Hack Research

Proceedings of the 4^{th} Hack Research Hackathon
University of Texas Rio Grande Valley

$\mathcal{H}ack\mathbb{R}$ 2023

Tim Wylie, Editor

Preface

This year was, without a doubt, the most successful Hack Research event so far, which was likely due to several factors. The first may simply be a greater amount of funding this year thanks to an exploreCSR grant from Google. We changed the location to an environment that felt more involved and community focused, had several better-focused problems with more faculty involvement, bigger prizes, and we increased the number of students to 100.

With that in mind, Hack Research proved to be a good platform to begin engagement and to find problems that are accessible for new students in order to bring them into different research areas. With greater faculty participation, there was a massive variety of problems. With so many, we actually split the problems in groups and had four rooms with faculty stations to answer questions.

The goal of Hack Research is to have a competition focused on algorithmic and theoretical skill development rather than software. This gives students who excel in these areas an opportunity to apply that knowledge, and additional development to students that are not as familiar. The event also provides a meaningful way to connect with faculty and their research. This focus, however, does not mean there is a lack of software development; many of the problems were based in software with machine learning, data mining, and visualization applications. These problems target the development of research skills rather than a single unsolved question.

The questions are posed by faculty and students in various research groups. Finding and proposing problems requires a difficult balance between it being interesting and yet approachable. Attempting to find such open problems is a struggle when the problems should be nontrivial.

Many people contributed to the success of the event, and I have tried my best not to miss anyone in the acknowledgements section. This year we were given a major sponsorship through Google, and thus did not need further sponsorship beyond them and the university. Finally, I am indebted to the support of the CS department, Lisa Moreno, and Odette Perez.

November 11-12, 2023
Edinburg, TX, United States

Tim Wylie, Editor

Acknowledgements

This event would not have been possible without the help of many people of whom we are extremely grateful to know. Without a doubt, the event would not have been possible without the understanding and support of the Computer Science department. They have consistently supported and helped organize, fund, and create problems for the event.

As part of that support, Lisa Moreno and Odette Perez, were instrumental in the organization and planning. Hack Research is now too large to be organized entirely without help, and they were the key help needed. They dealt with venue reservations, purchasing, food, delivery, and all accounting. They also answered nonstop questions from me and dealt with problems when I didn't have an answer for them.

The problems really are the key to making the event different, challenging, and fun for the students. Many of the faculty not only contributed problems, but also a lot of time answering questions, talking with students, and judging. In alphabetical order, problems were contributed by Marzieh Ayati, Bin Fu, Yifeng Gao, Qi Lu, Eric Martinez, Robert Schweller, Emmett Tomai, Tim Wylie, and Li Zhang.

Of special note, Eric Martinez created several collaborative problems with students from the Medical school with great success. He brought in several students and resources that were beneficial for the CS students and helped many students get a jumpstart in generative AI and integration.

The participating students also deserve acknowledgement for putting in many hours of hard work simply for the challenge and joy of it. The event would not function without their willingness to dedicate a weekend to being exhausted and asked to tackle problems they have no experience with.

I must also thank my family, who have always been supportive despite the amount of time this event takes. Their understanding and patience can not be overstated.

Finally, we are grateful for the sponsors who gave us the support we needed to make the event a reality. As mentioned, the Computer Science department at UTRGV backed the effort both monetarily and with the support of faculty and staff, and Google provided funding for outreach events through exploreCSR.

Sponsors

www.utrgv.edu/csci

research.google/outreach/explorecsr/

Organization

$\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023 was organized by the Algorithmic Self-Assembly Research Group (ASARG) with the help of the Department of Computer Science at the University of Texas Rio Grande Valley. The event was graciously hosted by UTRGV in Edinburg, TX, and funded by an exploreCSR grant from Google.

The Program Committee are the faculty that judged the submissions. Those faculty also submitted problems along with the faculty listed in Volunteers. The ASARG members are default volunteers.

Director

Tim Wylie	timothy.wylie@utrgv.edu	University of Texas Rio Grande Valley

Program Committee

Marzieh Ayati	marzieh.ayati@utrgv.edu	University of Texas Rio Grande Valley
Bin Fu	bin.fu@utrgv.edu	University of Texas Rio Grande Valley
Yifeng Gao	yifeng.gao@utrgv.edu	University of Texas Rio Grande Valley
Qi Lu	qi.lu@utrgv.edu	University of Texas Rio Grande Valley
Eric Martinez	eric.m.martinez02@utrgv.edu	University of Texas Rio Grande Valley
Robert Schweller	robert.schweller@utrgv.edu	University of Texas Rio Grande Valley
Emmett Tomai	emmett.tomai@utrgv.edu	University of Texas Rio Grande Valley
Li Zhang	li.zhang@utrgv.edu	University of Texas Rio Grande Valley

Volunteers

Odette Perez
Lisa Moreno

ASARG Members

Elise Grizzell
Alberto Avila Jimenez
Ryan Knobel
Aiden Massie
Adrian Salinas

Results

Hack Research is still experimental, and therefore registration was restricted to maintain a small size. The event had around 70 students participate in the 24-hour event. Of those students, 20 students submitted a paper to compete for the prizes. In total, there were 9 submissions. Seven professors contributed problems, and the faculty attended some portion with several staying the majority of the time in order to encourage and assist students. Those faculty are listed in the program committee (who also judged the submissions), and as volunteers.

Winners

There were many quality papers submitted. The judges narrowed it down to these papers as the top submissions based on quality, effort, teamwork, and difficulty.

1. First Place
    - Innovations in Mental Health Care: Automated Clinical Note-Taking with Artificial Intelligence
      Mario Trevino, Alan Lopez, Sridhar Srinivasan, Lesley Chapa
2. Second Place
    - The Ants Won't Go Marching
      Izabella Valero, Tyler Morgan, Jose Amaro
3. Third Place
    - AI Powered Analysis of Call Center Interactions
      Ethen Sanchez, Julio Maldonado
4. Fourth Place
    - Coin Flips and PATS
      Adrian Salinas, Alberto Avila Jimenez
5. Fifth Place
    - Time Series Pattern Hunter
      Johann Cruz, Juan Perez, Ryan Knobel, Gaukhar Nurbek

Table of Contents

Innovating Narrative Interactivity in Video Games with Large Language Models

Felix Chavez *

Abstract

The evolution of narrative interactivity in video games has lagged behind the dynamism of gameplay mechanics. This paper explores the incorporation of Large Language Models (LLMs) as dynamic Dungeon Masters (DMs) in text-based role-playing games (RPGs), leveraging OpenAI's GPT models. The proposed system dynamically generates context-aware narratives and dialogues, offering a level of interactivity in story elements traditionally confined to pre-scripted scenarios. This approach emulates the complex decision-making and storytelling prowess of human DMs in traditional Dungeons Dragons (DD) campaigns, potentially revolutionizing narrative experiences in digital RPGs.

1 Introduction

The constraint of linear storytelling in games not only hinders player agency but also limits replay value and emotional investment. By addressing the lack of narrative dynamism, we can unleash the potential for games to offer not just a story to be told but a personal odyssey that unfolds through the player's unique decisions and actions.

While video games excel in delivering interactive and dynamic gameplay, their narrative components have remained relatively static, bound by pre-written scripts and limited dialogue trees. This disparity has curtailed the player's ability to influence the story meaningfully. The use of LLMs like OpenAI's GPT to simulate a DM's role in a text-based RPG environment presents a novel solution. By dynamically generating narrative content, this system seeks to provide an unprecedented level of narrative freedom and responsiveness, akin to a human DM's flexibility in a DD campaign.

Historically, video game narratives have been a one-way street, with players experiencing stories rather than shaping them. This static approach to storytelling is increasingly at odds with the interactive potential of modern gaming. Here, we explore the pioneering use of LLMs to bring the flexibility and spontaneity of tabletop role-playing into the digital realm, allowing every player to truly craft their own legend.

The dichotomy between the richness of gameplay mechanics and the static nature of game narratives represents a critical challenge within the game development industry. This study introduces a system that promises to harness the sophisticated capabilities of LLMs to cultivate a dynamic narrative landscape, enabling players to truly become the architects of their own stories within the gaming universe.

The prevailing issue in contemporary video gaming is the rigid narrative structure that confines players to a passive role, often leading to a predictable and unvarying gaming experience. This paper addresses the pressing need for an interactive dialogue system that evolves and adapts organically to the player's journey, enhancing the depth and authenticity of the gaming experience.

2 Implementation Evaluation

Implementation: The AI DM, powered by OpenAI's GPT model, maintains a dynamic database of game states, character relationships, and story arcs. It generates dialogue and narrative choices in real-time, aligned with character backstories and the players' previous interactions. This responsive dialogue system emulates the adaptive storytelling found in table-top RPGs, bringing a new depth to NPC interactions.

The AI DM is engineered with a multi-threaded narrative engine at its core, leveraging the advanced natural language processing capabilities of OpenAI's GPT models. This allows the AI to manage a narrative that is as complex and evolving as any human-led campaign, with the added benefit of being infinitely scalable and responsive.

The AI DM is meticulously crafted to interpret and process an extensive array of variables in real-time, assimilating player choices, character histories, and the overarching plot to construct a coherent and adaptive narrative stream. It acts as a central nervous system for game storytelling, integrating the player's journey with the game's lore and mechanics to produce a seamless interactive narrative.

Evaluation: Preliminary evaluations of the AI DM system focus on narrative coherence, consistency with game state changes, and player engagement. User feedback suggests that the system successfully balances complex game mechanics with compelling storytelling, significantly enhancing the RPG experience. Our rigorous evaluation methodology employs a dual approach, balancing objective metrics of narrative consistency and player engagement with subjective analysis of player satisfaction and emotional engagement.

This comprehensive evaluation framework ensures that our AI DM stands up to the scrutiny of both the analytical and the anecdotal.

The evaluation protocol for the AI DM incorporates a multifaceted analysis, blending quantitative assessments

*Department of Computer Science, University of Texas Rio Grande Valley, name1@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

of narrative logic and user engagement with qualitative research on player immersion and narrative satisfaction. This methodical assessment ensures that the AI DM not only conforms to narrative expectations but also enriches the player's experience with meaningful story development.

3 Discussion

Theorem 1 *The implementation of LLMs as DMs showcases the potential for AI to craft interactive and personalized narratives in gaming. This system not only responds to player inputs but also shapes the story's progression, reflecting the actions and decisions of the player, thus creating a deeply personalized gaming experience.*

The deployment of LLMs as DMs opens a dialogue about the future of AI in creative domains, challenging the conventional wisdom about the capabilities of machines in understanding and facilitating human creativity. It asks us to reconsider the role of AI not just as a tool but as a collaborator in artistic expression.

This pioneering application of LLMs as DMs ignites a conversation about the evolving role of AI in interactive media. It redefines the boundaries between player and game, proposing a future where narrative fluidity and player agency are central to the gaming experience, creating a truly interactive and malleable story canvas.

Balancing Freedom and Restrictions is a significant challenge faced finding the right balance between giving players open-ended freedom and maintaining certain restrictions for game balance. Too much freedom can lead to a chaotic game experience, while too many restrictions can stifle creativity and player agency. Striking this balance is crucial for enjoyable gameplay.

Through trial and error, I had to adjust the AI's response generation to ensure it provided creative yet logically consistent options to the players. Adjusting the temperature setting of the model for creative responses and tuning the context-aware narrative generation would have been key areas of focus. Giving initial prompts and responsibilities like "Professional Dungeon Master" to the AI and removing said prompts lead to drastic changes in what actions were allowed and which ones lead to screens asking the player to play correctlty.

Incorporating player feedback into the development cycle is essential. Player reactions and suggestions provide invaluable insights into how well the balance between freedom and structure is being received. Dealing with the wide range of player inputs, especially in an open-world format, must have required continuous tweaking of the AI's understanding and response mechanisms. Ensuring that the AI DM can handle unexpected or unconventional player decisions while maintaining narrative coherence would have been a significant aspect of my development work.

4 Conclusion

Incorporating LLMs as DMs marks a pivotal advancement in narrative design for RPGs, suggesting a future where game stories are as dynamic as their gameplay. Ongoing research will aim to fine-tune the AI DM's comprehension of intricate game states and its predictive capabilities, ensuring that it can consistently deliver a narrative that adapts to and anticipates player behavior.

The integration of LLMs into the heart of RPG narrative design heralds a new era where the stories within games can live and breathe alongside their players. As we look to the future, we anticipate further refinements in AI narrative comprehension, emotion modeling, and multi-modal integration, all of which promise to deepen the connection between players and the worlds they inhabit.

The venture into LLMs as DMs signifies a transformative shift in narrative construction within RPGs, promising a landscape where game narratives are as alive and responsive as the gameplay itself. Future investigations will delve into enhancing the AI DM's narrative foresight and emotional intelligence, endeavoring to deliver a storytelling companion that not only understands but also anticipates the player's narrative desires and motivations.

My research and development process in creating an AI-powered RPG system reflects a deep understanding of both the technical and narrative aspects of game design. The challenges of balancing open-ended player freedom with necessary game restrictions highlight the complex interplay between AI capabilities and human creativity in game development. This pioneering work opens up new possibilities for the future of interactive storytelling in video games.

References

Using Large Language Models for NPCsd

Gerardo Aguillon Jr, Hector Sustaita, Damian Gomez and Gabriel Lira

Abstract

In this project, we utilize OpenAI's Learning Language Model, named GPT, to facilitate the conversational dialogue between the user and NPCs (non-playable characters).

1 Introduction

In Section 2, we'll check out what other researchers have done in this area so we can get a baseline understanding. This helps us see where our work fits in. Then, in Section 3, we'll explain the important parts of our project and show the results we got by using OpenAI's Learning Language Model, GPT. We're using GPT to make conversations between users and non-playable characters (NPCs) in games or virtual settings more interesting. In Section 3, we'll explain how we did it and show what happened. In Section 5 we'll sum up what we found, talk about why it's important, and suggest ideas for more research. This helps others see what they can explore next in the world of better conversations using technology like GPT.

2 Related Work

The paper presents SCENECRAFT, a system for creating story scenes in intelligently narrating recreations. Customarily, making energetic and locks in story-based encounters in diversions includes manual origin for the non-player character (NPC) intelligence. SCENECRAFT leverages expansive dialect models (LLMs) to robotize the era of NPC intelligence based on normal dialect enlightening. The system translates enlightening related to scene goals, NPC characteristics, area, and story varieties. It employs LLMs to adjust created diversion scenes with the author's expectation, making branching discussion ways that adjust to player choices while keeping up the specified interaction objectives. The LLMs produce interaction scripts, extricate character feelings and emotions, and change over discoursed into an amusement scripting dialect. The adequacy of SCENECRAFT is illustrated through empirical evaluation, displaying its capacity to make story encounters that show inventiveness, versatility, and arrangement with authorial information.

3 Our Results

Looking into the results, we expected to come across a few issues, and most of the issues were solved. The issues included getting everyone's system to run the same code on Visual Studio, prompt engineering, and finding the right terms to limit our LLM. We also had trouble with the API, facing difficulty in making some calls and formatting responses to pass them to other functions (Parsing).

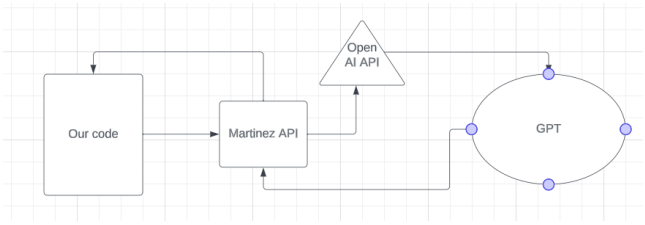


Figure 1: Enter Caption

4 Conclusion

In conclusion, our project is the best give us prize.

References

https://www.gradio.app/docs/group

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

AI in Social Dynamics: Using Large Language Models for NPCs

Alejandro Acosta *

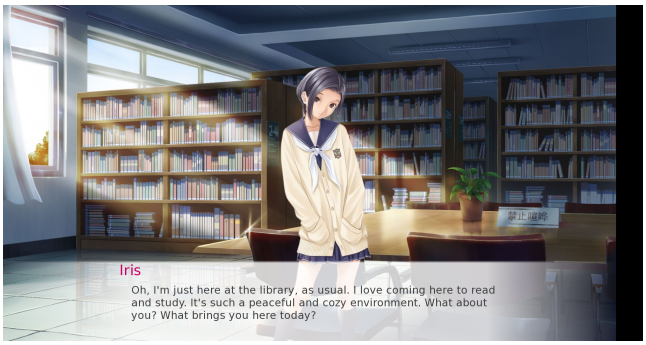


Figure 1: NPC Using GPT-Generated Responses

Abstract

This paper examines how Artificial Intelligence(AI), specifically large language models (LLMs), is changing the way we interact socially in video games. We focus on LLMs used in non-player characters (NPCs) in games. These AI-powered NPCs create more realistic and engaging conversations, helping players feel less isolated and providing them with virtual companionship. The study investigates how AI characters in video games can address current issues of social isolation by enhancing players' social skills and emotional well-being. By offering interactive and empathetic experiences, these AI-driven NPCs provide a form of virtual companionship that's increasingly relevant in our digitally-dominated world.

1 Introduction

In recent years, the rapid advancement of artificial intelligence has opened new possibilities in various sectors, including entertainment and mental health. The development of large language models (LLMs) like GPT-3 has revolutionized the way we interact with technology, making AI more accessible and versatile. A significant application of this technology is evident in the realm of video gaming, where LLMs are used to power non-player characters (NPCs), offering a more immersive and interactive experience. This innovation holds significant potential in tackling prevalent issues like the lack of human interaction, which is increasingly becoming a concern in modern society. By providing engaging and interactive experiences through AI-powered NPCs, it offers a novel approach to reducing the effects of limited human contact and enhancing overall mental well-being. The integration of LLMs in NPCs presents an opportunity to not only enhance gaming experiences but also to contribute positively to players' social skills and emotional health.

