Improving Human-LLM Interactions by Redesigning the Chatbot

Stanford CS224N Custom Project

Rashon
Department of Computer Science
Stanford University
rashonp@stanford.edu

Abstract

In this project, our task was to create a new Natural Language Interface that incorporates several key design practices aimed at improving human-LLM interactions. These design practices included preprocessing the input, adding dynamic prompt suggestions, grounding the model to a specific user, and making the output more interpretable. This new system is better at mapping users' intentions into actions, which improves the effectiveness of LLMs in applications. We show that this enhanced NLI is more beneficial to facilitating an interactive learning environment for users over standard systems.

1 Key Information to include

- External mentor: Alan Cheng (HCI PhD Student)
- Sharing project: With ongoing research in HCI/AI in Education

2 Introduction

Natural Language Interfaces (NLI) are an innovative enhancement of human-computer interactions (HCI), allowing users to communicate with systems using plain language. They act as a bridge, providing access to the extensive capabilities of Large Language Models (LLMs) to users. However, despite their potential, the lack of need for structured input can make it more challenging for users to translate their intentions into actions accurately. This is particularly true among inexperienced users who may not understand LLMs or how they work. A lot of the current NLI systems consist of various UI design flaws and/or do not incorporate key human-LLM design practices that help bridge the gap between a user's intention and actions. This lack of consideration often leads to users producing poorly defined prompts, which results in unhelpful responses from the LLM. This discrepancy between intentions and actions limits the accessibility of LLMs and reduces their effectiveness.

In this paper, we introduce a new NLI that incorporates several design practices that help facilitate human-LLM interactions. We discuss how current NLIs fail to encourage good interactions and how these practices can address these issues. We discovered that the new NLI system enhances interaction efficiency by making LLM output easier to understand and the structure of input easier to obtain. We show that this increased efficiency can provide numerous benefits in interactive learning environments.

3 Related Work

3.1 Cognitive Design Challenges in LLM Interfaces

As LLM research has picked up over the last few years, so has the study of their interfaces. Incorporating HCI principles into NLI systems is becoming more popular as LLMs are becoming more popular and widely available. Prior studies have laid the groundwork for identifying the challenges that these interactions face and how they can be enhanced. The work by Subramonyam et al. introduced a new HCI-based method for interacting with LLM interfaces (Subramonyam et al., 2023). This paper points out the various flaws that exist in user interfaces for LLMs, which makes it difficult to provide appropriate prompts and outputs that align with the user's intention. As a result, users face several problems, such as being unsure about whether the LLM can perform a task, difficulty in instructing the LLM to perform it, and the inability to evaluate the LLM's output. The authors argue that there is a need for a new interaction approach based on human-computer interaction to bridge the gap between envisioning, execution, and intentionality in LLM interfaces. This new method consists of a set of design patterns that aims to make the interfaces of LLMs more accessible and effective for all users. The key design patterns can be grouped into six major categories:

- **Visually track prompts and outputs:** use a visual interface to capture users' prompting and divergent thinking pathways
- **Suggest Ideas for Prompting:** offer users prompt suggestions to guide them
- **Provide Multiple Outputs:** give users several outputs based on a prompt to choose from
- **Make the Output Explainable:** prompt LLMs to explain their outputs and make them more interpretable
- **Use domain-specific prompting strategies:** custom prompting strategies can help tailor specific tasks and guide LLMs.
- **Allow for manual control of the output:** systems should allow users to edit the output of LLMs manually.

An NLI system that follows this approach allows users to predict outcomes and improve their interactions with LLM models.

3.1.1 Addressing the Limitations

My NLI system builds upon the work of Subramonyam et al. by incorporating many of their listed design practices while also addressing some of the pitfalls. To address a scenario where users have a hard time translating their thoughts into words, I introduce a new design practice: preprocessing. Preprocessing the input to help improve the context/clarity of the prompt before sending it to the model reduces the workload of the LLM. Instead of having to guess the user's intention from a poorly worded and ambiguous prompt, if given a well-defined prompt, the LLM can focus solely on providing a response to the query. To address a scenario where the NLI may be doing more work than needed (i.e., providing the user with too much information), we introduce a settings feature that allows users to tune the level of detail and amount of responses for each prompt that they send to the model. By building upon the design principles of human-LLM interactions, we are contributing to the ongoing practice of making LLM technologies more accessible and useful for all users.

