, D. (2019).

CoDraw: Collaborative drawing as a testbed for grounded goal-driven communication. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 6495–6513). Association for Computational Linguistics.

[5] Ilinykh, N., Zarrieß, S., & Schlangen, D. (2019). Meet Up! A corpus of joint activity dialogues in a visual environment. In Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers.

[6] Chen, D., & Mooney, R. (2011). Learning to interpret natural language navigation instructions from observations. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (pp. 859–865).

[7] Artzi, Y., & Zettlemoyer, L. (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. Transactions of the Association for Computational Linguistics, 1, 49–62.

[8] Andreas, J., & Klein, D. (2015). Alignment-based compositional semantics for instruction following. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1165–1174), Lisbon, Portugal. Association for Computational Linguistics.

[9] Long, R., Pasupat, P., & Liang, P. (2016). Simpler context-dependent logical forms via model projections. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1456–1465), Berlin, Germany. Association for Computational Linguistics.

[10] Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I.D., Gould, S., & van den Hengel, A. (2018). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018(pp. 3674–3683). IEEE Computer Society. DOI: 10.1109/CVPR.2018.00387.

[11] Thomason, J., Murray, M., Cakmak, M., & Zettlemoyer, L. (2019). Vision-and-dialog navigation. arXiv preprint arXiv:1907.04957.

[12] Narayan-Chen, A., Jayannavar, P., & Hockenmaier, J. (2019). Collaborative Dialogue in Minecraft. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 5405-5415). Florence, Italy: Association for Computational Linguistics.

## Appendix: Individual Contributions

### A: Contributions of Mackwyn Quadras

In this project, my role, Mackwyn Quadras, was to explore and implement aspects that were left untouched by the original authors, primarily focusing on Long Short-Term Memory networks (LSTMs). I developed and ran comparative studies between LSTMs and Gated Recurrent Units (GRUs), aiming to understand the unique strengths of each model and gaining a deeper insight into their structures and functionalities.

I introduced new and more precise metrics like the Jaccard similarity to measure the model's performance, providing a detailed understanding of how well our model was performing. Additionally, I also implemented advanced data prefetching methods to make the training process more efficient and resource-friendly, ensuring a smoother development process.

I also explored various activation functions, analyzing and comparing them to understand their impact on the model's learning ability and overall performance. I was responsible for maintaining the main GitHub repository, ensuring it's up-to-date and organized, allowing for a collaborative and harmonious working environment.

This project has been a significant learning journey for me, allowing me to deepen my understanding of deep learning models, and enriching my knowledge about their complexities and capabilities, all the while maintaining a professional and collaborative approach.

### B: Contributions of Stanley Joel Gona

In this project, I had the privilege to integrate the attention mechanism into our existing Seq2Seq model and transition the embeddings from GloVe to Word2Vec, comparing the performances to ensure optimal results. My primary focus was on implementing and unraveling the intricacies of LSTMs in our model.

This project was not just a task; it was a journey of discovery and learning, enabling me to delve deeper into the profound realms of deep learning. It marked my initiation into projects of extensive scale and complexity. The process of replicating and extending a project is a daunting endeavor, requiring meticulous attention to detail and a profound understanding of the original work.

It's been a challenging yet enriching experience, offering me invaluable insights into the nuances of deep learning models and sharpening my analytical prowess. This journey has been about learning, discovery, and pushing my boundaries, enhancing my comprehension of AI and deep learning.

Through the twists and turns of this project, I've gained more clarity on the intricate workings of AI

Repository https://github.com/GrimGamer1999/BAP-unipotsdam$_{ı}mgs$

and deep learning, and it has fueled my curiosity to explore more in this captivating field of study. The challenges and the learning from this project have been a beacon, guiding me towards more profound insights and innovations in the realm of AI.