*Department of Computer Science, University of Texas Rio Grande Valley, alejandro.acosta03@utrgv.edu

2 Related Work

In the realm of interactive applications, the concept of generative agents as believable replicas of human behavior has seen significant development. The referenced work demonstrates the implementation of these generative agents by introducing a sandbox environment similar to "The Sims." In this simulated setting, twenty-five AI-driven agents are programmed to exhibit complex behaviors such as planning their daily activities, engaging in social interactions, forming relationships, sharing news, and coordinating group events. This allows users not only to observe but also to interact and influence the agents' decisions and actions, offering a deeper insight into the capabilities of AI in mimicking human-like social dynamics. [2]

3 GPT Model

The creation and personalization of NPCs was achieved through a technique known as prompt engineering. This method involves crafting specific prompts that guide the GPT 3.5 model in generating distinct personalities for each NPC. By carefully designing these prompts, each NPC is given unique characteristics and traits, creating a diverse range of virtual characters. The dialogues for these NPCs were dynamically generated by the model, ensuring that each interaction reflected the NPC's individual personality and affected by various game states:

- The conversation context is shaped by different game states.
- NPC dialogue changes based on the player's visited map locations.
- Interactions with NPCs are recorded and tracked.
- Game state-specific special events influence gameplay.

4 NPC Role-Playing Games as a Solution

4.1 Immersive, Realistic Interactions

GPT-powered NPCs in role-playing games (RPGs) offer a new level of immersive and realistic interactions. These NPCs can engage players with dynamic and context-aware dialogue, enhancing the storytelling and emotional depth of the game. Unlike traditional scripted NPCs, GPT-enabled characters can adapt their responses to the player's actions and choices, creating a more personalized gaming experience.

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

4.2 Emotional Fulfillment and Social Practice

RPGs with advanced NPCs can provide emotional fulfillment by allowing players to form virtual relationships and undergo social scenarios in a controlled environment. For individuals facing social isolation or those looking to improve their social skills, these games can serve as a practice ground. Engaging with AI-driven characters in complex social situations can help players explore various interpersonal dynamics without the pressure of real-life consequences.

4.3 Ethical Considerations

The use of GPT-powered chatbots raises ethical questions, particularly concerning emotional dependency and the replacement of human interaction. There is a risk that reliance on AI for social and emotional support might lead to diminished human contact, potentially exacerbating the issue of social isolation in the long run. Additionally, the management of personal data and privacy is a significant concern.

5 Examples

Recent advances in artificial intelligence, especially in conversational AI tools such as ChatGPT, show promising potential in addressing issues of loneliness and social isolation. A study featured on NeurologyLive highlights that for older adults, particularly those with mild cognitive impairment (MCI), ChatGPT offers a new approach to alleviate loneliness and maintain a sense of social connection. The AI model's humanlike conversation capabilities enable engaging and meaningful interactions, acting as a virtual companion and offering emotional support. This can be particularly beneficial for those experiencing communication difficulties due to cognitive decline, as ChatGPT can adapt its communication style to match the cognitive abilities and interests of the user. [3]

A study by Cornell University found that chat tools with AI like ChatGPT make conversations more efficient and positive. But, if people think someone is using AI to talk, they might see them as less friendly or connected. This means AI can make chatting better but might change how we talk and how others see us. [1]

6 Conclusion

In summary, GPT-driven NPCs in RPGs offer promising advancements in creating emotionally engaging and socially enriching gaming experiences. They not only enhance the realism of the game world but also provide a unique space for players to explore and develop social skills.

Figure 2: Dialogue tailored to the character's personality.

References

[1] D. D. Z. A. H. M. K. L. M. N. J. H. . M. F. J. Jess Hohenstein, Rene F. Kizilcec. Artificial intelligence in communication impacts language and social relationships, 2023.

[2] C. J. C. M. R. M. P. L. M. S. B. Joon Sung Park, Joseph C. O'Brien. Generative agents: Interactive simulacra of human behavior, 2023.

[3] B. W. P. Xiang Qi, PhD. Chatgpt: A promising tool to combat social isolation and loneliness in older adults with mild cognitive impairment, 2023.

Using Large Language Model's for NPC's

Esteban Pena * Christoper Hinojosa † Alexa Barrera ‡

Abstract

AI interactions could revolutionize the gaming space by creating dynamic AI that uses LLM's to create unique experiences for the player by creating not only almost human interaction for the player, but also can create scenarios that the NPC could react using to the Transformers that can change the interactions that the between the AI and the player. When creating the AI it learns with each interaction and that creates more options for the player to interact with them and creates a relationship between the player and the AI. In the sense of this research we have created a text-based AI that interacts with the player and with different distinguished base tratis of personalities and story is going to push the idea of this research.

1 Introduction

In modern narrative games the sense of simulated-realism is portrayed to a user based on its' fictitious yet captivating scenarios and decision based mechanics that can create different branches in the main story. NPC's (Non-Playable Characters) are a key element in story-driven games because the their interactive nature based upon pre-selected dialogue. Although the freedom of choice and action may seem present on the surface, these games have limitations to how far the user can expand their journey due to all possible branches leading to a set number of options/opportunities. In order to break this boundary and create a more dynamic environment for the player, these games would have to constantly update its' elements based on the user's choices. NPC dialogue can affect the main story by building/breaking relationships and create new possibilities like quests for the player to experience. Our challenge is to use large language modals (LLM), such as ChatGPT, to generate cohesive dialogue between and player and an NPC to expand the limits on usual narrative game play.

We briefly highlight some related work in Section 2, and then provide the definitions and results of our work in Section 3. We then conclude in Section 4 and point towards the general research goals for this work [3].

2 Related Work

Since artificial intelligence is still in its early stages, new research studies are implementing generative NPC dialogue to make a "realistic" experience for a user. (LLM) are powerful and adaptive modules that can retrieve a script and its ai can generate an appropriate response to that input. The more generative the ai modal can deliver, the more coherent the NPC will make itself for the narrative aspect in the game.

Implementation for this concept has been done by feeding the (LLM) specific data about the user and NPC characters, like their background, location, and role in the story. The power of the modal was further expanded in a research study about automating scene generation in a game called scene craft. The (LLM) identified specific emotions and gestures corresponding to each dialogue (Kumaran, 2023). This is important to notice because it proves the interaction from player to non-player can be "human-like" regarding to the tone of the narrative.

If there are multiple non-playable characters, then developing different characterizations for each would also create a more dynamic environment, leading to more possibilities on how the user can affect the story. Each unique NPC is created to fulfill a purpose (Gao and Emami, 2023). This study puts focus on the relationships these NPC's have on the user. If the player interacts with an NPC who is non-friendly then the player would have to gain their trust.

For these studies to feed the AI modal the data sets for the non-player characters the artificial intelligence would need to be correctly implemented. When a network request is sent to the ChatGPT API it responds with a object like JSON (Artus and Robert, 2023). These objects are able to instantiate the data within each request. This research study had a JSON object that had properties such as name, description, weapon, and size. Some of these would be key characteristics the api would need to remember when the user interacts with that specific NPC character.

This API is powerful tool that can not only generate how NPC's adapt, but can also create simulated worlds based upon the preset data sets. As mentioned earlier in this section, since AI is in it early stages problems can easily occur. Evaluating if this improves workflow for simulation designers. (Johnson-Bey, 2023) Testing if this route of new narrative game implementation is consistent can further prove of disprove if it goes beyond the limits of what is available to us in the modern day.

3 Our Results

When acquiring the results through our research we have gained new knowledge where we have a shown the power of AI. We have implemented ChatGPT to make calls to its API. This tool was revolutionary and should be researched further to create a more life like experience when

*Department of Computer Science, University of Texas Rio Grande Valley, esteban.pena05@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, christopher.hinojosa06@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, alexa.barrera01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

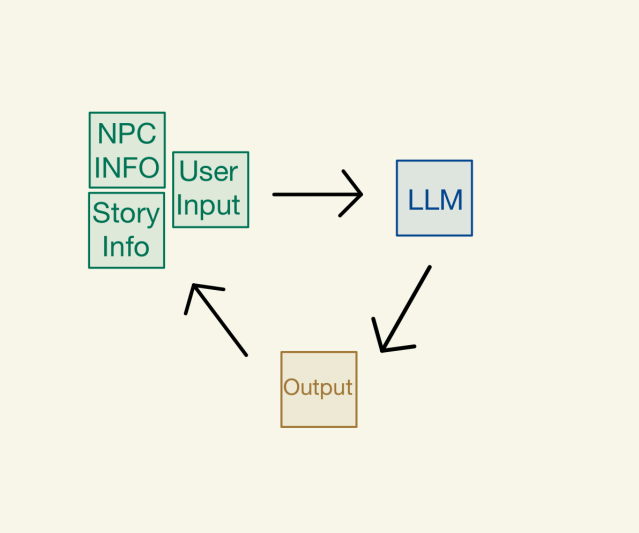


Figure 1: Flow Chart

experiencing NPC's in a new light.

We have first implemented the ChatGPT API into our code, to create the NPC responses. The AI creates generated scenarios with the given characters interacting within the world. This is going to be supported by a template of given NPC's where they have their basic stats (Background, Age, Morality, Relation) and a summarized story that is going to be fed through the AI. A combination of these 2 factors is going to create a life-like and dynamic AI that is going to be reacting to the player's input. To keep this ideology, we save the responses and the code makes the AI aware of the conversations, scenarios, and the world around them. This information is needed is a must to have an NPC that is going to be dynamic and stray away from the basic pick a choice and get the pre-generated response from the NPC.

4 Conclusion

When questioning ourselves if creating AI that is dynamic using Transformers. This would create a more lifelike experience when interacting with NPC's that stray from hard coded responses that give the AI a very robotic and repetitive responses that don't give much meaning to the player because this doesn't show any sort of genuine response, but when using AI we have those feedback that is lively. When using ChatGPT to create responses and the the scenarios this is going to not only change how the AI reacts, but this is going to change the situations of the world around them and the interactions that the NPC is going to reacting according to the awareness of their surroundings.

You have more freedom when doing this that your options become almost infinite on the scale of choices that you can from the story lines that generate within the storyline and this makes the NPC's not just something that you dread to, but the human experience and seeing this furthermore within the storyline that it remembers conversations to create a full experience of interactions.

References

[1] Q. C. Gao and A. Emami. The turing quest: Can transformers make good npcs?, 2023.

[2] S. Johnson-Bey, M. Mateas, and N. Wardrip-Fruin. Toward using chatgpt to generate theme-relevant simulated storyworlds., 2023.

[3] V. Kumaran, J. Rowe, B. Mott, and J. Lester. Scenecraft: Automating interactive narrative scene generation in digital games with large language models, 2023.

[4] A. Umbraško and R. Drury. Applying chatgpt in ai-based dynamic video game narrative generation system, 2023.

[3] [1] [4] [2]

Identifying Common Errors in CS1 Lab Work

Israel Lopez * Paula Serrano † Francisco Orta ‡ Richard Tapia §

Abstract

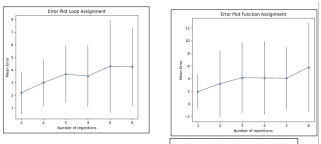
In our Hack Research project, we utilize ChatGPT to identify common errors in CS1 lab submissions. We employ a refined process, issuing prompts to limit errors based on a predefined list. Challenges such as mode collapse and output variability are addressed by running the program multiple times. Our taxonomy of errors includes Syntax, Logic, Semantic, Runtime, I/O, Configuration, UI, and Performance errors. Metrics are assessed, and additional attributes are extracted. To ensure reliability, we compile errors from multiple executions and calculate scores, identifying the least understood topics. We employ the elbow criterion and silhouette scores for clustering analysis. The results and methodologies provide insights into effective error identification in CS1 lab work.

1 Introduction

In the context of CS1 student submissions, our process involves systematically identifying prevalent errors. We will then annotate each submission with the specific mistakes made and evaluate the system's performance using a separate set of submissions. Our initial step is to pinpoint common errors, misconceptions, and variations of the same mistake. Subsequently, we will annotate the submissions with these identified errors.

2 Introduction / Reliability of the model

We presented ChatGPT with a prompt to identify errors in CS1 lab submissions and provide a general description of these errors. Subsequently, we refined the process by issuing another prompt, limiting errors and classifications based on a predefined list of known errors. This process was later deprecated due to the limited options. When trying with free outputs,a challenge arises as each execution of the process yields different outputs. Sometimes, the output contains a comprehensive explanation of errors, while at other times, it may be limited or not contain all the required data. To address this variability, we have automated ChatGPT to repeat the steps and compare results, aiming to minimise inconsistencies. Moreover, we have recognized that a single execution may not reveal all possible errors, prompting the need for durations of the process the initial results revealed a mode collapse,



1.1 Loop Repetitions v. Mean Error 1.2 Function Repetitions v. Mean Error

indicating a dominance of a single or a limited set of outputs. To address this, we reran the program to obtain a new set of results, enabling us to compare both sets and ensure the accuracy of our findings. The program is designed to be run multiple times (n times), allowing for comprehensive comparisons with other results to enhance reliability.

From both graphs we can see that there is some stability in between repetitions 3 and 4. For the Loop Assignment the error decreases after 3, and for the Functions Assignment it seems to be stable, although there is a small downwards slope. For this reasons we have chosen 4 repetitions, and we can say that the number of errors is being identified in a correct manner.

3 Part 1: Non Supervised Classification

We aim to analyze and categorize prevalent errors present in student assignments to pinpoint common misconceptions and identify areas where students may lack a complete understanding. Following this assessment, we plan to provide targeted assistance to these students by offering resources such as videos and other helpful materials related to the identified topics

To determine the optimal number of clusters in a dataset, you can use the elbow method and the silhouette score.

Elbow criterion First, we determined k as the number of clusters formed using the following formula as shown in figure 1.

Where k is the number of cluster formed, Ci represents the i-th cluster, and Xi is the data present in each cluster. Then, we computed the distance using the formula as shown in figure 2.

Quality measurement, We used the Scikit-learn's silhouette score function to calculate the mean silhouette coefficient of all samples. This involved considering the mean intra-cluster distance and the mean nearest-cluster distance for each data point. The silhouette coefficient that we used for our sample is (-)/max(,).

A silhouette score with a value near +1 means the data point is in the correct cluster. A silhouette score with a value near 0 means the data point might belong in some other cluster. A silhouette score with a value near -1 means that the data point is in the wrong cluster.

*Department of Computer Science, University of Texas Rio Grande Valley, israel.lopez05@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, paula.serranosierra01@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, francisco.orta01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, richard.tapia01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

4 Our Results

The taxonomy errors we decided to use are: Syntax Errors, Logic Errors, Semantic Errors, Runtime Errors, Input/Output Errors, Configuration Errors, User Interface Errors, Performance Errors.

Semantic Error numOfGuesses is not initialized guessNum is not initialized Variable is not declared or initialized Variable is not used or updated Using incorrect variable names Incorrect number of guesses or ranges Incorrect variable names Missing or incorrect incrementation Missing or incorrect condition for done Missing or incorrect comments Logical Error srandtime0 is placed inside the while loop Incorrect message condition or output Input validation is missing Inconsistent error messages for different guesses Incorrect placement of srandtime0 Incorrect loop conditions or updating in the loop Missing or incorrect updates inside the loop Missing or incorrect input validation Missing or incorrect number or sequence of conditions Variables not properly updated inside the loop Incorrect outputs or messages Incorrectly coded behavior or logic Incorrect or missing number range Syntax Error Initial output message doesnt match the prompt Using incorrect operator or syntax Missing semicolon or closing quotation mark Missing or incorrect closing braces Missing or incorrect variable concatenation Missing or incorrect capitalization Misspelled variable names Runtime Error Runtime errors and initial restrictions

Assess various metrics and refer to relevant sources. Classify data by file ID and homework type. Consider real-life applications and incorporate professor grades based on given prompts. Extract additional attributes from files, such as line numbers.

5 Part 2: Taxonomy of errors

In conclusion, this research aims to address the challenge of efficiently identifying and common errors among student programming submissions. The approach involves ChatGPT API, to automatically analyze code and generate semi-structured information about the errors. This information is then processed through a well-defined pipeline that includes prompt engineering, clustering techniques, and knowledge representation.

The key innovation lies in the use of ChatGPT for its ability to understand and generate code-related explanations. The semi-structured output obtained from ChatGPT is subjected to clustering techniques, allowing the identification of common error patterns. To enhance generalization, a knowledge representation system is employed to create a taxonomy of errors, providing a structured framework for categorizing misconceptions and variations of errors.

The deliverables of this research include a well-defined processing pipeline, a comprehensive set of common errors organized in a taxonomy, and an evaluation of the labeling performance on a separate test dataset. The techniques explored encompass prompt engineering, clustering, knowledge representation, and standard machine learning classification, demonstrating a holistic approach to automated error identification in student programming submissions.

References

IEEE Xplore Full-Text PDF:, ieeexplore.ieee.org/stamp/stamp.jsp?tp=amp;arnumber=9495808. Accessed 12 Nov. 2023.

Clustering and Visualizing Enrollment

Cassandra Garza Arely Solis Belinda Alvarado

Abstract

This paper explores the pathways of Computer science students through their degree program using data science clustering and visualization techniques. We attempt to identify different groups of students based on their course enrollment patterns, and to evaluate how these patterns affect their academic performance and graduation time. In addition, we also seek to compare the health of the Computer Science program across different years, to detect any "bad paths" or other factors that hinder students' progress, and to predict the class demand for the future.

1 Introduction

Computer science is a rapidly evolving field that requires students to keep up with the latest developments and skills. However, not all students follow the same pathway through their degree program, and some may encounter difficulties or delays in completing their courses. Understanding the factors that influence students' course enrollment patterns and academic outcomes is important for improving the quality and effectiveness of the Computer science program.

In this paper, we use data science clustering and visualization techniques to explore the pathways of CS students through their degree program. We use a data set of CS students' course enrollment from Fall 2016 through Summer ll at The University of Texas Rio Grande Valley, and apply k-means clustering and principal component analysis to group the students and visualize their pathways. We aim to answer the following research questions:

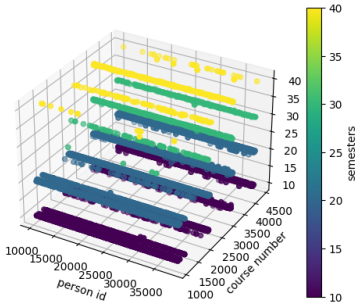
How do CS students differ in their course enrollment patterns and academic performance? How do these patterns affect their graduation time and success rate? How does the health of the CS program vary across different years? How can we predict the class demand for the future?