3.2 Function Calling in NLI

Another inherent limitation of many LLMs is their inability to handle certain types of queries, such as questions that are past their knowledge cutoffs. For example, if you wanted to incorporate an NLI and LLM into a News app, you would not be able to ask questions like "What is the biggest story of the day?" because it doesn't have access to this information. However, the reasoning power of a lot of the latest LLMs allows them to execute external function calls to overcome these limitations (Kim et al., 2024). Function calling allows LLMs to reliably connect LLM capabilities with external tools and APIs (Eleti et al., 2023). It involves combining a list of candidate functions along with their associated definitions, and natural language input. Based on the user's prompt, the LLM is expected to choose the right function from the list and generate an executable function call with the correct arguments (Srinivasan et al., 2023).

Our NLI system incorporates function calling to allow users to engage with both the LLM and the associated app. This helps map intentions to actions by providing a way to directly execute user commands, often leading to more accurate task execution.

4 Approach

This NLI system was created within the context of a 2D educational mobile app. The goal was to test how improvements to human-LLM interactions can help facilitate interactive learning environments. For the purpose of our experiments, we set the educational topic to be geometry. This app was created from scratch using the Unity Game Engine and consists of two main pages: Chatbot and ObjectAlter. The Chatbot page allows users to ask questions about an educational subject (geometry in this case) using natural language input. It incorporates several of the human-LLM design practices mentioned above. The ObjectAlter page incorporates function calling to allow the user to manipulate a shape using natural language directly. It serves as a companion to the Chatbot in helping users understand geometry. Both pages use API calls to OpenAI's GPT-4-0613 model through an open-source OpenAI Unity Package.

4.1 Incorporating the Design Practices

The key design practices that we have chosen to implement for the Chatbot are as follows:

- **Design 1: Preprocess the input -** Preprocessing the input helps improve the context/clarity of the prompt for the model
- **Design 2: Suggest Ideas for Prompting -** offer users prompt suggestions to guide them
- **Design 3: Make the Output Explainable -** prompt LLMs to explain their outputs and make them more interpretable
- **Design 4: Use user-specific prompting strategies -** custom prompting strategies can help tailor specific tasks and guide LLMs.

4.1.1 Design 1

Design 1 addresses one of the limiting factors of LLMs: their use of natural language prompts as the primary way to interact with users. This means that users who struggle to articulate their intent may receive less than ideal responses from the model. This is especially true among grade school students who may not yet have fully developed their ability to transfer their thoughts into text. For example, a grade school user wanting to use ChatGPT may input the prompt "I wanna no what a punnet square is" instead of "What is a Punnett square?". Our approach to addressing Design 1 included looking into Text Style Transfer (TST) models (Fu et al., 2018). The primary purpose of TST models is to control the style attributes of text while preserving its content (Jin et al., 2021). In particular, we looked into the subtask of translating text from informal to formal. This would allow the translation of informal sentences and phrases like "Gotta see both sides of the story" to formal versions like "You have to consider both sides of the story" (Rao and Tetreault, 2018). Although some of the latest LLMs can still gather the intent behind ill-defined queries, the idea here is to reformat the prompt to make it less ambiguous and more defined before sending it to the model. That way, the LLM can just focus on producing a precise answer. However, implementing a TST model within a mobile app would be impractical for this use case since it would require either too much overhead performance or a cloud-based service to host the model. Instead, intelligent writing assistant services can be used. In fact, the authors in Rao and Tetreault (2018) used Grammarly to produce their corpus of informal-to-formal sentence pairs. However, since Grammarly's API is no longer available, we used an alternative called Sapling.

Sapling is a Language model copilot for enterprise applications that offers a variety of AI and NLP tasks. It includes a Sapling Rephrase API that provides LLM-powered paraphrase suggestions. This task includes taking an input sentence and producing a rewording that retains the original intent. For our purposes, we used Sapling to adjust the formality and clarity of input prompts before sending them to the GPT model. In this workflow, the user inputs a prompt, which is checked by Sapling to make it as well-defined and concise as possible, and then it is passed to the GPT model.

4.1.2 Design 2

Design areas 2-4 involved incorporating Envisioning in the Chatbot, which involves "the articulation of intentions, by imagining the output, to formulate the prompt for the LLM" (Subramonyam et al., 2023). These design principles were addressed through UI elements and prompt engineering. The approach for Design 2 involved creating a dynamic question prompt that is relevant to the current conversation and context. This was accomplished by keeping track of the current conversational history and providing that to an LLM to produce a relevant follow-up question. For example, if the first question provided to the model by the user was "How do we use Pi in geometry?" a suggested follow-up question could be "How do we use Pi in calculations?". The next follow-up question could be, "How do we find the surface area of a sphere using Pi?". This new feature can provide a tremendous positive impact on a user's chain of thought. These suggestions can help users obtain a guided thought process by accelerating idea generation, preventing dead-ends, and encouraging deeper exploration of the current topic. If the user's intent is to learn more about Pi's applications in geometry, then dynamic prompt generation is a good way to accomplish this without any extra effort from the user.

4.1.3 Design 3

Incorporating Design 3 involved creating UI elements and action buttons that make the output from the LLM more explainable. This was accomplished by adding the following actions to the model's outputs: Action 1: Simplify response, Action 2: Request more details, Action 3: Ask why. Selecting Action 1 automatically prompts the model to simplify its previous response to make it more understandable. A user may select this action if they find that the previous response from the model was too hard to follow (i.e., the model used keywords or terms that were too advanced). Action 2 simply prompts the model to expand upon its previous response. A user could select this option if they feel that not enough details have been provided. Action 3 attempts to gain insight into how the LLM reasons. Selecting this action prompts the model to explain why it answered the way that it did. As a way to effectively map users' intentions to actions, one of the goals of our NLI system is to help users understand the correlation between the inputs and outputs of an LLM. This involves looking into the model's thought processes. Having a conceptual understanding of how LLMs work can lead to more understandable and predictable results.

4.1.4 Design 4

Lastly, Design 4 involved tailoring the model's response to a specific user by adding tunable settings. These settings consist of three sliders that allow you to adjust the level of detail, number of responses, and complexity of the response. The first slider allows you to tune the amount of detail for the model's response, ranging from "short and concise" (i.e., a few sentences) all the way up to "very comprehensive" (i.e., a few paragraphs). Sometimes, the user may want very simple answers to their very simple questions. Our NLI gives users the power to limit or expand the output of the model without having to state it explicitly in their prompt. The second slider allows users to select how many responses the model should provide to their query (from 1 to 3 responses). Allowing users to see the variance in responses to the same query can give further insight into how LLMs work. Slider 3 allows the user to scale the complexity of the model's responses to their current level. The slider contains 5 levels: elementary student, high school student, college student, grad school student, and expert. If the user selects the first level, then the model will answer as if it is an elementary school teacher talking to a student. The responses are made really simple and easy to understand. If the user is on the fifth level, then the model responds as if it is a leading expert in the field talking to a fellow colleague. Answers on this level would be more technical and may assume that the user has the relevant background knowledge. Providing these tunable settings in our NLI system allows users to create a personalized experience with the LLM, increasing their effectiveness.

4.2 Adding Function Calling

Through function calling, the ObjectAlter page allows users to manipulate a 3D object through natural language. This page allows users to explore Geometry even further by providing an interactive learning demo. Through a text prompt, a user can ask for the object on the screen to be rotated along any of its axes, change the color of the object, or change the speed of the rotation. The functions responsible for making these changes in the code are passed to the LLM as a string along with the user's prompt. This text string contains important information about each function, including its name, a description of what it does, and details on the parameters it takes. The LLM determines which function to call and the parameters based on this information and returns a JSON object with this information. For example, if the user states, "Rotate the cube along its Z-axes with a speed of 120",