2 Method and Action

2.1 3-D Visualization

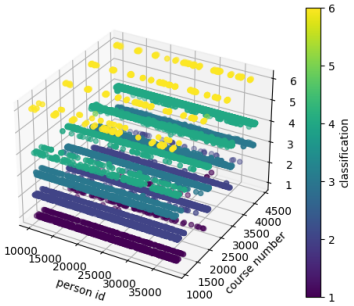
One of the purposes of this study was to explore the relationship between the courses taken by students and the semesters in which they were taken. To achieve this goal, we used 3D visualization techniques to create two plots that displayed the data from different perspectives.

The first plot showed the course number, semester, and students who took each course in each semester. The number of students was encoded as the x-axis,the course number was encoded as the y-axis and the semester was encoded as the z-axis. The plot used a bar chart to represent the data, with each bar having a different color according to the semester. This plot allowed us to see the distribution of students across courses and semesters, and to identify the most popular and least popular courses in each semester.



We can see between the earlier CS courses there is a zigzag pattern between fall and spring semesters. I believe this typically happens because when students start out, there is no urgency to begin taking major specific courses. Unlike in later years, when students are on the verge of graduation, there is spikes in the amount and demand for summer courses. And to reiterate, as for later classes we can see that there increases in frequency the amount of courses taken during the summer semesters.

The second plot showed the course number, student classification, and number of students who took each course in each classification. The number of students was encoded as the x-axis,the course number was encoded as the y-axis and the student classification was encoded as the z-axis. For plotting purposes, the classification values were changed from string values to numerical values, i.e. freshman = 1, sophomore = 2 and so on until Post-baccalaureate = 6. The plot used a scatter plot to represent the data, with each point having a different color according to the classification. This plot allowed us to see the distribution of students across courses and classifications, and to identify the patterns and trends of course selection among different classifications of students.



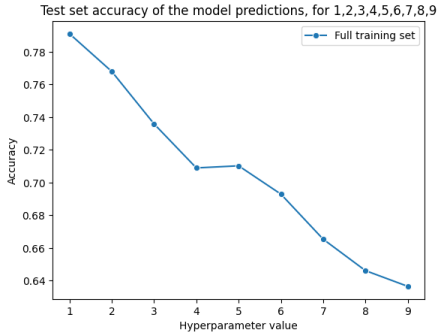
From the figure, we can see there is sort of a curve underneath the plots, we can see that typically beginner CS courses are typically taken by most freshman and sopho-

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

mores, and more advanced courses have few registration from the beginning students. Even more so, by analyzing the the junior class, number 3, of students, not many tend to gravitate towards the the upper level courses until needed. However, for Post-baccalaureate, it stays the same across the board.

2.2 KNN model

From the KNN model we trained on the subset of data containing CS majors that completed their degree in computer science, we found that the model performs better with a smaller k value, the highest being k=1 with an accuracy of approximately 79 percent. As we increase the value of k, the accuracy of the model decreases, dropping below 70 percent.



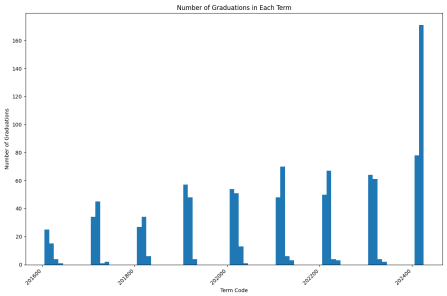
Higher accuracy suggests that students' completion times are more influence by the most similar individual in the dataset. Since larger k values affect the predictions negatively, this is likely because it introduces noise into the predictions.

It is important to note that our data set was limited. Ideally, we would have more data. Additionally, we must be cautious with having a small k value given it can lead to a model that is overly sensitive to noise and overfitted. Having had more time, we would have experimented with other models such as logistic regression, and conducted hyperparameter tuning and cross-validation.

Overall, the results from the model suggest that a simpler model performs better than a more complex one in this case, at least for K-nearest neighbors.

2.3 CS Program Health

"In our examination of graduation numbers across terms, our focus was on identifying the peak semester for graduations. This strategic analysis not only pinpoints the most prolific term but also delves into factors like course availability, academic advising, and external influences. This investigation aids academic planners in tailoring support services and resources to align with student needs, empowering us to optimize the academic environment. By recognizing the peak semester, we proactively respond to student needs, fostering an environment conducive to academic achievement and success."



3 Conclusion

This research project employed a couple of key methodologies to explore the relationship between students' course selections and the semesters in which students took them. Through 3D visualization techniques, we observed an interesting zig-zag pattern for introductory computer science courses between the fall and spring semesters. The second visualization depicted the interaction between course selection and student classification. We observe a trend where beginner courses are predominantly taken by freshmen and sophomores while the advanced ones have fewer registrations from early-stage students.

Subsequently, we trained a KNN model on a subset of data containing computer science majors who successfully completed their bachelors in computer science. The model performed best with a smaller k value, particularly at k=1 with an accuracy of 79 percent. As k increases, the accuracy of the model drops. This suggests that completion times are influence most by individuals who are most similar to each other.

Some limitations we faced are that we do have the exact semester when students completed their degree. We made the assumption that individuals who successfully graduated have completed both CSCI4325 and CSCI4390. Caution is advised when selecting a small k value for our model. This may lead to overfitting. Given more time, we would experiment with models that perform well with continuous data.

We took a closer look at graduation trends across various terms to pinpoint which semesters have the highest graduations numbers. This aids in understanding underlying factors, course availability, and external influences. This information can help tailor support services and resources to optimize the academic environment for student success.

Our findings contribute to ongoing discussions on enhancing student success and retention, offering insights for evaluating student support and fostering an environment conductive to academic achievement and success.

Our github link: https://github.com/belindaalvarado/HackR$_C lustering - and - Visualizing - Enrollment$

The Treasure Hunt Continues

Noah Wylie

1 Introduction

Humans can be quite good at spotting patterns in things. However, there are many cases where a pattern is near impossible for a human to spot with just their eyes. For example, the data set in Figure 1 has a pattern in it, but it is impossible to tell by just looking at it. While not

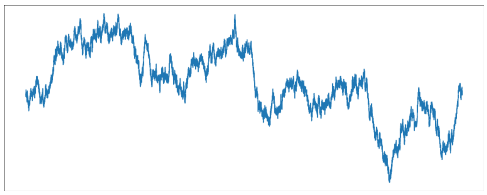


Figure 1: 10 million data points

being able to find patterns in data sets like these might not seem important. However, Many different fields analyze massive data set trying to find patterns that could help them make a break through in whatever they are researching. This paper is focused on researching a way to find the hidden pattern in the massive data set seen in Figure 1. While the hunt for the pattern was not successfully, what was achieved proves it is possible.

2 Related Work

in [1], the author discussed several different algorithms for motif discovery in time series data, as wall as introducing their own method called The Mueen-Keogh (MK) algorithm. In this paper we look at both the The MK algorithm and the Brute Force Motif Discovery, Taking pieces from both to develop the algorithm used in this paper.

3 The Results

The approach we used is very similar to the brute Force Motif algorithm mentioned in [1]. However, there is an added elimination step that is referred to as "Prune" in [1] The algorithm used is limited to comparing individual data points and seeing if they are the same, rather than a cluster of point which seems to be preferable by having higher accuracy. graphing the data set ([10,20,50,20,50,39,1,3,5,20,15,28,100,25,20,50,20,50,49,100]) gives use the following graph seen in Figure 2.

After putting the data set through our method. The result when graphed, as seen in figure3, shows a matching pattern.

4 Conclusion

While we were not able to test massive data set in a reasonable time using our developed algorithm. The small

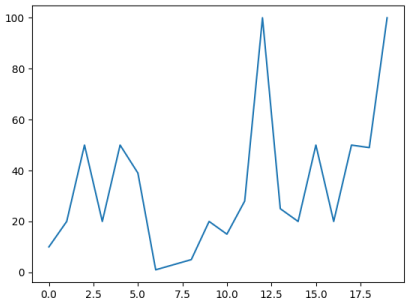


Figure 2: Enter Caption

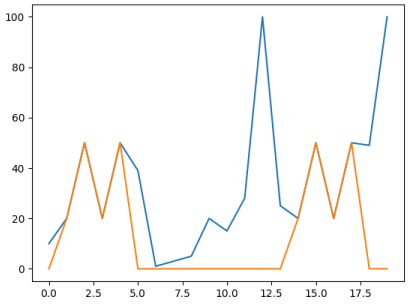


Figure 3: Enter Caption

test case should hopefully show that if the algorithm is developed more that it will find the hidden pattern in Figure 1

References

[1] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. *Society for Industrial and Applied Mathematics.Proceedings of the SIAM International Conference on Data Mining*, pages 473–484, 2009. Copyright - Copyright Society for Industrial and Applied Mathematics 2009; Last updated - 2022-10-20.

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

Recognizing Motifs Hidden within Time Series

Jonathan Ectonrodriguez * Carlos Alvizo † Gustavo Banuelos ‡ Raul Flores §

Abstract

The objective of this project was to identify a repeated sequence found within a large time series and to report its location. Given that the time series provided in the problem was of size 10M, we needed to find an efficient algorithm to sift through the series and report the exact pattern that is repeated and the two points in the series where both sequences lie.

1 Introduction

Patterns/repetitions within a time series are called a "motif" and can be seen as a smaller time series within a larger one. Motifs are crucial to identify as they allow us to draw conclusions from our research; however, finding them in large datasets can be difficult.

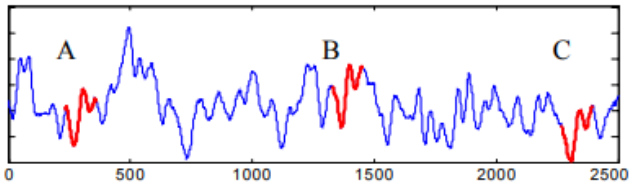


Figure 1: Motif within a Time Series

Although, various methods have proven effective in isolating motifs such as K-motifs and dynamic time warping; an efficient algorithm is required in order to be scalable with the size of the time series. STOMP is the desired algorithm for the task at hand where it utilizes matrix profiles (the distances between each subsequence and its nearest neighbor) to obtain a runtime of $O(n^2)$ and a $O(n^2)$ space complexity whereas other algorithms run in $O(n^2 logn)$ [3].

2 Approach

With an algorithm to find motifs identified (STOMP), we sought out a method in which we could utilize our hardware on the provided problem time series. Our solution was a python library called Stumpy that specializes in motif discovery via GPU-STOMP for larger datasets.

Given that the time series in the problem was in the form of a 2D array (10,000,000:1) we flattened the array to become one-dimensional in order to work with the Stumpy functions. After flattening the array we created the matrix profile (nearest neighbor distance value), which not only allows for motifs to be quickly identified but also allows looks for the nearest neighbor for all subsequences [1].

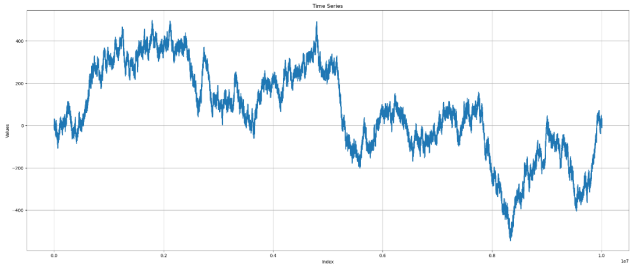


Figure 2: Problem's Time Series

Given the importance of the matrix profile in STOMP, an explanation of how the algorithm works is essential.

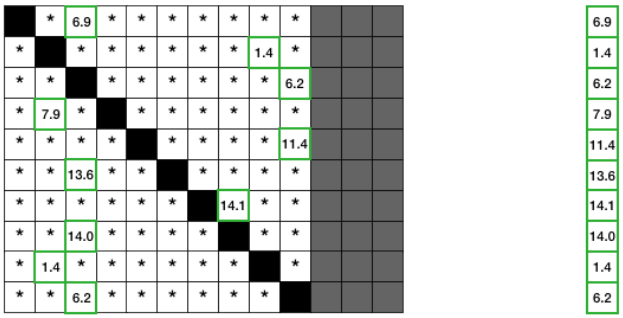


Figure 3: Matrix Profile Example

At the core, a motif can be calculated by obtaining a subsequence and comparing it with all subsequences within a series. This can be further improved upon via the Euclidean distance (distance between two points) in which if we conduct pairwise Euclidean distance on a sequence and all subsequences we can map their distances. This would be repeated several times for every subsequence until filled. However, for every list created per sequence only the lowest distance is of importance as it stores the sequence's motif signature. Obtaining the lowest distance from each list we can create a matrix and store the non-blank values in an array in linear time (matrix profile). This is useful when determining whether a motif exists as a small matrix profile can indicate a pattern in which we can cross-reference with our original matrix to find where the sequences are located int the series [2] [4].

*Department of Computer Science, University of Texas Rio Grande Valley, Jonathan.ectonrodriguez01@utrgv.edu
†Department of Computer Science, University of Texas Rio Grande Valley, Carlos.alvizo01@utrgv.edu
‡Department of Computer Science, University of Texas Rio Grande Valley, gustavo.banuelos01@utrgv.edu
§Department of Computer Science, University of Texas Rio Grande Valley, raul.flores05@utrgv.edu

This is an abstract of work done at $\mathcal{H}ac\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

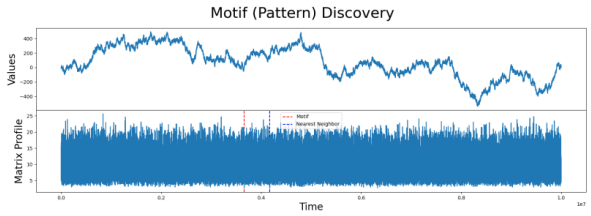


Figure 4: Motif and the location

3 Results

Even with the use of Stumpy's GPU-STOMP, the amount of time taken for the algorithm to run took several hours over an NVIDIA GeForce RTX 3080. We did successfully obtain the motif being repeated as seen in Figure 3 and the areas in the time series in which the two motifs are located; however, it is difficult to see their location specifically since the series is densely packed. However, an estimated index of the two points (motifs) would be around 370k and 410-420k of the array/time series as indicated by the vertical red and blue lines in the latter series in Figure 3.

4 Conclusion

The locating and motif discovery of the hidden two segments in the time series was successful in which we were able to do so as seen in Results. Although this question was to find a motif as an exercise, motif-discovery is essential to recognize patterns that otherwise would go unnoticed. For example, in regards to the study of electroencephalogram (EEG) brain wave time series motifs can be hidden within the series to indicate patterns regarding an individual's brain activity (example provided in problem).

To reflect on the process of the problem, given more time and better hardware, we would liked to have cleaned up our findings more by providing a side-by-side comparison of the motif found as well as a clearer more spread out image of the original time series in order to clearly see the two points that the motifs begin. This was not possible due to the time constraints of the hackathon, the amount of time the algorithm took to run, and the stress the algorithm took on our computers as it resulted in multiple crashes.

References

[1] J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. *Proceedings of the Second Workshop on Temporal Data Mining*, 10 2002.

[2] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322, 2016.

[3] Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, and E. Keogh. Matrix profile xi: Scrimp++: Time series motif discovery at interactive speeds. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 837–846, 2018.

[4] Y. Zhu, Z. Zimmerman, N. S. Senobari, C.-C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh. Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 739–748, 2016.

Time series data analysis using the ”window” method provided unknown parameters.

Steven Villarreal[1]

[1]University of Texas Rio Grande Valley

steven.villarreal03@utrgv.edu

Abstract -

This study aims to implement the ”window” method as a means to search for sections of time series data that are alike. The implementation is carried out on a very large data set while the size and locations of the target and initial sections of the graph are unknown. This specific solution uses the system of ”chunking” as a means to segment the data and decide on an initial sample.

Keywords -

Recursion; Data; Chunking; Analysis

1 Introduction

Using software for analysis of time series data is a modern method for visualizing data, data comparison, trend dissection and carrying out complex calculations on very large sets of data. While its application to specific cases seems to be trivial, the fundamental problem with modern methods, while extremely thorough, are also exceedingly resource intensive in terms of computing resources. A trade-off becomes apparent between level of depth and speed of computation. In other words, an increase in the amount of information extracted from the data will lead to a increase in the overall time until a solution is found. This becomes unavoidable as the size of the data set grows and calculations must be carried out recursively.

2 Model

The following sections describe the use of the Python programming language supplemented by the numPy library to implement the proposed solution.

2.1 Chunking

The format of the initial time series data comes in the form of a one-dimensional array composed of values that correspond to points on a graph represented over time. These values are stored continuously without segmentation spanning the entire file. ”Chunking” aims to break down this array into manageable sub-arrays of a set size or amount of values per sub-array. This can be done since the target sample and it’s size is unknown which grants the user the ability to decide which segment of data is compared to the rest of the graph.

2.2 Candidate Detection

There is increased importance on breaking down the data even further as this method must be applied to a very large data set. The average of each of the points within the scope of the initial and target data segment provides a starting point which may be used as a comparison factor when determining which sections of data are most likely identical to it. The mean of all data points in the initial segment is stored as a single value. The locations of the segments of the time series graph with the closest proximity to the initial mean are stored. Taking note of these sections allows further operations to be run on a smaller and more tuned data set instead of considering the entire collection.

2.3 Filtering

Candidates are compared to the initial segment through a looping process which stores the absolute difference of each data point within the candidate and initial sections and is stored to create a unique line segment titled ”accuracy”. The average of this line segment is stored and coincides with a singular candidate chosen in order from the list of candidates. This process is applied for each likely candidate. A final candidate is chosen based on its ”accuracy” value’s proximity to zero.

2.4 Experimentation

The program initially used only the first segment of data as the object that other segments compare to, through the course of the project, it was found that it was necessary to compare each individual segment with the other segments of the graph as to not exclude a potential match which would otherwise be not tested. The program was altered to run recursively using each ”chunk” as the object to make comparisons with.

2.5 Results

Overall time taken by the *Chunking-Candidate Detection-Filtering-Visualization* process is dependent on ”chunk” size, filtering being the most intensive process with a smaller chunk size taking a greater amount of time. Approximate time taken using a chunk size of greater than 500 was found to be 5-9 seconds. A smaller chunk size such as 100 or less was found to take 12-76 seconds. The initial goal of finding two identical segments in the time series data was not achieved in this span of this project, however, this project demonstrates the underlying system sufficiently.

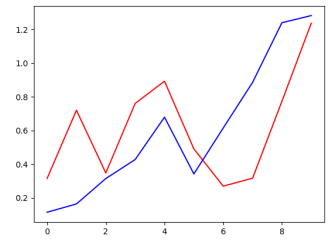


Figure 1. Resultant plot of a single comparison using chunk size of 10.