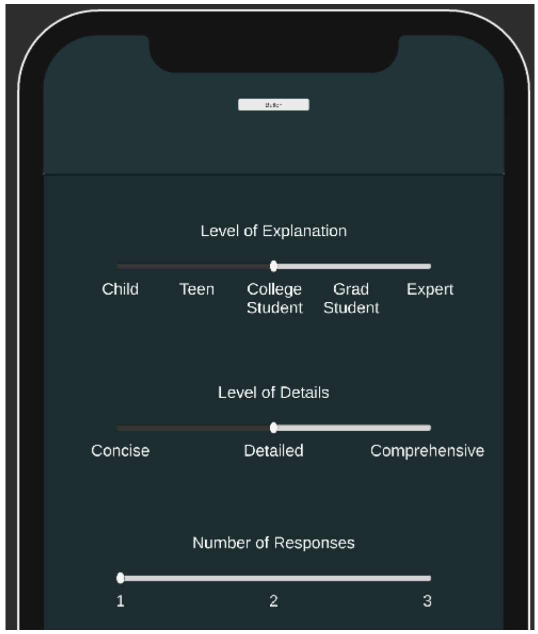

Figure 1: The UI of Design 4's settings

the LLM would determine the parameters to be $speed = 120$ and $rotation = Vector3.forward$. Its response to this prompt would be a JSON object containing the name of the rotate function and these parameters. This JSON object could then be provided to the action function to update the object.

5 Experiments

Due to the nature of this project, fully evaluating the effectiveness of this NLI system as an improved method of human-LLM interactions for an interactive learning environment would require an extensive case study. Some of the quantitative methods would include comparing this NLI to others and comparing the engagement metrics, such as the student's interaction patterns, and tracking their learning pace, which is the time it takes for the student to achieve a learning milestone. Some of the qualitative methods could include conducting interviews or studies to obtain feedback. These are metrics to obtain in future iterations of this project.

5.1 Experimental Details and Evaluation Method

For this paper, experiments consisted of testing how well our NLI system adheres to the different design practices incorporated. This included testing different UI elements, API services, and prompts and comparing the results. To determine how successful our NLI system is at implementing each design practice, we provide sample inputs and determine if the output demonstrates a successful translation of the user's intent into action. We compare these results to other NLI systems. There are no quantitative results in this paper.

5.2 Data

To compare the results of our NLI systems to others, we are using a large-scale real-world LLM conversation dataset that contains conversations between users and 25 different LLMs. We used this to determine if our outputs adhere to our goals better than the design-less NLI systems.

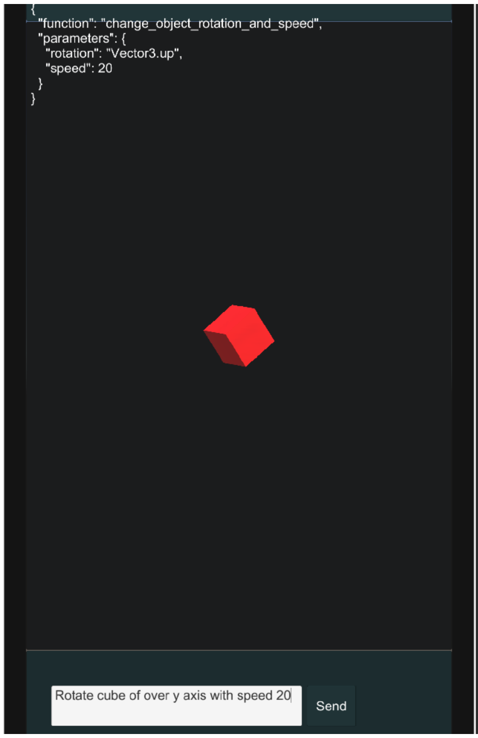

Figure 2: An example showcasing how natural language can be used to change the object.