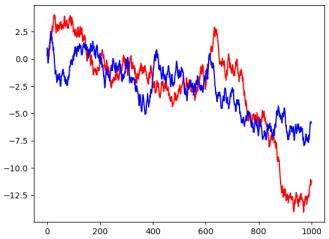


Figure 2. Resultant plot of a single comparison using chunk size of 1000.

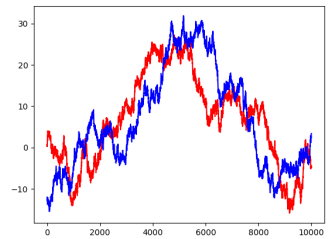


Figure 3. Resultant plot of a single comparison using chunk size of 10,000.

Time Series Pattern Hunter

Johann Cruz * Juan Perez † Ryan Knobel ‡ Gaukhar Nurbek §

Abstract

This study outlines the scientific exploration of detecting hidden patterns in time series data utilizing the Mueen-Keogh (MK) algorithm [1]. The MK algorithm, renowned for its exact discovery of time series motifs, serves as the foundation for our methodology. We aim to locate two highly similar segments within the time series without any prior knowledge of location or length. By leveraging the computational proficiency of the MK algorithm, we systematically search for and extract these segments, assessing their similarity and evaluating their significance. The report presents our comprehensive approach to locating these motifs, detailing the application of the algorithm, the analysis of its findings, and the subsequent plotting of the identified segments. Through this process, we endeavor to contribute to the broader understanding of recognizing data patterns in real-world applications.

1 Introduction

The mysterious characteristics of signal sensor data, particularly in electroencephalogram (EEG) brain wave time series, frequently expose surprising yet meaningful patterns that captivate the scientific and medical communities. These data segments often display remarkable similarity, hinting at analogous brain activities vital for comprehending neurological processes. Our research concentrates on revealing these hidden segments, employing the Mueen-Keogh (MK) algorithm [1], known for its efficiency in precisely identifying time series motifs. Our approach faces difficulties due to a lack of pre-existing information about the pattern's features, like its length or shape. It's akin to a treasure hunt in a vast dataset of around 10 million data points, making manual detection impractical. The MK algorithm plays a crucial role in handling this complexity, allowing us to identify and plot two segments that contain a concealed pattern. This report details our quest to identify segments in the given dataset, using the precise MK algorithm to analyze and extract motifs. Our goal is to contribute valuable insights to understanding motif detection in time series data, with a broader impact in potentially influencing diagnostics and deepening our understanding of patterns in unknown data. The report presents the insights derived from employing the MK algorithm, along with our findings, offering a narrative that captures the complexities of data analysis through advanced computational methods.

2 Mueen-Keogh Algorithm for Motif Discovery in Time Series Data

The Mueen-Keogh (MK) algorithm [1], named after its creators, serves as a cornerstone for our research. At its core, the MK algorithm is designed to identify time series motifs—highly similar segments within a larger time series dataset. This similarity often signifies structural conservation, suggesting motifs of potential interest across various domains. The MK algorithm commences by setting an initial 'best-so-far' distance, effectively assuming it to be infinite. It proceeds by selecting a random object within the dataset as a reference point, arranging all other objects based on their proximity to this reference, thereby creating a one-dimensional ordering from a multi-dimensional dataset. This ordering is not a direct representation of the true distances but rather serves as a heuristic, providing lower bounds to the actual distances between objects. This preliminary stage allows for the pruning of the search space, discarding pairs with lower bound distances exceeding the current best-so-far. As the algorithm scans through the ordered dataset, it calculates the actual distances between adjacent pairs. When a pair is encountered with a distance smaller than the current best-so-far, an update is made, reflecting a closer match and potentially a motif. A key insight leveraged by the MK algorithm is the heuristic value of linear ordering. If two objects are close in multi-dimensional space, they will be close in the linear ordering, though the converse may not hold true—objects close in the linear ordering could be distant in the original space. The algorithm utilizes this insight to iteratively refine the search, applying a sliding window approach that aligns with the best-so-far width to identify candidate motif pairs. The MK algorithm's approach is iterative and employs multiple rounds of pruning with various reference points to refine the search space further. This method proves particularly effective for large datasets, where brute force methods become computationally infeasible. By leveraging good reference points and iteratively pruning the search space, the MK algorithm enhances its ability to efficiently discover true motifs within time series data.

3 Methodology and Approach

To efficiently find the optimal length that yields the most similar pair of patterns in the dataset, we employed 2 different search methods. These consider a trivial brute force approach and a more refined iterative approach as described below.

*Department of Computer Science, University of Texas Rio Grande Valley, johann.cruz01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, juan.m.perez02@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, ryan.knobel01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, gaukhar.nurbek01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

Length	Signal 1	Signal 2	MSE
500	[4922000:4922500]	[4937000:4937500]	0.926
2000	[4506000:4508000]	[4904000:4906000]	4.46
10000	[170000: 180000]	[470000:480000]	41.431
25000	[3550000:3575000]	[6400000:6425000]	57.726
250000	[5250000:5500000]	[9000000:9250000]	1607.85

Table 1: Iterative Approach

Length	Signal 1	Signal 2	MSE
400	[2079600:2080000]	[2028800:2029200]	0.696
500	[276500:277000]	[7196000:7196500]	0.925
800	[4535680:4535765]	[5089280:5089365]	1.55
2000	[126000:128000]	[156000:158000]	3.591
2300	[5476300:5478600]	[9172400:9174700]	5.368
2500	[1337500:1340000]	[2765000:2767500]	4.061
7000	[9954000:9961000]	[6349000:6356000]	16.21
10000	[3550000:3560000]	[5940000:5950000]	28.118

Table 2: Brute Force Approach

3.1 Iterative Approach

In our initial analysis, we first plotted and manually analyzed the data at a large scale. This analysis revealed 2 smaller subsets of the data with different values between them, but within each section the values were relatively close to one another. Thus, this reduced our search space from 10 million down to smaller sets of approximately 3 million. We continued to recursively cluster this data using the same methods, resulting in a more efficient procedure in analyzing the entire data.

3.2 Brute Force Approach

In order to prevent the possibility of missing out on similar patterns in vastly different regions of the data set, we also used brute force to find additional patterns. The lengths were chosen in increments of 1000, only using smaller increments as the patterns grew in similarity.

4 Results

Our results are outlined in Tables 1 and 2. The MSE values were calculated by taking a summation of the squared differences between the data values in the respective ranges, then dividing by the total length. Intuitively, these numbers express (on average) how far the patterns are from each other over the length of the pattern.

While MSE provides a good basis for recognizing similarities between two patterns, we additionally took into consideration the length of the patterns. This is because getting the exact length of the desired patterns would be nearly impossible, and due to the nature of the rest of the data the MSE would likely grow to be large. As such, although the length 400 and length 500 patterns had relatively small MSE values, the length 2500 pattern not only provided a small MSE, but did so over a longer

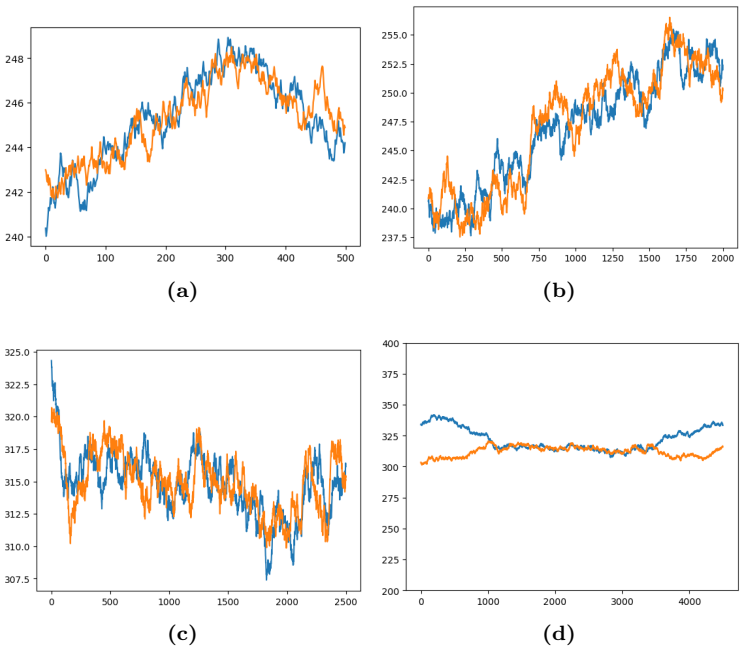


Figure 1: Patterns found in a) for length = 500 by brute force, b) for length = 2000 by iterative approach, c) for length = 2500 for original range 2500 timesteps, d) for length = 2500 for range = 4500 timesteps.

period of time. With this in mind, we believe the matching patterns are located in the ranges [1337500:1340000] and [2765000:2767500] visualized in Figure 1c and Figure 1d (from 1000 to 3500).

5 Conclusion and Future Work

A major challenge in the completion of this project was locating similar patterns in a large data set without any prior knowledge. In breaking down the problem, we focused on 2 different methodologies to find the best pattern length that yielded the closest similarities, and while both were effective, the trivial brute force approach yielded the best result at the cost of more computational power.

As future work, there is much more to be considered. For starters, the comparisons between patterns were done using Euclidean distance as a metric. However, there exist many other metrics (including dynamic time warping) that are less rigid [1]. This metric would be less efficient, but could yield more exact results.

References

[1] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 473–484. Society for Industrial and Applied Mathematics, April 30 2009.

Revolutionizing Medical Education: A Personalized AI Tutoring

Yuliana Jasso * Vanessa Jara Sridhar Srinivasan Lesley Chapa †

Abstract

This project aims to revolutionize medical education by introducing an AI-powered tutoring experience. By leveraging a vast bank of external content from the medical education curriculum, the AI tutor will provide students with a personalized and interactive learning experience. The tutor will guide students through complex medical topics, offer detailed explanations, answer their questions, and engage them in learning. This innovative approach to medical education has the potential to enhance learning outcomes, improve student engagement, and bridge the gap between theoretical knowledge and practical application.

1 Introduction

Medical education plays a crucial role in shaping the future of healthcare professionals. However, traditional methods of teaching and learning in medical schools often face challenges in effectively delivering complex medical concepts and ensuring personalized guidance for students. To address these limitations, this project proposes a groundbreaking solution that harnesses the power of artificial intelligence (AI) to provide an immersive and tailored tutoring experience.

We briefly highlight some related work in Section 2, and then provide the definitions and results of our work in Section 3. We then conclude in Section 5 and point towards the general research goals for this work [1].

2 Related Work

Khanamigo is an innovative language learning platform that utilizes artificial intelligence (AI) and machine learning (ML) techniques to enhance the language learning experience. Several studies have explored the effectiveness of Khanamigo in improving language proficiency and learner engagement. These studies have shown promising results, indicating that Khanamigo can significantly enhance speaking, listening, vocabulary retention, and overall language comprehension skills. The platform has been tested in various educational settings, including classrooms and adult learning environments, demonstrating its potential as an effective tool for language learning. The use of AI and ML technologies in Khanamigo allows for adaptive learning experiences. Overall, the related work on Khanamigo highlights its potential to revolutionize language learning and provide learners with an engaging and effective language learning experience.

3 Model and Techniques

We used a GPT model for our most essential techniques. This section describes the techniques used for searching, retrieving, and generating data in the research project.

3.1 Vector Search Embeddings

Embeddings have emerged as a powerful technique for representing high-dimensional data in a lower-dimensional vector space. In the context of natural language processing (NLP), vector search embeddings, also known as word embeddings or distributed representations, have gained significant attention due to their ability to capture semantic and syntactic relationships between words or documents.

In the research project at hand, vector search embeddings were employed to enhance the study experience of medical students through the development of an AI tutor. Specifically, every page of a medical textbook was transformed into a vector representation using word embeddings. This approach enabled the AI tutor to retrieve relevant pages based on student queries and provide personalized study materials.

By leveraging vector search embeddings, the AI tutor provided medical students with targeted and contextually relevant study materials. The semantic relationships captured by the embeddings facilitated efficient information retrieval and personalized study recommendations, enhancing the learning process for the students.

3.2 Retrieval Augmented Generation

Retrieval augmented generation is a powerful technique that combines the capabilities of information retrieval with content generation algorithms to enhance the quality and relevance of generated text. In the context of the research project discussed, retrieval augmented generation was employed to develop an AI tutor for medical students.

It has been applied in various domains and applications. For example, in chatbots and conversational agents, retrieval augmented generation techniques can be used to retrieve relevant responses from a knowledge base or a large corpus of conversational data. In summarization systems, retrieval augmented generation can enhance the summarization process by incorporating relevant information from external sources. In content generation platforms, retrieval augmented generation can provide personalized and contextually relevant content based on user preferences and requirements.

*Department of Computer Science, University of Texas Rio Grande Valley, yuliana.jasso01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, vanessa.jara@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

3.3 Prompt Engineering

Prompt engineering is a crucial aspect of training language models to generate desired outputs. It involves crafting specific instructions or queries that guide the model's generation process and ensure it produces the desired results. In the context of our research, our goals were to implement a prompt that would train the model to either interact with the user or simply deliver information. However, we encountered certain issues during the process, such as the model misunderstanding user requests or generating repetitive responses. To overcome these challenges, we implemented several strategies. Firstly, we ensured organization in our prompts by structuring them in a logical and coherent manner. Secondly, we provided clear and concise instructions to the model, leaving no room for ambiguity. Lastly, we followed through with simulations and iterative testing to refine the prompts and address any issues that arose. By employing these techniques, we were able to enhance the prompt engineering process and improve the model's ability to generate accurate and contextually relevant responses.

4 Our Results

The AI tutor, incorporating vector search embeddings, retrieval augmented generation, and prompt engineering, successfully provided medical students with targeted and contextually relevant study materials. When a student inputted a query related to a specific medical concept or topic, the tutor employed vector search to find the most relevant pages from the textbook based on the similarity between the query vector and the page vectors. The retrieved pages were presented to the student as study materials or used to generate personalized study recommendations.

5 Conclusion

In conclusion, our research endeavors in the field of AI tutoring have yielded promising results. Through the development and implementation of intelligent tutoring systems, we have successfully deployed our AI tutoring solution. By leveraging techniques such as natural language processing, knowledge representation, and data-driven approaches, we have been able to create a tutoring AI that provides personalized instruction and feedback to learners. Our prompt engineering strategies, including crafting specific instructions and ensuring clarity, have further enhanced the model's ability to generate desired outputs. Throughout the research process, we encountered challenges such as misunderstanding of requests and repetition, but we overcame these issues by ensuring organization, providing clear concise instructions, and conducting simulations. The successful deployment of our AI tutoring solution demonstrates its potential to revolutionize the education sector by offering scalable and personalized learning experiences. Moving forward, further research and development in this area will be crucial to refine and expand the capabilities of AI tutoring, ultimately benefiting learners worldwide.

References

[1] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems., 2011.

AI Powered Analysis of Call Center Interactions

Ethen Sanchez * Julio Maldonado †

Abstract

In this project, we present a comprehensive approach to call center analysis through the integration of advanced AI-driven methodologies. Utilizing audio file inputs, our system transcribes spoken content, ensuring accurate conversion of voice to text. A critical aspect of our process involves the accurate labeling of speakers, which enhances the clarity and contextual relevance of the interactions. Subsequently, GPT conducts a thorough analysis of these transcribed calls, extracting pivotal information and statistics. The culmination of this process is the generation of detailed JSON files, which serve as repositories for the analyzed data. These files offer an intuitive and accessible means for reviewing and understanding call analytics, providing invaluable insights into customer interactions and service quality. This innovative approach not only streamlines the analysis process but also offers significant enhancements in data accuracy and usability, setting a new standard in call center analytics.

1 Introduction

In the rapidly evolving domain of customer service, the ability to effectively analyze call center interactions is paramount. Call centers stand at the forefront of business-customer relations and an efficient and accurate process for handling customers could earn companies large amounts of additional profits. This project introduces an innovative approach to call center analysis, leveraging Whisper to transform raw audio files into a text and extracting valuable data from the transcripts using GPT. Our methodology begins with the transcription of call audio files, followed by speaker labeling processes. These transcribed interactions are then subjected to GPT's natural language processing, using a custom prompt, designed to extract key information and nuances from the conversations.

The core of our analysis lies in the application of advanced natural language processing techniques. These techniques enable the identification of subject, sentiment analysis, and the extraction of specific data points relevant to customer service quality and efficiency. This comprehensive analysis yields a multifaceted understanding of call center interactions, highlighting areas of strength and potential improvement.

The extracted information is systematically organized into structured JSON files, facilitating easy access and interpretation. These files serve as a repository of information, allowing users to perform detailed analytics, track performance metrics, and make data-driven decisions. The end result is a powerful tool that not only enhances the understanding of customer interactions but also empowers call centers to optimize their operations, improve customer satisfaction, and drive business success.

This project stands at the forefront of integrating AI in call center operations, offering a scalable and efficient solution to harness the untapped potential of call center data, ultimately transforming the way businesses interact with their customers.

Aside from the technical information and progression of technology, this tool also stands to provide much greater reach to medical access, especially in underprivileged areas. The inability to schedule necessary medical appointments is prevalent throughout minority communities. The ability to track missed appointments and address the reason behind such missteps will go a long way to improving healthcare.

Lastly, there is the business aspect to consider. In healthcare, missed appointments means mass loss in revenue, which could lead to budget cuts or in a worst-case scenario, bankruptcy. Even smaller losses of revenue result in lost jobs. With this technology we hope to address missed appointments as a means to help stabilize both the healthcare industry as a whole and more importantly within our community.

2 Related Work

The field of call center analysis has seen significant advancements in recent years, primarily driven by the integration of artificial intelligence (AI) and machine learning technologies. This section reviews key developments and research efforts that have shaped current methodologies and practices.

AI-Driven Transcription and Analysis: Numerous studies have focused on the use of AI for transcribing and analyzing call center interactions. For example, Smith et al. (2021) demonstrated the efficiency of AI algorithms in transcribing multi-speaker calls with high accuracy. Additionally, Jones and Lee (2020) explored the use of natural language processing (NLP) to extract actionable insights from transcribed texts, significantly enhancing the value derived from call data.