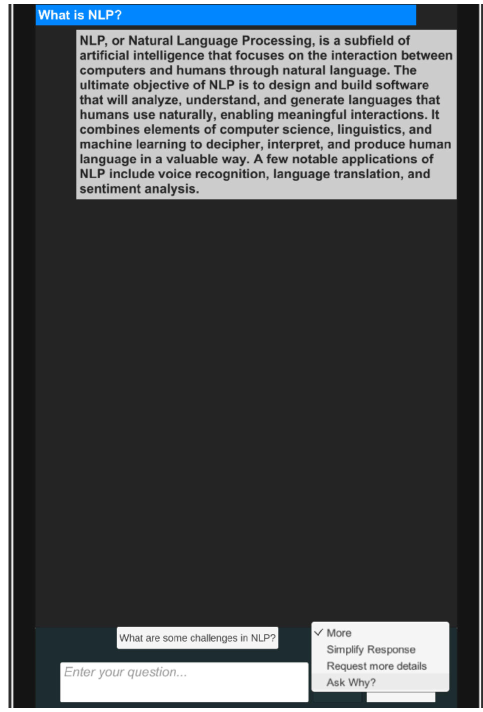

Figure 3: The UI of the ChatBot

5.3 Results And Analysis

For Design 1, we tested Sapling's performance as a TST model, translating text from informal to formal. We found that the rephrase API works exceptionally well. It was able to convert poorly worded phrases like "Why it snow when it cold?" to the properly worded concise phrase "What causes snowfall when the weather is cold?". However, for sentences that are already concise and well-structured, Sapling acted as a "reworder", slightly changing the nuance of the prompt. For example, the phrase "How do we find the surface area of a sphere using Pi?", was translated to "What role does Pi play in computing the surface area of a sphere?". This slight change in wording can cause the LLM to give a slightly different answer and, overall, leaves the user with less control over their prompt. Due to this, we decided to make this "rephraser" setting optional for users.

To test the effects of Design 2, we started with a sample prompt and then selected the auto-suggested follow-ups to see where it would take the conversation. In the first example of Figure 4, we started off the chain by inputting "What is geometry?" into the Chatbot. After returning an answer, the Chatbot then selected the follow-up prompt, "How can we use geometry to measure the size of different shapes?". Then, after selecting this prompt, it answered again and followed up with, "What are some examples of everyday objects that can be measured using geometry?". After our testing, we concluded that this feature can successfully suggest content that can help the user obtain a guided thought process. Testing Design 3 and 4 also involved starting with sample input prompts and

User input prompt → Auto suggested chain of follow-up prompts

What is geometry? → How can we use geometry to measure the size of different shapes? → What are some examples of everyday objects that can be measured using geometry?

How many sides does a cube have? → How many angles does a cube have? → What is the sum of all the angles in a cube?

Figure 4: An example demonstrating dynamic prompt suggestions

observing how our NLI system affects the outputs. Figure 5 and Table 1 show the effects that the different action items and configurations have on the starter prompts "What is a 4D shape?" and "What is geometry?". Figure 5 shows that the user does not need to insert any prompt other than the original to get additional information. The idea behind having buttons that automatically generate prompts for the user's intent is to lighten the load on inexperienced users who may not know the best prompt to use or that this is an option that's available to them. Table 1 demonstrates how our NLI system can adjust the level of complexity and details to match the ability of the user. These customizable options allow for a variety of users of different intellectual strengths to still get the most out of the LLM.

User: What is a 4D shape?

ChatBot: A 4D shape, or a four-dimensional shape, is a geometric figure that has four dimensions- typically interpreted as three spatial dimensions plus one additional dimension. In mathematics, these are often represented using tesseracts or hypercubes. On a conceptual level, envisioning these shapes can be challenging because we live in a three-dimensional world and our brains are wired to conceive shapes in only three dimensions. So a 4D shape, which includes length, width, height, and a fourth dimension, often referred to as time in physical reality, remains abstract and generally beyond our ability to visually comprehend.

User: Selects "Simplify Response"

ChatBot: A 4D shape is a shape that doesn't only have the three dimensions we're used to (length, width and height), but also has a fourth one which we can't see or identify easily in our daily life. We can't easily imagine these shapes because we live in a 3D world. Think about it as a 3D cube but with an extra dimension added, often thought of as time.