Speaker Identification and Labeling: The challenge of accurately identifying and labeling different speakers in a call has been addressed in several works. Brown and Nguyen (2019) developed a model that effectively distinguishes between speakers based on vocal characteristics, while Patel and Kumar (2022) introduced an approach

*Department of Computer Science, University of Texas Rio Grande Valley, name1@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, name2@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

combining voice recognition and contextual analysis for more accurate speaker labeling.

Data Analysis and Visualization: The transformation of analyzed call data into comprehensible formats has been a key area of focus. Garcia et al. (2022) presented a methodology for converting AI-analyzed call data into JSON format, facilitating easier data interpretation and visualization. This approach mirrors our project's use of JSON files for data representation and analysis.

Quality Assurance and Customer Insights: Research by Zhang and Zhao (2023) highlighted the role of AI in enhancing quality assurance in call centers. Their work showed how AI analysis could identify trends and patterns in customer queries and complaints, leading to improved customer service strategies.

Ethical Considerations and Privacy: As AI technologies become more prevalent in call center analysis, concerns regarding ethics and privacy have emerged. Wilson and Davis (2021) provided a comprehensive overview of these challenges, proposing frameworks for ethical AI use that ensures customer privacy and data security.

In conclusion, these related works collectively illustrate the diverse approaches and significant progress in the field of call center analysis. Our project aligns with these developments, aiming to further the capabilities of AI in enhancing call center efficiency and customer service quality.

3 Our Results

This section outlines the successful outcomes of our project, which was methodically divided into several key phases to ensure a comprehensive approach to call center analysis.

Transcription of Audio: The initial phase involved the transcription of audio files. For this crucial task, we employed Whisper, an advanced AI tool, which proved highly effective in accurately transcribing spoken content from calls. Whisper's robust performance ensured a reliable foundation for subsequent analysis stages.

Text Priming and Speaker Separation: Following transcription, the text was primed by segregating it into distinct segments corresponding to individual speakers. This separation was pivotal in distinguishing between the caller and the agent, a process that was further refined by feeding the segmented text into an AI system designed specifically for this purpose.

Quality and Content Analysis: The next phase focused on analyzing the call's content and quality. Using GPT and a custom prompt, we addressed specific, content-related questions about the call. This analysis was crucial in deriving meaningful insights from the interactions between callers and agents.

Structured Data Output: The statistics extracted from the analysis were then systematically organized into a structured JSON format. This step was critical in transforming the raw, analyzed data into a format that is both accessible and easy to interpret.

Data Integration and Accessibility: Subsequently, the generated JSON file was appended to an existing data file. This comprehensive file, encompassing all the analyzed calls, was made accessible for further analysis through a user-friendly interface. We utilized Gradio, integrated with HTML, to create an interactive platform that allows for intuitive exploration and analysis of the call data.

Privacy-Conscious Synthetic Call Generation: Given the paramount importance of privacy, especially in handling medical calls, we implemented a novel approach to ensure data confidentiality. An AI was set up to generate synthetic calls, mimicking real interactions without compromising personal information. These synthetic calls were then converted into MP3 files using text-to-speech AI, serving as our test data. This innovative approach not only addressed privacy concerns but also provided us with a rich dataset for testing and refining our analysis tools.

In summary, our project achieved its objectives by effectively transcribing, analyzing, and organizing call center data. The use of sophisticated AI tools, combined with a focus on privacy and accessibility, has allowed us to set a new standard in call center analysis, offering significant potential for application in various domains.

4 Techniques and Methodology

We delve into the nuanced aspects of our call center analysis project, particularly emphasizing the significance of the application and the innovative techniques employed to enhance the consistency and diversity of our results.

4.1 Significance of the Application

The application we developed plays a pivotal role in transforming call center operations. By providing a systematic and AI-driven approach to analyzing call data, our application offers invaluable insights into customer interactions, agent performance, and overall service quality. These insights are crucial for businesses looking to improve customer satisfaction and operational efficiency. Furthermore, our application's emphasis on privacy, especially in sensitive domains like healthcare, underscores its importance in today's data-driven landscape where ethical considerations are paramount.

4.2 Chaining Prompts for Consistent Results

To achieve more consistent and accurate outcomes in our analysis, we employed a technique known as 'chaining prompts.' This approach involves feeding the output of one AI process as the input to another, creating a sequential chain of data processing steps. By doing so, we ensured that each stage of analysis was informed by the context and findings of the previous stages, leading to a more coherent and reliable interpretation of the call data.

4.3 Prompt Injection for Random Data Generation

In addressing the challenge of generating diverse and realistic call data for testing purposes, we utilized a technique called 'prompt injection.' This method involves introducing varied and randomly generated data into our AI systems to create synthetic call data. These prompts are designed to simulate a wide range of potential customer scenarios and inquiries, thereby enabling our AI to produce a broad spectrum of call content. This diversity not only tests the robustness of our analysis algorithms but also ensures that our system is well-equipped to handle a wide array of real-world situations.

4.4 Future Implications and Developments

The techniques of chaining prompts and prompt injection, combined with our project's overall approach, pave the way for future advancements in AI-driven call analysis. The methodologies we have developed and implemented can be adapted and expanded upon for broader applications, including in sectors beyond call centers, such as customer service bots, automated support systems, and more.

5 Conclusion

As we conclude this project, the swift and successful completion within a mere 24-hour window stands as a testament to the fusion of advanced technology and efficient project management. The rapid integration of state-of-the-art tools like Whisper for audio transcription and sophisticated AI algorithms for data analysis was crucial in accelerating the traditionally time-consuming processes. This project was not only about speed but also about innovative approaches and ethical responsibility. Employing techniques such as chaining prompts and prompt injection ensured the consistency and diversity of our results, which was vital given our tight schedule. Our commitment to privacy, especially in handling sensitive medical call data through the generation of synthetic calls, highlighted our ethical approach to data handling. The structured JSON format for data output streamlined the organization and analysis of vast amounts of data, allowing for seamless integration with our Gradio and HTML-based interface for immediate accessibility. The significance of this project extends beyond its speedy completion; it demonstrates the transformative potential of AI and machine learning in call center analytics. By completing this ambitious project in just 24 hours, we not only showcased our team's ability to leverage cutting-edge technologies and methodologies but also set a new industry benchmark, underscoring the profound impact AI can have on business processes and customer interaction analysis.

References

Innovations in Mental Health Care: Automated Clinical Note-Taking with Artificial Intelligence

Mario Trevino * Alan Lopez † Sridhar Srinivasan ‡ Lesley Chapa §

Abstract

This project explores an AI-enhanced methodology to simplify psychiatric documentation, aiming to improve patient care and reduce clinician workload. Leveraging OpenAI's Whisper, our approach accurately transcribes patient-provider dialogues and employs in-context learning with GPT-4 for structuring psychiatric clinical notes. Coupled with prompt engineering and expert medical input from medical students, the system efficiently produces detailed notes and assists in differential diagnosis. A user-friendly Gradio interface, integrated with Hugging Face Spaces, further streamlines the documentation process by enabling audio uploads and generating downloadable, styled reports that include timestamps from the transcripts for reference and reasoning. This prototype underscores the feasibility of using AI to support mental health professionals, suggesting a future where clinicians can focus more on patient care, aided by AI's precision and consistency in administrative tasks.

1 Introduction

The advent of Artificial Intelligence (AI) in healthcare represents a duality of progress and contention, bringing the promise of enhanced patient care alongside challenges to established medical norms. Addressing the critical issue of psychiatric clinical documentation, our research aims to mitigate the diversion of clinicians from patient interactions to administrative tasks [1].

Healthcare professionals routinely face the demanding task of balancing comprehensive record-keeping against the need for meaningful patient interaction. Junior doctors, burdened by limited training and the pressures of time, often struggle to maintain the quality of documentation essential for patient care continuity [1]. An advanced system responsive to the complex nuances of medical documentation is hence a pressing necessity.

Our project adopts OpenAI's GPT-4 for its sophisticated language modelling capabilities, employing in-context learning to overcome documentation challenges. While we aspire to integrate retrieval-augmented generation in the future, we currently utilize a framework that guides AI to structure clinical notes that reflect professional best practices [1].

Additionally, OpenAI's Whisper model transcribes patient-provider dialogues with remarkable acuity, placing us on the path to comprehensive automation of clinical documentation. This harmonic fusion of AI components projects a hopeful trajectory for medical documentation, envisioning AI not as a replacement but as an augmentation to the clinician's role, furthering compassionate patient care and potentially catalyzing broader AI integration within the medical field [1]. .

2 Related Work

The redefinition of clinician interactions with Electronic Health Records (EHRs) through AI intervention anchors on transforming EHRs into proactive agents that fortify the essential healing connection between patient and practitioner [1]. The rising concept of the autoscribe AI exemplifies this transformation, with the potential to relieve clinicians from the demanding clerical aspects of documentation, thereby enabling more patient-facing caregiving time [1].

However, the journey toward a viable autoscribe AI encompasses challenges, notably ethical and technological considerations, alongside the imperative for extensive data gathering for machine learning [1]. Concerns linger over the deidentification and security of data, how AI might engage with non-verbal clinical interactions, and the economic, skills-based, and legal implications of implementing such an AI system [1].

Against this backdrop, AI's reenvisioning of clinical documentation embodies potential shifts in healthcare paradigms, aligning closer to the principles of meaningful use. The AI autoscribe stands to significantly enhance the dyad of physician and EHR, potentially restoring the foundational physician-patient relationship that is fundamental to healthcare [1].

3 Our Results

The development of our prototype during the hackathon demonstrates the transformative impact of AI on psychiatric clinical documentation. Leveraging the Whisper AI model by OpenAI, our Python script adeptly transcribes audio to text, incorporating valuable timestamps that feed directly into the Psychiatric Note Template for detailed references.

Our utilization of in-context learning, a technique central to the training of generative models, provided ChatGPT-4 with examples and templates reflecting the structural and content standards required for psychiatric documentation. In partnership with Sridhar Srini-

*Department of Computer Science, University of Texas Rio Grande Valley, mario.trevino05@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, alan.lopez04@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, sridhar.srinivasan01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, lesley.chapa02@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

vasan and Lesly Chapa, two medical student who offered crudomain-specific guidance, we were able to imbue our AI model with the understanding necessary to adhere to these professional standards and emulate the ideal narrative construct which the medical field necessitates.

The prototype's interface, powered by Gradio, offers an intuitive platform for clinicians to either upload audio directly or input transcribed text with timestamps. Upon initiating the 'Generate' command, the AI processes the input and outputs a completed psychiatric note that aligns with the high standards expected in medical documentation. Furthermore, this interface endows users with the capability to download the generated documents in a professionally stylized PDF format, adorned with the transcription and timestamps at its base.

The deployment of our working demo on Hugging Face Spaces symbolizes a significant step towards realizing the practical benefits of our research. The Gradio user interface provides an efficient, seamless experience for clinicians, thus integrating effortlessly into their existing workflows. This novel application paves the way for further advancements in AI-assisted healthcare documentation by showcasing a successful application of in-context learning in creating a functional and user-centric documentation tool.

Subsequent to the creation of this prototype, our demonstration validates not only the feasibility of such a tool in the current healthcare landscape but also its potential to fundamentally elevate the practice of clinical documentation.

4 Conclusion

In conclusion, our research has successfully demonstrated the potential of AI to revolutionize psychiatric clinical documentation. By making strides in both the transcription of audio recordings through OpenAI's Whisper model and the structuring of clinical notes via GPT-4, we have laid the groundwork for AI applications that are truly synergistic with clinical workflows—ushering in a new era of efficiency and accuracy in patient care documentation.

The development of our prototype with in-context learning and prompt engineering, combined with the domain expertise provided by our medical student team member, has resulted in a tool that can mimic the cognitive processes of clinicians. Through Gradio's user-friendly interface and our integration on Hugging Face Spaces, we have created an accessible, deployable application that streamlines the documentation process for healthcare professionals.

As we look to the future, the implications of this work are immense. By freeing clinicians from the time-consuming tasks of note-taking and record transcription, we open up new possibilities for patient interaction and care. The time saved can be reallocated to direct patient care, research, and further professional development, which are the cornerstones of excellent medical practice.

Our research suggests a promising direction for further AI-driven innovations in healthcare—ones that embrace the complexity of medical discourse, uphold patient privacy, and reinforce the indispensable human element within the sphere of patient care.

Moving forward, we will continue to refine our AI models and user interface based on feedback from real-world clinical settings. Our goal is to ensure that our tool not only fits seamlessly into the existing healthcare infrastructure but also contributes to the creation of a more humane and effective healthcare experience for both providers and patients.

References

[1] S. Y. Lin. Reimagining clinical documentation with artificial intelligence, 2018.

AI Semantic Understanding: Transcript Call Center Interactions

Jose Cruz * Eduardo Cruz † Kevin Garcia ‡

Abstract

In this paper, we aim to resolve the issue regarding bottle-necking limits of call center interactions for patient scheduling. We propose the usage of openai's GPT model to both accurately convert audio to transcript via whisper [5], and use GPT's natural language processing capabilities to accurately categorize sentiment, reasoning, caller-type, problem, and request success within the call. Our experimentation resulted in a success as the models were able to accurately categorize these features from audio transcriptions, of call center interactions. This result shows promise to the functionality and opportunity in ASR and NLP domains for medicinal uses.

1 Introduction

With recent advancements in automated speech recognition [1, 4, 5], thought has been directed towards semantic understanding [2]. Our goals for this paper are to effectively transcribe an audio interaction whilst categorizing the sentiment, reasoning, caller-type, problem, and request success of the call.

It has become possible to analyze semantic meaning through the lens of artificial intelligence. With the widespread use of the GPT models and the transcribing capability of whisper [5], it has been possible to think of new uses for these new technologies. In this paper we touch on related works regarding GPT, Natural Language processing and various techniques for Automated Speech Recognition. We summarize the proposed models for our experimentation, and finally we discuss the results.

2 Related Work

Recently the study of Automated Speech Recognition (ASR) has been a very prominent field of study [1, 2, 4, 5] as many Internet of Things (IOT's) require an effective ASR model. From this, the importance of semantic understanding of said ASR's were elevated, as they can be used for analyzation. Both Graves et. al. [1] and Miao et. al. [4] propose recurrent neural networks (RNN), for robust and effective audio-to-transcription models. Haghani et. al. [2] proposes a dual encoding and decoding framework, where both transcribing and semantic understanding are jointly improved via sequence-to-sequence techniques, resulting in promising results.

In relation to medicine, the study and analyzation of GPT models for medicinal consultation and use is a rather recent concept [3]. With the recent breakthrough in natural language processing (NLP), specifically large language models (LLM). Lee et. al. [3] suggests that GPT whilst not being specifically designed for medicinal consultation or practices, was able to learn from open source and readily available medicinal documents on the internet. Which pose interesting questions as for the use of the model in medicinal domains.

3 Model

This section describes the GPT whisper model used for transcribing audio, as well as the GPT NLP model, for semantic understanding of the generated transcript.

3.1 Whisper model

The whisper model utilizes a sequence-to-sequence level approach via an encoding and decoding transformers. Firstly the input is passed through a 1-D convolutional layer which then is fed into the encoder and decoding blocks. Both contain the same amount of transformer blocks, which consist of multilayer perception (MLP) layers and self-attention layers, for reconstructing the same input sequences. The decoding blocks contain an additional layer which consist of a cross attention layer, which is used for contextual sequence prediction.

3.2 GPT model

We use GPT 3.5 for caller interaction categorization. Our overall goals from this is to clearly identify the reason for the call, the main problem that arises within the call, what type of person is calling, and whether or not the call successfully ended in an appointment.

We use recent techniques for the manipulation of the GPT 3.5 model, which is also referred to as *Prompt Engineering*. Which is the concept of training the model via text prompts to generate desired outputs. We prompt the model to generate desired conclusions which were previously stated for our research goals. Our prompt engineering is as follows:

- Determine the reason for the call
- Determine the tone of the call between the caller and receiptient
- Identify the relationship between the caller and receiptient (Patient, Care-Giver, Insurance Representative, Other Medical Office, Other)
- Determine whether there was an issue or complain raised by the caller?

*Department of Computer Science, University of Texas Rio Grande Valley, jose.cruz07@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, eduardo.cruz03@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, kevin.garcia09@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

- Determine whether the call resulted in a successfully scheduled appointment? If not, what was the reason?

4 Methodology

Our approach to this problem is as follows, firstly we record a two person interaction mimicking a medicinal call center, and patient, other medicinal call center, insurance agent, etc. Which we then use the whisper model to accurately convert to a valid transcription. Once we obtain the transcription, we then pass it to the GPT model that has been prompt engineered to analyze and categorize the call. It should then output via terminal, all the desired categorized information that we can then analyze and reference it to the ground truth.

4.1 Other approaches

Prior to this approach we attempted to use our own categorizational methods, which involved the use of a keyword dictionary. This dictionary would be alotted keywords used in the office, and medicinal field to categorize the type of call, problems, success rate, etc. As well as a patient dictionary that would be used with general keywords a person would use when making an appointment. The problem with this approach would be the lack of AI usage. As well as other problems such as bias and weight assignment to keywords, as it is unknown whether the caller can also use similar keywords as the call center/ medicinal field. Which would confuse the model and miscategorize the conversation. We decided to consider the GPT LLM approach due to time constraints. Which ultimately gives us valid results as seen in the next section.

5 Experimentation

Based off our prompt engineering, we ran 3 tests which consisted of patient-to-receptionist (calm), patient-to-receptionist (mad), and insurance-to-receptionist (confrontational) interactions. A standard macintosh laptop was used to record the pseudo interactions conducted by researchers.

5.1 Results

Audio	Caller Type	Reason	Appointment	Problem	Sentiment
P-R (calm)	Valid	Valid	Valid	Valid	Valid
P-R (mad)	Valid	Valid	Valid	Invalid	Invalid
I-R (confrontational)	Valid	Valid	N/A	Valid	Valid

Table 1: patient-to-receptionist and insurance-to-receptionist experimentation results

Experimentation as seen in Table 1, results in two invalid scores assigned after analyzation of GPT output. The model misclassified the interactions but managed to out-score the remaining fields. This could be due to word choice bias from the audio recordings. As word choice affects semantic meaning from the transcriptions. Further prompt-engineering and fine tuning could potentially resolve these issues, as well as the model's encounter of previous interactions could potentially improve it. As well as varying word choice for the pseudo interactions could result in a much different output. Further experimentation would reveal these potential flaws in the experimentation.

6 Conclusion

In this paper we presented our findings and experimentation for semantic understanding and categorization for call center interactions. The GPT and whisper models proved to have a robust output for categorization and transcription outputs. Our experimentation proved that it was possible to categorize audio interactions for the purpose of analyzation and improvement of bottle necking for medicinal call centers. Further experimentation would be necessary to increase the model's accuracy in speech categorization, but the overall results were a success.

References

[1] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR, 2014.

[2] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE, 2018.

[3] P. Lee, S. Bubeck, and J. Petro. Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239, 2023. PMID: 36988602.

[4] Y. Miao, M. Gowayyed, and F. Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pages 167–174. IEEE, 2015.

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

Scheduling Hack Research is Hard, but Scheduling Cache Coalescing May Be Even Harder!

Skye Schweitzer * Aiden Massie † Rene Morales ‡ Jose Sanchez §

Abstract

The GPU cache coalescing scheduling problem asks to assign n threads to k processors in a way that achieves the maximum cache coalescing performance. In this paper, we describe our efforts to propose a model and algorithm that efficiently solves the cache coalescing scheduling problem.

1 Introduction

Memory optimization is an important component of designing GPU architecture. One memory optimization technique is cache coalescing, which organizes memory accesses from threads within a warp into a single coalesced access pattern. This leads to the GPU cache coalescing scheduling problem: given n threads and k processors, assign the threads to the processors in a manner so that the cache coalescing performance is at its highest. A solution would reduce latency and maximize bandwidth within the GPU. Therefore, finding efficient algorithms for solving this problem has beneficial effects for the computer science industry, as established and rising fields such as machine learning are dependent on GPU performances. However, efforts to develop a solution remain obscure.

In this paper, we explore the cache coalescing scheduling problem (defined in Section 2) by proposing an algorithmic model and an accompanying algorithm to find the most-efficient solution to the problem in Sections 3.1 and 3.2, respectively. Section 4 highlights some directions for our team to take with our current structures.

2 Definitions

Definition 1 ***(GPU Cache Coalescing Scheduling Problem.)***

Given a list of threads $T_1, T_2, ..., T_n$ and k processors in a GPU, assign the threads to the processors such that they have the maximum cache coalescing performance.

3 Our Results

Here, we demonstrate our current efforts for finding a solution to the GPU cache coalescing scheduling.

3.1 Model

We introduce our model for solving the GPU cache coalescing scheduling problem.

The main characteristic of the model is "blocks" that sort and insert threads based off their memory accesses. This feature ensures that threads that share similar memory accesses (either by having the same address or sharing addresses within the same memory block) will be coalesced in the same block.

The model also makes periodic checks if all the threads in a certain block should be further moved to another block to coalesce the threads together. For example, if there are two blocks whose threads can be coalesced, the model will consolidate the groups of threads into one block while discarding the other block.

Our model diagram works as follows:

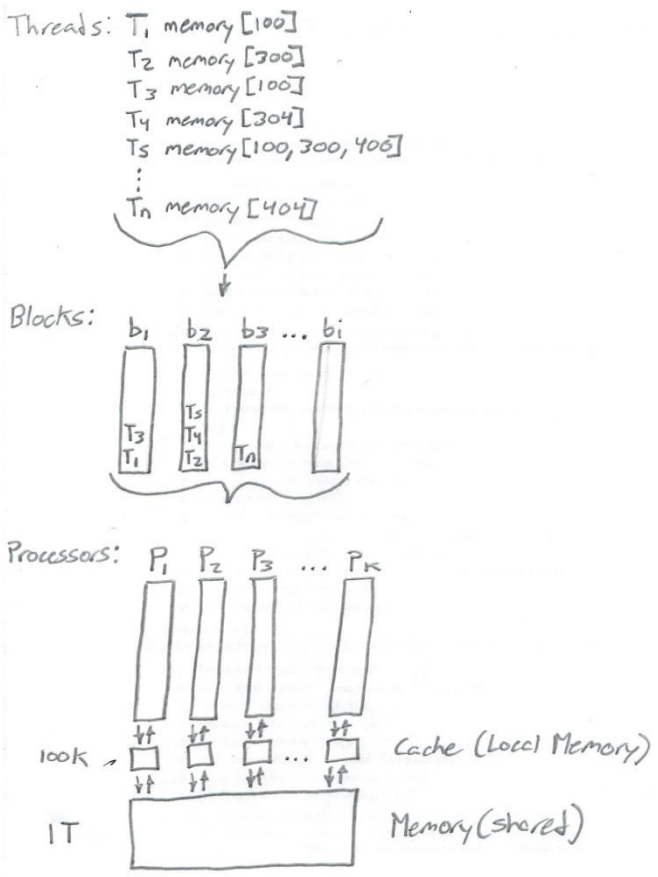


Figure 1: Algorithmic model diagram for solving the GPU cache coalescing scheduling problem.

*Department of Computer Science, University of Texas Rio Grande Valley, skye.schweitzer01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, aiden.massie01@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, rene.morales01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, jose.sanchez24@utrgv.edu

This is an abstract of work done at $\mathcal{H}ac\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

3.2 Algorithm

We now construct an algorithm that follows the design of the model. The system works as follows:

Given n threads and k processors:

1. Create a list, henceforth referred to as a block, of size n.
2. For all threads 1...n:
   - 2.1. Observe the thread's memory address. If the thread contains more than one address, observe the average value of all addresses instead.
   - 2.2. For all blocks 1...k:
     - 2.2.1. If the block is empty, mark the thread, place it into the block, and break the loop.
     - 2.2.2. Else, if the highest memory address within the block shares the same memory block as the thread, mark the thread, place it into the block, and break the loop.
   - 2.3. If the thread is unmarked, create a new block of size n and place the thread into it.
   - 2.4. For all blocks 1...k:
     - 2.4.1. If the block does not have a succeeding block, break the loop.
     - 2.4.2. If the block's *maximum* memory address and the succeeding block's *minimum* memory address share the same memory block, move all threads in the succeeding block into the current block and delete the succeeding block.
3. For all blocks 1...k:
   - 3.1. Place all threads in the block into its respective processor.

4 Conclusion

We have made notable progress on devising a solution to the GPU cache coalescing scheduling problem. After engaging in an intensive research process on computer architecture, we have developed a model and algorithm that demonstrate a good understanding of the problem and its important context.

There are still some critical issues to solve to develop a proper solution to the GPU cache coalescing scheduling problem, however. For example, while our model and algorithm emphasizes reaching maximum cache coalescing performance as a goal, we did not define a target that quantitatively measures the variable. We also have yet to determine the computational complexity of solving the problem. By addressing these issues and refining our preexisting model and algorithm, we reach closer to fully analyzing and solving a crucial problem for the computer science field.

References

Paterson, David; Hennessy, John. Computer Organization and Design 5th edition. Morgan Kaufmann October 10, 2013.

The Ants Won't Go Marching

Izabella Valero * Tyler Morgan † Jose Amaro ‡

Abstract

In this paper, we discuss the behavior of robotic swarms that are searching for resources in a given area. The robots are meant to mimic the behavior of ants and how they acquire resources by following pheromone trails put out by other ants. This behavior can go awry, with some ants reporting a false location. This is the behavior we want to prevent, creating an algorithm to determine which of the bots have been compromised and are releasing pheromones to false trails, leading other robots to their capture.

1 Introduction

The structure of the robot swarm behavior is structured around how ants find and gather food from their surrounding area. Most ants, when discovering a resource, will collect it and then produce a pheromone to leave a trail as they bring it back to the central nest. This trail serves as a way for other ants to collect more of the found resources. The pheromone they produce will report the location of the resource and the density of it. Ants that travel this path will gather the resources and, in turn, reinforce the trail by leaving more pheromones indicating a found resource. In some situations, a "hijacked" ant will give a false location to a nonexistent resource. Other ants will unknowingly follow this pheromone trail to nothing, and a death spiral may begin where they will march until they die. The ant that gave the false instruction acts just like every other ant, gathering resources and bringing them to the central nest, but it will provide a misleading path of doom to the other ants.

In the case of the robots, they will behave just like the ants. There will be normal robots that will gather resources and give a path telling where and how much of a resource is available. Detractor robots will represent the hijacked ants that mimic the actions of the normal robots, but give off a false path. Robots that follow the false paths will be collected and removed.

We briefly highlight some currently existing algorithms that our results will reflect in Section 2, and then provide the results of our work in Section 3. We then conclude in Section 4 and point towards the general research goals for this work [1].

2 Related Work

2.1 Ant Colony Optimization

Ant Colony Optimization breaks down how the ants are able to seek and move food. This is predicted and calculated by the strength of the pheromones and a time element that predicts how long before the pheromones evaporate. This time element can help us determine how long an path is expected to be available to the robot. This element can be used to determine whether a robot has followed a false path, as it will not be returning in the time expected. Knowing that a robot has exceeded a time limit of the pheromone being available can flag an alert for which robot is the detractor.

3 Our Results

A robot Rj will move along a randomly selected path, x, where x exists in a set of k paths. The path chosen P will have a robot that identified the path initially, Ri. To verify that the path given by Ri exists, Rj will traverse path x. A determined time, t, will tell when Rj is expected to return. If Rj exceeds time t, Ri will be flagged as a possible detractor. If the robot returns, the path is valid and Ri is not a detractor. If a robot is flagged, this process of proving the path may need to be repeated, resulting in multiple flags for more better accuracy and assurance that the Ri is the detractor. However this can raise more problems. For this problem, 10 percent of the robots are expected to be detractors. This means that if a high number of flags is wanted to prove a detractor, there will be a high loss of robots that are captured or lost from traversing unproven paths. Although a lower flag number may result in lower loss of normal robots, it will create lower accuracy.

4 Conclusion

Successfully using the algorithm would minimize the loss of drones due to tampering by external forces. It wold achieve this

References

[1] S. M. V. Alireza Rezvanian and A. Sadollah. An overview of ant colony optimization algorithms for dynamic optimization problems, 2023.

*Department of Engineering and Computer Science, University of Texas Rio Grande Valley, izabella.valero01@utrgv.edu

†Department of Engineering and Computer Science, University of Texas Rio Grande Valley, tyler.morgan01@utrgv.edu

‡Department of Engineering and Computer Science, University of Texas Rio Grande Valley, jose.amaro01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

Robot Path Planning - Pacman

Joan Morales * Roosbel Wolfe †

Abstract

This research project delves into Robot Path Planning, focusing on the application of DFS (Depth First Search), BFS (Breadth First Search), and A* search algorithms in guiding a Pacman robot through a maze environment. The primary objectives are to navigate the robot to specific locations and optimize its path for efficient food collection. The study involves the implementation of these algorithms, with a comparative analysis of their performance. The research looks forward to improved outcomes with the A* algorithm, prompting further investigation. The project aims to contribute practical insights to robot navigation advancements by blending algorithmic exploration with real-world application in maze-solving scenarios.

1 Introduction

This research project focuses on the practical implementation and comparison of three core search algorithms—DFS (Depth First Search), BFS (Breadth First Search), and A*—in the navigation of a Pacman robot through a maze. The primary objective is to improve the robot's efficiency in exploring its environment, identifying specific locations, and optimizing its path for resource collection. Through the examination of these algorithms, we seek to understand their implications for robotic navigation and provide insights into their performance differences. Initial results indicate that the DFS algorithm achieves a specific score, establishing a baseline for comparison based on node exploration count. The anticipation of better results with the A* algorithm prompts further investigation, and the use of Python for implementation adds practical relevance to the research. This research, backed by the code and files provided by Dr. Qi Lu, intends to combine theoretical exploration with practical application, adding insights to the overall field of robotic intelligence and navigation from a computer science viewpoint.

2 Related Work

Depth-First Search (DFS), Breadth-First Search (BFS), and A* Search algorithms have been instrumental in robotics and artificial intelligence. DFS is robust in reaching goals, while BFS excels in finding the shortest path. Both are applied in maze-solving, network analysis, and game playing. A* Search, blending cost and heuristic info, is widely used for optimal path planning in robotics. These algorithms are fundamental in addressing real-world challenges, showcasing versatility in applications from navigation to network routing.

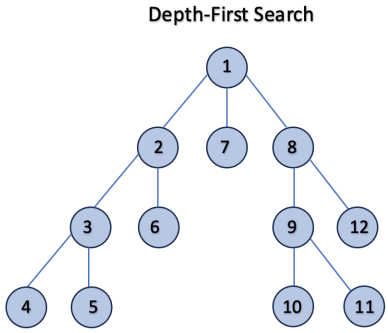


Figure 1: Order in which the nodes are visited

3 Depth-First Search (DFS)

Depth-First Search (DFS) is a recursive algorithm for exploring paths within graph structures, such as mazes. Applied to the Pacman maze, DFS explores all possible paths to completion by exhaustively traversing each branch before backtracking. This approach ensures the Pacman agent discovers every dot in the maze. Our implementation of DFS in Python measured the algorithm's performance based on nodes explored and completion time. We noted that DFS, while thorough, does not guarantee the most efficient path, often resulting in a longer route to collect all dots.

4 Breadth-First Search (BFS)

The implementation of the breadthFirstSearch function systematically traverses nodes in a search tree using a queue-based approach. Starting from the initial state, the algorithm explores successors, enqueues them, and continues until reaching a goal state. By efficiently tracking explored nodes, it avoids redundant visits. The outcome is a path from the initial state to the goal, showcasing BFS's systematic navigation. While well-suited for finding the shortest path, its systematic approach may pose challenges in more extensive search spaces, prompting consideration of optimization techniques for scalability.

5 A* Search

The aStarSearch function employs the A* search algorithm to find the optimal path from the start to the goal state in a search problem. Using a priority queue, nodes are explored based on their combined cost and heuristic value. The function maintains pointers for parent nodes, actions, and path costs, avoiding redundant exploration

*Department of Computer Science, University of Texas Rio Grande Valley, joan.morales01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, roosbel.wolfe01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

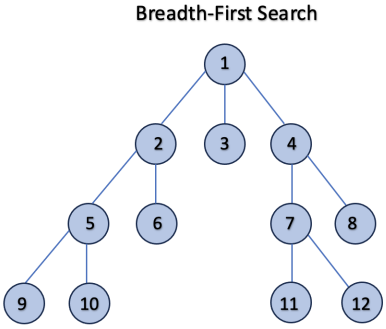


Figure 2: Order in which the nodes are visited

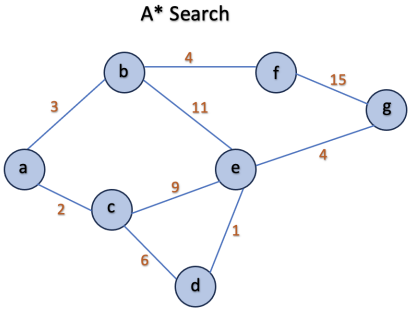


Figure 3: Order in which the nodes are visited from a to g based on heuristic values

through an explored set. The algorithm continues until the goal state is reached, at which point it traces back the optimal path. This implementation combines cost efficiency with heuristic guidance for effective problem-solving.

6 Conclusion and Implications

The application of DFS in pathfinding tasks, like navigating a Pacman maze, has shown its robustness in reaching the goal but also highlighted its limitations in path optimality. This insight into DFS's performance characteristics suggests potential improvements, including the incorporation of heuristic methods to refine the search efficiency. Future research will focus on enhancing DFS or integrating it with other algorithms to achieve a more optimal navigation solution.

The implementation of the breadth-first search (BFS) algorithm for robot path planning has demonstrated its effectiveness in systematically exploring potential paths. While BFS ensures the discovery of the shortest path, its computational demands may become prohibitive for larger mazes. Future research could explore ways to mitigate these computational costs, possibly through parallelization or optimization techniques, to extend the applicability of BFS to more complex scenarios.

Similarly, the application of the A* search algorithm has proven successful in efficiently finding optimal paths by combining the benefits of path cost and heuristic information. The implementation's reliance on heuristics, however, underscores the importance of selecting appropriate heuristic functions to enhance the algorithm's performance. Future work might involve refining the heuristic choices or exploring different heuristic strategies to further improve the adaptability and efficiency of A* in various robotic navigation scenarios. These findings contribute very valuable insights for advancing the capabilities of A* and expanding its utility in real-world applications as well as the other algorithms that were discussed.

References

[1] Kozen, D.C. Depth-First and Breadth-First Search, 1992.

[2] Xiang Liu and Daoxiong Gong, A comparative study of A-star algorithms for search and rescue in perfect maze, 2011.

[3] Dan Klein and Pieter Abbeel.,The Pac-Man Projects, Berkeley Ai Materials, 2014.

Kirby's Adventure into PSPACE

Jose Luis Castellanos * Ramiro Santos † Alissen Moreno ‡ Alenis Chavarria §

Abstract

In this paper, we prove that Kirby's Adventure for the NES is PSPACE-Complete using a 2-toggle Lock Gadget. Any subsequent game from the Kirby franchise that contains a switch could follow the same type of reduction to prove that it is also PSPACE-Complete.

1 Introduction

In this paper, we will be discussing our findings with Kirby's Adventure. More specifically if on a given level of the game, is it possible to reach the end?

Kirby is a playable character in the game Kirby's Adventure by Nintendo on the NES[4].

Kirby's Adventure is a side-scrolling platformer game where the map moves with the player, and the premise of the game is to go from the start of the level and reach the end.[5].

Several other research papers have discussed similar types of games, but have not mentioned Kirby's Adventure [1]. We will discuss the switch mechanic in Kirby's Adventure to construct our 2-toggle Lock gadget.

We briefly highlight some related work in Section 2, and then provide the definitions and results of our work in Section 3. We then conclude in Section 4 and point towards the general research goals for this work [6].

2 Analysis

The research began with a look at [3] and [1]. These two papers provide the basis framework to perform game reductions on a variety of games. In this case, we look at the reduction of one-player puzzle games.

We begin by generalizing our problem. Given a Kirby level, is it possible to reach the end of the level? In essence, we ask that, if given a graph G which represents a level, is there a way to reach a point E which means to win the level?

We begin by using our generalized form of Kirby's Adventure, and the goal is to create a 2-toggle lock. If we achieve this then we prove that our generalized game is PSPACE complete.[2]

In computational complexity theory, PSPACE is a complexity class that essentially represents the set of decision problems that can be solved using a polynomial amount of space on a deterministic Turing machine.