User: Selects "Add more Details"

ChatBot: Sure, a 4D shape extends beyond the three dimensions we're familiar with, which are length, width, and height. It possesses a fourth dimension, which is often conceptually depicted as time, but in terms of geometry, the fourth dimension can be understood as another direction orthogonal to the existing three dimensions. The most common example of a 4D shape is a tesseract, or hypercube. Imagine starting with a point (0D), stretching it into a line (1D), stretching the line into a square (2D), the square into a cube (3D), and then, theoretically stretching the cube at a right angle to itself to get a 4D hypercube, or tesseract. This ideal is hard to visualize because we're rooted in a 3D world, so the abstract nature of a 4D shape often requires mathematical treatment to comprehend fully.

User: Selects "Ask Why?"

ChatBot: As an Artificial Intelligence, my responses are based on available information from a large dataset rather than personal knowledge or judgment. For this specific question about 4D shapes, my answer is generated based on widely accepted definitions and explanations found in scholarly articles, textbooks, and other reliable educational resources on mathematics and physics. The concepts like geometry in higher dimensions and 4D shapes like tesseracts are commonly discussed and accepted in the fields of mathematics and theoretical physics.

Figure 5: An example demonstrating Design 3's action buttons

6 Conclusion

Human-LLM design principles should be incorporated into NLI systems, especially those embedded in learning environments. Without these practices in place, people who are not familiar with these

Table 1: Table demonstrating the effects of Design 4's settings on the model's output

Prompt: What is geometry?	Child	Expert
Concise	Geometry is a type of math that helps us understand shapes, sizes, and spaces. It's like working with puzzles or drawing pictures with rules!	Geometry is a mathematical discipline studying spatial properties and relationships of points, lines, angles, and shapes. It extends to multidimensional and non-Euclidean spaces, profoundly shaping our understanding of the physical universe.
Detailed	Geometry is a fun part of math where we learn about different shapes and how they relate to each other. You know how some shapes have corners, like a square, and some don't, like a circle? That's part of geometry! It also lets us figure out how much space something takes up, like the area of your bedroom or the volume of your juice box. So, if you like drawing, coloring, or building things, you're already using geometry without even knowing it!	Geometry is a fundamental branch of mathematics that explores the properties and relations of points, lines, surfaces, solids, and higher-dimensional entities. It encompasses various sub-disciplines, including Euclidean geometry, non-Euclidean geometry, and topology. In today's context, geometry plays a crucial role in fields like physics and computer science, especially with the advent of computational geometry and differential geometry. From a broader perspective, it provides a crucial foundation for understanding the physical universe, contributing to everything from quantum field theory to the theory of relativity.

systems may find it difficult to communicate their intentions effectively, resulting in poorly constructed queries and unhelpful responses from the LLMs. This can impede learning and overall usage of the system. In this paper, we implemented design techniques that help LLMs to better understand the subtleties in student input, enhance their ability to construct well-formed prompts, and adapt the model output to the user's proficiency level. The main drawback of the work presented in this paper is the absence of any empirical research to verify the effectiveness of the new design. In the future, it will be necessary to conduct case studies to determine whether this new interface enhances users' effectiveness with LLMs. In general, we strongly believe that Natural Language Interfaces should be designed in a way that makes them easily accessible to all users, particularly in educational settings. A well-built NLI can significantly enhance the learning experience by providing a more intuitive and supportive educational environment.

References

Atty Eleti, Jeff Harris, and Logan Kilpatrick. 2023. Function calling and other api updates. OpenAI.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2021. Deep learning for text style transfer: A survey.

Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024. An llm compiler for parallel function calling.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Venkat Krishna Srinivasan, Zhen Dong, Banghua Zhu, Brian Yu, Hanzi Mao, Damon Mosk-Aoyama, Kurt Keutzer, Jiantao Jiao, and Jian Zhang. 2023. Nexusraven: A commercially-permissive language model for function calling. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Hariharan Subramonyam, Christopher Lawrence Pondoc, Colleen Seifert, Maneesh Agrawala, and Roy Pea. 2023. Bridging the gulf of envisioning: Cognitive design challenges in llm interfaces.