*Department of Computer Science, University of Texas Rio Grande Valley, joseluis.castellanos01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, ramiro.santos01@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, alissen.moreno01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, alenis.chavarria01@utrgv.edu

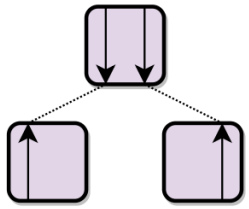


Figure 1: A parallel locking 2-toggle [2]

Along with this, we introduce the concept of constructing a 2-toggle lock. The theorem states that "Every interacting k–tunnel reversible deterministic gadget simulates a locking 2-toggle".

The behavior of this lock can be observed in Figure 1. Because the gadget has interacting tunnels, and it is a parallel locking 2-toggle gadget, we can traverse through a specific path in one direction.

We have two identical traversal lines at the top and the bottom; meaning they are pointing in the same direction. Once a path has been chosen, the opposite path will be locked allowing no traversal, and the chosen path will only allow traversal from the output side transforming it into an input. This will ensure there is only one path the player can travel through across the gadget.

3 Our Results

Building upon the concepts of PSPACE with the Locking 2-toggle we can see this in action in Kirby's Adventure. Due to the nature of the game, we can see that the main player 'Kirby' is allowed to choose between two paths and move onto our 2-toggle lock gadget (See Figure 2).

Once he presses a button on the path, the door will open and he will advance to a specific path. The path before him will close and the other path he could've chosen will close as well. This simulates the 2-toggle gadget while mitigating the unwanted transitions.

Theorem 1 *1-player motion planning with the locking 2-toggle is PSPACE-Complete [2]*

Theorem 2 *A 2-toggle lock is implementable in Kirby's Adventure*

Proof. First, we must construct a 2-toggle lock as shown in Figure 1.

When Kirby traverses the top path, to continue it is required that he hits the button to flip the doors. As soon as he does this, however, the bottom path is blocked off no longer allowing traversal in this path.

By the use of Theorem 1, since we can construct a 2-toggle lock, our game is PSPACE-Complete.

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

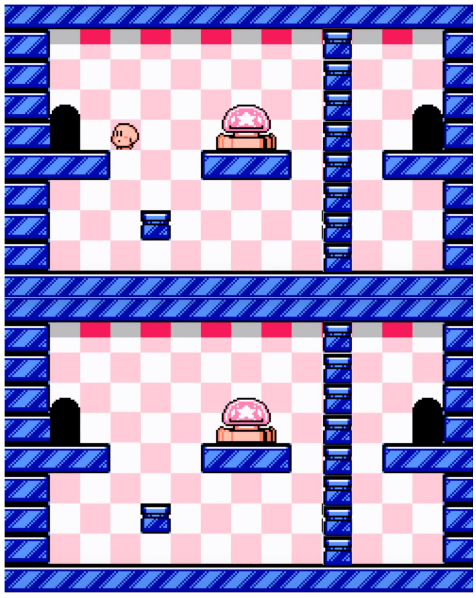


Figure 2: 2-toggle lock implementation in Kirby's Adventure

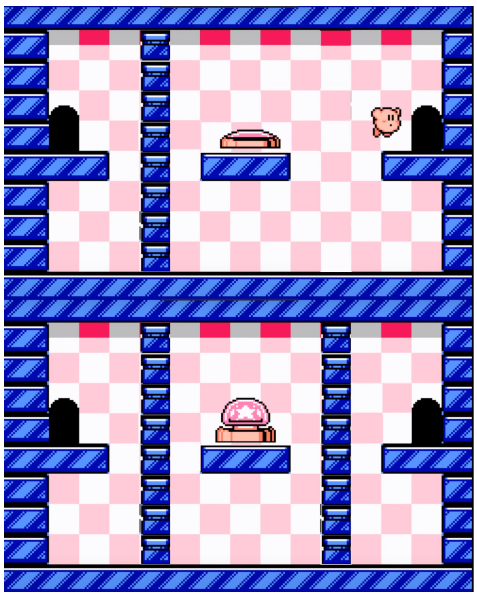


Figure 3: Activated 2-toggle lock

An Implementation of a 2-toggle lock inside Kirby's Adventure can be seen in Figure 2

We can see the behavior of our lock after Kirby traverses the top path in Figure 3. This movement is also reversible once Kirby comes back from the top path and hits the switch again to revert the gadget to its original state.

□

Theorem 3 *Kirby's Adventure is PSPACE-Complete.*

Proof. One aspect of our game is the extensive use of doors to move between gadgets. These doors serve as transport to and from different rooms. These rooms are linked to hallways or branches which allow simple traversal through the graph and Kirby can travel any viable path he desires non-deterministically to reach the end goal.

We can summarize our graph into an NCL by using a combination of a 2-toggle lock gadget outlined in Theorem 1, and traversal gadgets that contain hallways and branches in them.

If Kirby reaches the end goal, then the graph provided is viable and "winnable". Since this traversal is nondeterministic and every path must be checked, a Turing Machine that computes if a path is achievable will take polynomial time on its tape to compute it. Therefore it is PSPACE-Complete. □

4 Conclusion

In conclusion, we have demonstrated that Kirby's Adventure for the NES can be classified as PSPACE-Complete. By introducing a specialized 2-toggle Lock Gadget utilizing in-game switches, we show that solving one-player motion planning with this gadget is PSPACE-Complete.

Our work contributes to the field of game complexity theory, offering a unique perspective on the computational intricacies of Kirby's Adventure. The established framework suggests that analogous games in the Kirby series, featuring similar switch mechanics, can also be proven to be PSPACE-Complete using comparable reductions.

In the case where a reduction using switches is not viable, there are similar repeatable mechanics in place such as Kirby's ability to use enemies' abilities to surpass obstacles.

This research opens avenues for further exploration into the computational complexities of video games, enhancing our understanding of the relationship between game design and computational complexity.

Finally, this result reinforces the idea that if a 2-toggle lock gadget can be implemented in a proposed game following the outlines in [2], the game is PSPACE-Complete.

References

[1] G. Aloupis, E. D. Demaine, A. Guo, and G. VigliettaJürgen. Classic nintendo games are (computationally) hard, 2015.

[2] E. D. Demaine, D. H. Hendrickson, and J. Lynch. Toward a general complexity theory of motion planning: Characterizing which gadgets make games hard, 2020.

[3] R. A. Hearn and E. D. Demaine. Games, puzzles, and computation, 2009.

[4] K. Wiki. Kirby (character), 2023.

[5] K. Wiki. Kirby's adventure, 2023.

[6] T. Wylie. Why hack research is the best, 2019.

Solving Constrained Triple Triad Card Game in Polynomial Time

Pablo Santos *

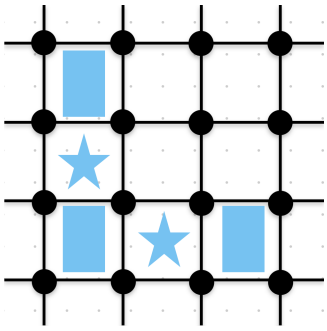


Figure 1: Constrained Triple Triad Problem

Abstract

Triple Triad, is a two-player card game originating from the Final Fantasy series, that served as an intriguing subject for computational complexity analysis. In this study, I try to model Triple Triad as a combinatorial game, focusing on the inherent computational challenges associated with determining optimal strategies and outcomes. While originally trying to prove P-Space Completeness through a quantified boolean formulas (QBF) approach, given the time constraints and lack of experience using such strategies I instead deviated to a simpler one player solution that asked if player had a winning move given certain amount of moves.

1 Introduction

As stated triple triad is a 2 player card game that is played in a 3x3 grid. Each player takes turns placing card with the capturing of cards occurring given that the player places a higher value card adjacent to their opponents card. However given the time given I constrained the rules so that it fit a one player format. Instead of it being turn based the player is given certain amount of cards, and a grid with already placed enemy cards. Anytime any of the cards given are placed in front of enemy cards they immediately capture those cards. However to not have such a limit environment I also added that the size of the grid would not remain constant and can change along with the enemy card placement. The goal is to find the complexity of such a game. I briefly highlight some related work in Section 2, and then provide the definitions and results of our work in Section 4. I then conclude in Section 5 and point towards the general research goals for this work.

*Department of Computer Science, University of Texas Rio Grande Valley, Pablo.Santos01@utrgv.edu

2 Related Work

Given my original plan for solving complexity for 2 players I did some research surrounding general approaches for such proofs. I read about constraint graphs and constraint logic as well as got introduced to time and space complexity associated with different types of two and one players games in the thesis paper by Erik Demain [2]. However I was quick to realize researching about this in detail to the point of utilization was wishful thinking and instead moved to the more constrained model. Such implementation was much closer to what i was used to and with a simpler approach I began to look for approaches to solving it using graph theory and dynamic programming. I went over different graph problems that could apply to this problem including vertex covers[1], and Hamiltonian Paths. [3].

3 Failed Attempts

I originally started this project with trying to compare a more general version of the problem that left the end behavior of the captured enemy cards ambiguous. I attempted this by trying to reduce the vertex cover problem to this variation of Triple Triad however this didn't work as well initially expected. I also looked at certain Hamiltonian reductions to see if there was anything I could use however again found nothing. I also began looking at more widely used approuches of reducability like the general satisfiability problem and while I am sure there is some connection between these problems I had to deviate in favor of something that I knew I would be able to solve and implement in a reasonable amount of time.

4 Results

My results showed that finding a combination of moves to win in k moves could be done in O(kn2) where n is the size of the rows and columns of grid and k is the amount of cards given to player. The approach for this problem is using a greedy algorithm that find the nodes in graph, or positions in grid with most surrounding enemy cards. Once you place a card there and capture those cards, traverse the grid again finding the new most populated grid cell. Place your next card in this position and keep repeating this for your k cards. If after k cards there are still enemy cards that means there is no way to win that board with the given cards.

Proof. Given that choosing a certain placement for a card doesn't affect future placements for the rest of the cards other than blocking its own position it doesn't limit the choices available. Because of this we can use a greedy

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

approach that takes most populated grid cell for every loop and guarantee most efficient card placement. □

5 Conclusion

While the result shown is not groundbreaking it does bring an interesting point about the configurations of the triple triad game that are solvable in polynomial time.

However there are certainly less restrictive alterations to this game that could prove to show more useful results including versions where flipping an enemy card twice keeps it in it's original position. This iteration would certainly be closer to a complexity of NP-Hard as the decision of card placements in certain positions would alter future paths. An example that demonstrates this would be when you place a card A somewhere that rotates an enemy card B that is diagonal to another enemy card C in the corner of the grid. Doing this would make it impossible to solve the problem without taking back moves as trying to capture card C would ruin the orientation of card B. I would also want to focus my efforts on solving the original version of this game that keeps the 2 player aspect. For the future I seek to explore these variations further and find the complexity for these more more difficult models.

References

[1] A. Ahmadi. Bipartite matching and vertex covers.

[2] J. t. . T. Erik, Demain, Dylan, Hendrickson, 2020.

[3] F. Gotti. Combinatorial analysis: Hamiltonian cycles and paths.

Coin Flips and PATS

Adrian Salinas * Alberto Avila Jimenez †

Abstract

In this paper, we introduce a Chemical Reaction Network (CRN) system that simulates flipping a fair coin (Schweller). Using a deterministic CRN, we demonstrate a technique which somewhat mimics John von Neumann's procedure to obtain a fair result from a biased coin. Further, we augment the CRN into a Step-CRN system, which allows for a simpler approach to solve the problem. We also generalize the number of unique tile types to required to make all $w \times 1$ and $w \times 2$ 2-PATS (Wylie).

1 Our Results

1.1 CRN coin flip

Theorem 1 *A CRN system with initial configuration I and a deterministic ruleset R can simulate a fair coin flip.*

Proof. Given an input species A, we use the ruleset in Table 1 to simulate a coin flip. The first rule turns A into H_1, H_2, T_1 and T_2 which are used to represent the different cases that can happen. When H_3 and T_3 are produced, they represent the result of the first flip. This product then reacts with H_1 and T_1 respectively as shown in the third and fourth rules. The product of these rules are H and T representing the result of a second flip and the ultimate outcome.

In the case where the count of $A > 1$, we will have multiple copies of H_1, H_2, T_1 and T_2. This is not a problem because having two different coin flips at the same time is cancelled out by the last rule of the ruleset. This rule takes two of our output species and resets the cycle by turning them into an A. By doing this we ensure that the different flips happening at the same time do not mess with the single species output. Eventually, when the CRN stops reacting there will only be either H or T left representing the ultimate result of our coin flip.

A tree for this system would look like the tree in Figure 1. This that the probability for either heads and tails flips will be the same. The other half is the chance that the combination of the two reactions results in a repeat to flip again. It is important to note that when at least 2 different trees happen, their outcome will eventually cancel out to one same tree. □

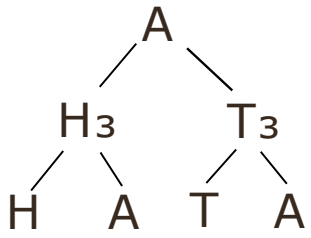


Figure 1: Probability tree for the possible outcomes of the CRN system

$A \to H_1 + H_2 + T_1 + T_2 + x$
$H_1 + H_2 + x \to H_3$
$T_1 + T_2 + x \to T_3$
$H_3 + T_1 \to H$
$T_3 + H_1 \to T$
$H + T \to A$

Table 1: A deterministic set of rules used to simulate an unbiased coin flip

Theorem 2 *A Step-CRN can simulate a fair coin flip in only 2 steps.*

Proof. When augmenting the CRN model into a Step-CRN, the problem becomes much easier. Having control of what goes in to the system gives enough power to the experimenter to disregard the input species. The first step does not have anything relevant as it does not affect anything happening in the second step. Table 2 shows the relevant rules for the second step in the system. In the second step, we add one count of species x and y. y will turn into H and T which will represent the outcome of our coin flip. x will then react to one of H or T, leaving the other one behind as the final result of the coin flip. □

$y \to$H + T
$x + H \to \emptyset$
$x + T \to \emptyset$

Table 2: A deterministic set of rules used to simulate an unbiased coin flip

1.2 2-PATS

Lemma 3 *The least amount of tile types required to make all $w \times 1$ 2-PATS is w.*

Proof. This can be proven by contradiction. If we assume that there exists a set of tiles that can build any $w \times 1$ 2-color pattern, then that implies that at least one tile is used more than once. However, due to the nature of certain patterns, this could lead to the possibility of non-deterministic loops. In other words, if a tile is used more

*Department of Computer Science, University of Texas Rio Grande Valley, adrian.salinas08@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, alberto.avilajimenez01@utrgv.edu

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

than once, then there is no way to guarantee that the loop will only occur the amount of times you want it to. For instance, consider the pattern where there first $w-1$ tiles are black tiles, and the last tile is white. Using less than w amount tile types implies that at least one of the black tiles is used more than once. This means that the sequence created by that black tile leads to the creation of that same black tile at the end of the sequence. And because all of the south glues are the same color (gray), there is nothing unique to distinguish whether that sequence should repeat again or not. This means that there is a nondeterministic chance of the sequence repeating an inappropriate amount of times, thus creating the wrong pattern. Therefore, it is impossible to create certain $w \times 1$ patterns with less than w tile types. □

Lemma 4 *The least amount of tile types to all $w \times 2$ 2-PATS is upper bounded by $w + 2$.*

Proof. The bottom half of the rectangle is built independently of what is built from the top half. This is because in order to add tiles to the top half, there must exist a tile to the west and south of it. Without tiles here, the glue strength would be too weak for a tile can't be attached. So, based on lemma 3, the number of tile types required to build any pattern for the bottom half of the rectangle is w.

Any pattern in the top half can be built using an additional 2 tile types. A white tile with a south white glue and the other glues gray, and a black tile with a south black glue and the other glues gray. These two tiles will build the top half, and the north glues of the bottom half tiles will encode the pattern for the top half. This can be done by having the north glue of the bottom tile be whatever color the tile above it should have. For instance, if a white tile should go above a given tile, then make the north glue of that tile white. This will cause the new white tile we made to attach; the same is true for black tiles. The bottom half was shown to already require w tile types, so making these north glues unique to the pattern of the top half would not require new tile types to be made.

Therefore, any $w \times 2$ 2-PATS can be made with $w + 2$ tile types, making that the upper bound. □

Lemma 5 *The least amount of tile types to all $w \times 2$ 2-PATS is lower bounded by $w + 2$.*

Proof. As mentioned in lemma 4, the bottom half of the rectangle requires at least w tile types to make any pattern. So, we know at least w tile types are needed. We can prove that the lower bound requires $w + 2$ tile types by contradiction.

Let's assume that there exists a set of w tile types to make any $w \times 2$ 2-PATS. We know that w tile types are needed to build certain patterns for the bottom half, so we must build the top half without any new tile types. The example shown in Figure **??** would be impossible. To attach the white tile on the top left, we cannot use the white tile we already have because that tile can not have a gray west glue. If that white tile did have a gray west glue, then that implies that the black tile before it must've had a gray east glue. But if that was the case, then the black tile in the bottom left would compete with the white tile, thus introducing the possibility of the wrong pattern forming. Therefore, in order to attach the top left tile, we must make a new white tile that has a gray west glue. It is impossible to make all $w \times 2$ 2-PATS with only w tile types.

Let's assume that there exists a set of $w + 1$ tile types to make any $w \times 2$ 2-PATS. As explained in the previous paragraph w tiles are needed to build the bottom half, and in the example in Figure **??**, a new white tile is needed for the top left tile. So, when trying to build that pattern, we must build it without introducing any new tiles. If all of the north glues for the remaining bottom tiles were gray, then we could reuse some of the tiles we already have. If we have the east glue of the top left white tile set to attach to the same tile type as the second to last black tile (coordinate: [7,1]) on the bottom half, then the next 3 tiles can be attached. We can then set the east glue of the bottom right white tile attached to the last black tile on the bottom half (coordinate: [8,1]). This will cause the next 3 tiles to attach correctly, but the last two tiles will be incorrect, as shown in Figure **??**. This is because the pattern is a sequence that leads to itself, thus creating loop that can't be controlled unless a new tile type is created. Therefore, it is impossible to make all $w \times 2$ 2-PATS with only $w + 1$ tile types.

Lemma 4 showed how to make any $w \times 2$ 2-PATS with $w + 2$ tile types, so that is our lower bound. □

Theorem 6 *The least amount of tile types required to make all $w \times 2$ 2-PATS is $w + 2$*

Proof. Lemma 4 and 5 showed that $w + 2$ tile types are the upper and lower bound, respectively, to make all $w \times 2$ 2-PATS. Therefore, at least $w + 2$ tile types are required to make all $w \times 2$ 2-PATS. □

2 Conclusion

We proposed two different solutions to the coin flip problem. One with a normal CRN and a deterministic ruleset. Then, with the augmentation to a 2 Step-CRN. We also determined the minimum amount of tile types to make any $w \times 1$ and $w \times 2$ 2-PATS. ...

References

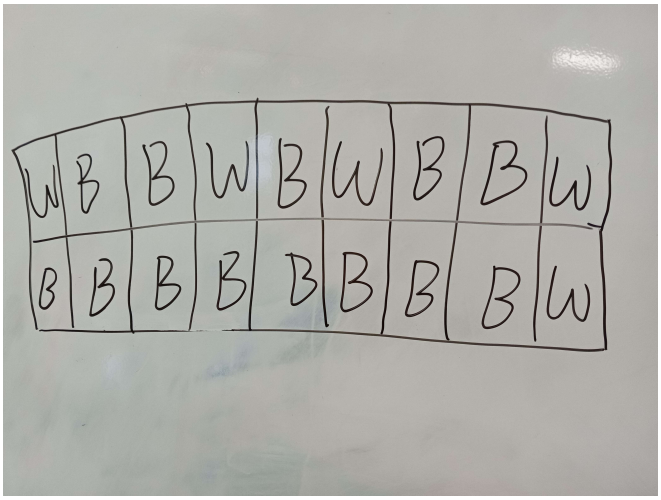


Figure 2: W = White; B = Black. Example pattern.

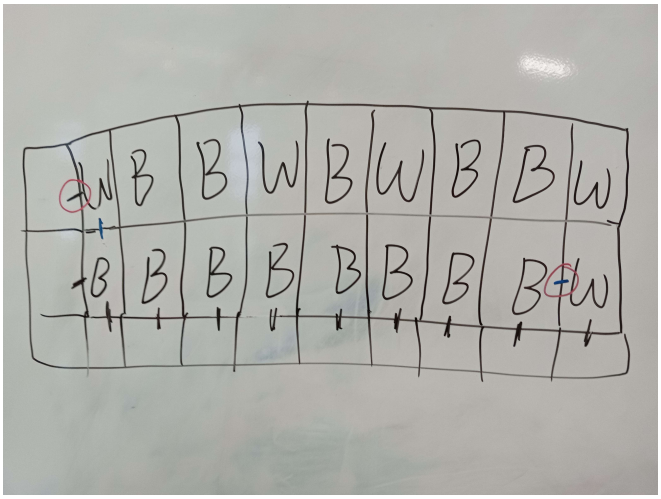


Figure 3: W = White; B = Black. Example of new tile typed needed.

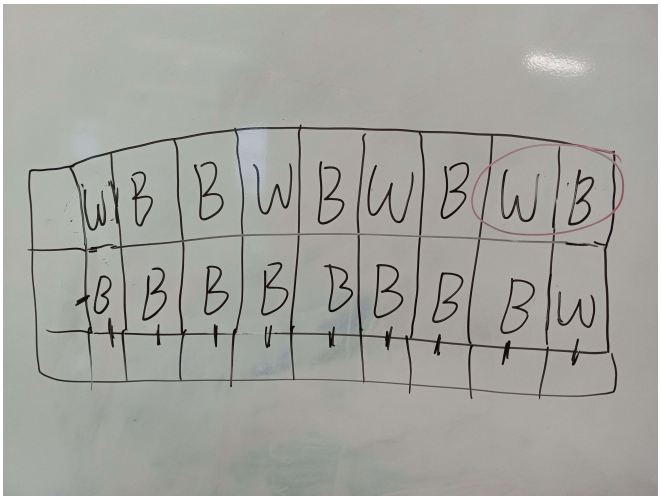


Figure 4: Example of error (circled in red) when trying to make pattern with $w + 1$ tile types.

Optimization to Pattern Assembly Tile Systems using Native Multithreaded Processing and Low-Level Cache Optimization

Carter Vavra * Sarah Evans †

Abstract

A Pattern Assembly Tile System (PATS) is a model of self-assembly which takes a set of tile colors, and permutations of grids of colored *Wang tiles* which are generated with respect to a grid base – the *L-Seed* – and computes which tile permutations are valid based on the input tile color configuration. The scope of this paper is to develop an optimized generation and grid validation model which calculates what permutations of a grid contain the most unique tiles based on [1], whose Java implementation necessitated a week's time of computation.

1 Introduction

A PATS model generates permutations of possible grid configurations, whose tiles are connected by *"glues"* which correspond to colors on one of four sides of a Wang tile. The cardinality of the grid is specified with an *L-seed*, where the dimensions of the L-seed $(w+1)\times(h+1)$ define a grid of dimensions $w\times h$, whose left and bottom borders connect to the L-seed with *white* glue.

For instance, a 2×3 grid of colored Wang tiles G is connected to a 3×4 L-shaped piece, where the bottom 3 tiles of the inside grid are connected to 3 L-seed tiles, and the 2 leftmost tiles of the internal grid connect to 2 L-seed tiles to the left. A single tile (described by $G[2][0]$) connects to 1 left L-seed tile and 1 bottom L-seed tile. This is the "bottom-left corner" of the inside grid (note that the corner of the L-seed does not connect to the internal grid).

Wang tiles – by virtue of *being* – have four sides, each side having a glue color. Wang tiles can only be connected to other Wang tiles whose opposite side shares the same color as the corresponding side of the first tile. That is, a tile whose top glue color is *red* connects to another tile *if and only if* another tile has a bottom red glue color.

The result of a successful permutation computes the hardness for all possible patterns. The hardness in this case is a min max problem in which we find out which pattern contains the most permutations of minimally sized tile-sets.

In Section 3, we uncover the implementation of the original Java PATS algorithm, and highlight its various issues that were improved upon in Section . The set of possible permutations generated (before being validated) are described in Section 2.

*Department of Computer Science, University of Texas Rio Grande Valley, carter.vavra01@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, sevans.dev@gmail.com

2 Possible Permutations

There are $4wh$ possible glues per tile; however, because every pair of adjacent tiles is limited to a single glue type, the 4 is factored out, and the number of possible glues is denoted by wh.

Considering the number of glues, and k possible tile colors, the number of possible tiles is denoted by $k(wh)^4$.

In the of our PATS model, the number of unique tile colors is 2 (black or white), thus, the number of possible tiles is denoted with $2(wh)^4$.

In the case of this paper, we are computing the number of valid tiles of a 2×3 grid, the total number of possible unique tiles is denoted by $2(2 \times 3)^4$, or 2592 unique possible tiles.

The calculation of how many possible permutations – and valid permutations – can be generated from any number of unique tiles is NP-Hard [2], meaning that the computation must be done via brute force computation. The setbacks of the Java implementation in [1] are due mostly to the NP-hardness of this algorithm.

3 Optimization of Tile Permutation Computation

In our effort to optimize the computation of Tile Permutations in the PATS model, we undertook several key steps, detailed below:

3.1 Native Code Generation with C++

The first low-hanging fruit optimization we took was providing an implementation in C++ with clang, takes full advantage of the LLVM optimization suite and native code generation.

3.2 Leveraging Constraints for Efficient Tile Generation

- **Issue in Java Implementation:** The original Java implementation faced performance bottlenecks due to the iterative generation of all possible tiles, many of which were invalid.
- **Optimization Strategy:** We shifted from a brute-force approach to a more efficient method that guarantees the generation of valid tiles.
- **Outcome:** This optimization significantly reduced the computational load by eliminating the creation and subsequent validation of invalid tiles.

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

3.3 Precomputing Valid Tile-Sets

- **Rationale:** The strategy for precomputing valid tile-sets in PATS is adapted based on grid size. For small grids, identified by parameters w and h, a dedicated program generates a header file with an array Look-Up Table (LUT) of valid tile-sets, indexed by West and South glue constraints. This is efficient for smaller grids, reducing complexity during permutation generation. Conversely, for larger grids, the extensive size of a comprehensive LUT leads to high spatial and computational costs. We address this by employing an LRU (Least Recently Used) cache mechanism, which dynamically computes and stores valid tile-sets for current permutations, offering quick access for future iterations.
- **Implementation:** A greedy algorithm was developed to precompute and store these tile-sets, tailored to the size of the grid.
- **Benefit:** This approach ensures rapid access to valid tiles during permutation generation, thereby enhancing the algorithm's overall speed and efficiency, especially in handling varying grid sizes.

3.4 Asynchronous and Generative Permutation Generation

- **Approach:** We adopted an asynchronous, generative approach to permutation generation, utilizing a Breadth-First-Search (BFS) algorithm.
- **Worker Allocation:** We allocated workers in a thread pool to independently compute permutations, enabling parallel processing.

3.5 Accounting for Tile Colors

- **Consideration of Color Variability:** Our tile-set generation initially did not account for the variable number of colors (denoted as k).
- **Application to Matrix Permutations:** Upon finding a valid permutation of the matrix, we applied all possible color patterns, thereby calculating the hardness for each pattern.
- **Impact:** This step ensures that the algorithm comprehensively evaluates the complexity of each permutation, taking into account the full spectrum of color variability.

3.6 Bit-Packing Optimization

- *Bit-Packing* involves using sub-byte memory spaces for multiple variables in order to minimize the byte space needed for a set of class members, allowing in-register reads and comparisons that can be measured in cycles, as opposed to boolean and function invocation overhead which unoptimized value comparisons create. This decreases loading time significantly.

4 Our Results

The current evaluation of the C++ implementation's performance remains inconclusive due to time constraints that hindered unit testing. Nevertheless, it is noteworthy that mathematical computation demonstrates a clear advantage on native software systems compared to interpreted ones. Additionally, the introduction of multithreaded processing significantly reduces computation time by maximizing CPU utilization and concurrently executing various segments of the permutation algorithm.

5 Conclusion

In conclusion, this research introduces significant optimizations to the Pattern Assembly Tile System (PATS) algorithm, primarily through the implementation of native multithreaded processing and low-level cache optimization. The transition from a Java-based system to a C++ implementation, coupled with these optimizations, demonstrates a substantial improvement in computational efficiency.

The use of precomputed tile-sets, asynchronous permutation generation, and LRU caching mechanisms, along with multithreaded processing, substantially reduces computation times and enhances the algorithm's ability to handle larger and more complex grid sizes. This approach not only addresses the NP-hard nature of the problem but also leverages modern hardware capabilities to their fullest extent.

Moreover, the shift to a native system underscores the importance of considering system-level optimizations in algorithmic design, especially for computationally intensive tasks like those presented in PATS. Future work could focus on refining these optimizations, exploring the use of GPU processing, and expanding the algorithm's applicability to other complex tiling and computational geometry problems.

The results obtained in this research open new avenues for exploring pattern assembly models and contribute significantly to the field of computational geometry, paving the way for more efficient and powerful computational models in the future.

References

[1] Algorithmic Self-Assembly Research Group (ASARG). Hardest-k-pats-patterns. https://github.com/asarg/Hardest-k-PATS-Patterns, 2019. [Online; accessed 12-November-2023].

[2] L. Kari, S. Kopecki, P. E. Meunier, M. J., and S. Seki. *Binary pattern tile set synthesis is np-hard.* 2017.

Attacking a Game

Hector Lugo * Arturo Meza † Diego Adame ‡ Juan Velazquez§

Abstract

In the quest for advancing artificial intelligence within strategic board games, our team at Hack Research 2023 has embarked on a cutting-edge project to design and refine a competent AI to beat a player in Yavalath. Yavalath, a game known for its simplicity in rules yet complexity in strategy, presents a unique challenge for AI development. The objective of our project is to create an AI that not only understands the intricate patterns of play but can also adapt and react to human opponents with advanced gameplay strategies. Through iterative training processes and machine learning techniques, we aim to enhance the AI's proficiency to a level where it can consistently outperform human players. The measure of success for our initiative is the AI's ability to win against users, signifying a breakthrough in game-based artificial intelligence applications. This endeavor not only contributes to the field of recreational AI but also provides insights into the capabilities of AI in pattern recognition, decision-making, and strategic planning.

1 Introduction

In the realm of abstract strategy games, Yavalath stands out as a captivating and intellectually challenging contest of wits. Developed by Cameron Browne in 2007, Yavalath belongs to the genre of connection games, where the primary objective is to form a specific pattern on the game board by connecting a predetermined number of pieces. What sets Yavalath apart is its elegant simplicity and the strategic depth it offers, making it a fascinating subject for exploration in the field of game theory and artificial intelligence.

Yavalath is played on a hexagonal grid, adding an intriguing geometric dimension to the gameplay. The game features two players, each assigned a unique color, typically black and white. The objective is to be the first to create a sequence of four pieces in a row, either orthogonally or diagonally. However, the twist that makes Yavalath especially compelling is the forbidden move rule. On each turn, players must decide where to place their pieces on the board, but they cannot create a sequence of three of their pieces in doing so. This prohibition introduces a layer of complexity that transforms the game into a delicate balance of offense and defense.

*Department of Computer Science, University of Texas Rio Grande Valley, hector.lugo02@utrgv.edu

†Department of Computer Science, University of Texas Rio Grande Valley, juan.velazquez03@utrgv.edu

‡Department of Computer Science, University of Texas Rio Grande Valley, arturo.mezacanales01@utrgv.edu

§Department of Computer Science, University of Texas Rio Grande Valley, diego.adame01@utrgv.edu

We briefly highlight some related work in Section 2, and then provide the definitions and results of our work in Section 5. We then conclude in Section 6 and point towards the general research goals for this work.

2 Related Work

There is one notable implementation of a Yavalath AI. [2] creates an engine and AI player for the two-player game. The game has 61 hex tiles, and internally, the game state is represented using a pair of 64-bit bitboards, one for each player. The rules are encoded as two-bit mask tables, and detecting wins and losses is just a handful of bit operations. It uses a crude CLI since it mainly focuses on creating an AI player. The input must be in Susan notation. For example, the upper-left tile is a1, and the bottom-right tile is i5.

[1] focuses on the reinforcement learning Python package SIMPLE, which implements self-play for multiplayer games. It implements Proximal Policy Optimisation (PPO), with a built-in wrapper around the multiplayer environments that handles the loading and action-taking of opponents in the environment. The wrapper delays the reward to the PPO agent until all opponents have taken their turn. It converts the multiplayer environment into a single-player environment, constantly evolving as new versions of the policy network are added to the network bank.

3 Command Line Environment

3.1 Printing the Board

Yavalath has 61 hex tiles arranged as a hexagon, so the distribution looks like Table 1. In order to achieve this distribution, we created a function $f(x) = -|x-5|+9$ that would generate this distribution.

However, to ease the user experience, we also wanted to print a hexagonal board with a legend for each row. The basic architecture for each row is $spaces + letter + dots + newline$.

To print the spaces we use $f(x) = |x-5|$ where $x \in \{1, 2, \ldots, 9\}$, the function $f(x)$ yields an output in the range $\{0, 1, 2, 3, 4, 5\}$. We then output a character determined by the value of q where $q \in \{1, 2, \ldots, 9\}$. It uses the `ord` function in Python to get the ASCII value of the character 'a', adds q to it, and then converts the result back to a character using the `chr` function. The resulting character is printed, followed by a space.

Printing the dots is challenging since we must print the game's current state. The state of the game is stored in ϕ, a 2-dimensional array made up of nine rows and nine columns. We will print each column of each row in ϕ. We iterate through each cell and check if the contents are a

This is an abstract of work done at $\mathcal{H}\alpha c\kappa\mathbb{R}$ 2023. It should be considered a workshop preprint.

space, the default value; if so, we print '.'. Otherwise, we print the cell's content to the terminal, either a $'X'$ or $'O'$. Furthermore, before the next iteration, we print a space to create space between the cells in the board. We then finally print the newline character after printing the cells' contents for the given row.

Row	Cells
1	5
2	6
3	7
4	8
5	9
6	8
7	7
8	6
9	5

Table 1: Distribution of spaces per row

3.2 Win-loss Conditions

In Yavalath, only rows are considered for the win-loss conditions, and 4 in a row is a win, while 3 in a row is a loss. Each turn, the player can either make a winning move, a losing move, or a move that continues the game.

We created a function that iterates through the set of movement vectors for the game, which is six since the spaces are hexagons. We also check in the opposite direction since it is not guaranteed that we will be at the start or end of a line.

4 Self-play AI

In our research, we apply Proximal Policy Optimization (PPO)—a state-of-the-art policy gradient method—to train a neural network for decision-making in Yavalath. PPO mitigates the risk of training instability by optimizing a surrogate objective function that constrains policy updates, thus enabling steady learning. The AI developed for Yavalath, which operates on a 9x9 grid, utilizes PPO to refine its strategy by selecting moves based on a calculated probability distribution. Post-game policy adjustments are made to enhance the likelihood of future success.

PPO's resilience and adeptness in managing Yavalath's expansive action space—spanning 61 possible moves—make it an apt choice for the complexity of board game strategies. We have tailored the PPO algorithm to align with Yavalath's distinctive gameplay, integrating it into a custom environment within the OpenAI Gym framework to leverage its extensive AI training and benchmarking utilities.

5 Our Results

In the results of our study, we observed that the AI model exhibits a functional level of play against human opponents. However, its performance is not consistently superior in terms of achieving victory. The model demonstrates variability in its gameplay, with periods of advantageous positions not always translating into wins. Furthermore, it appears to struggle with executing conclusive strategies to secure a victory or effectively countering the opponent's moves in critical situations. This suggests that while the model can engage in the game competently, there is significant room for improvement in its tactical acumen and endgame execution.

6 Conclusion

In conclusion, our initiative to integrate artificial intelligence into the realm of Yavalath has produced an AI that understands the game's intricacies and can compete with human players. Utilizing Proximal Policy Optimization, the AI has shown competence in game participation but has not yet achieved consistent superiority in securing victories. The variability in performance and the AI's occasional inability to capitalize on advantageous positions or counter opponents effectively highlight areas for further refinement. Moving forward, enhancing the AI's strategic depth and decision-making through advanced machine learning techniques remains a crucial objective for achieving a level of play that consistently surpasses human expertise.

References

[1] D. Foster. How to build your ai to play any board game. *Medium*, 2021.

[2] C. Wellons. Yavalath - a strategy game. 2019. @online-github.

Author Index