Do We Need Natural Language? Exploring Restricted Language Interfaces for Complex Domains

Jesse Mu
Stanford University
Stanford, CA, USA
muj@stanford.edu

Advait Sarkar
Microsoft Research Cambridge
Cambridge, UK
advait@microsoft.com

ABSTRACT

Natural language interfaces (NLIs) that aim to understand arbitrary language are not only difficult to engineer; they can also create unrealistic expectations of the capabilities of the system, resulting in user confusion and disappointment. We use an *interactive language learning game* in a 3D blocks world to examine whether limiting a user's communication to a small set of artificial utterances is an acceptable alternative to the much harder task of accepting unrestricted language. We find that such a restricted language interface provides same or better performance on this task while improving user experience indices. This suggests that some NLIs can restrict user languages without sacrificing user experience and highlights the importance of conveying NLI limitations to users.

KEYWORDS
natural language interfaces; user experience; speech technology; chatbots

INTRODUCTION

Due to recent advances in natural language processing, machine learning, and speech recognition [8], *natural language interfaces* (NLIs) have become a major medium of human-computer interaction (HCI). As voice assistants such as Siri and Alexa master simple one-shot commands (e.g. *what's the weather?*),

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3312975

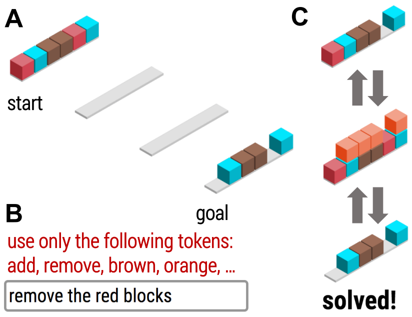
A
start
goal
C
solved!
B
use only the following tokens:
add, remove, brown, orange, ...
remove the red blocks

Figure 1: SHRDLURN. A: Game with *start* and *goal* states and 2 intermediate states. B: The player issues a language command. "use only..." message appears only for players in restricted condition. C: The player scrolls through candidate configurations until she finds the one matching the meaning of the utterance. The correct interpretation (bottom) solves the puzzle.

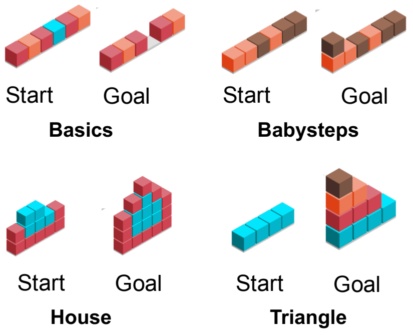

Figure 2: Example SHRDLURN levels. Earlier levels (top row) are substantially easier than later levels (bottom row), which rely on abstractions learned earlier.

the next frontier of NLIs is more complex, multi-step domains such as analyzing and visualizing data [4], text [10], and images [11], querying databases [2], and controlling robots [14]. These NLIs must learn to map ambiguous human language to precise computer actions in a collaborative dialog where the user's goal is continuously updated and expanded.

Arguably, not a single NLI today has achieved human-level abilities in these settings, as natural language understanding is a notoriously difficult problem [8]. However, little existing work in HCI examines whether such broad-coverage language interfaces for these domains are actually desirable from a user experience perspective. While NLIs are thought to have several benefits *prima facie*, such as familiarity and flexibility [7, 16], several studies have demonstrated worrying user experience issues with current-generation personal assistants due to mismatches between user expectations and system capabilities [3, 12]. These problems could be compounded in the aforementioned domains where quickly understanding the underlying logic of the system is crucial for productivity.

For complex, multi-step domains, is it possible to preserve the advantages of language interaction while better guiding user behavior (and freeing researchers from the burden of handling unrestricted language)? In this work, we explore whether users prefer true NLIs compared to a simpler, "restricted" language interface. We study an *interactive language learning game* [15] (Figure 1) where a user and a state-of-the-art natural language processing system jointly develop a language from scratch to build structures in a toy blocks world, and either (1) freely develop their own language or (2) are restricted to a small quasi-language of well-defined utterances which mirror the computer action space.

We find that restricting user languages to be artificially simple results in same or even better task performance, and even unrestricted users tend to organically prefer simpler languages and perform better as a result. Crucially, restricted languages also significantly improve user experiences in the task, reducing cognitive load and perceived effort while increasing perceived performance. These results suggest that for some well-specified domains, guiding users towards consistent, artificial languages can improve user experience while simultaneously making NLI design and engineering far easier.

RELATED WORK

Research into user experiences of voice- and natural language-based interfaces has been historically scarce [1]. Studies have assessed experiences and expectations with commercial products such as Alexa and Siri [3, 12, 13] and so are limited by the relatively simple services they currently provide.

NLIs for more complex, domain-specific tasks have often assumed various advantages of natural language interaction: they eliminate the need to learn a specialized programming language [4, 10] or advanced GUI [11] and can more succinctly convey abstract behavior [14]. Some user studies have demonstrated increased satisfaction with NLIs compared to a non-linguistic baseline [11].

Table 1: SHRDLURN logical primitives. Primitives are recursively composed into candidate actions z_i (e.g. `add(red, cyan)`, `remove(except(brown))`).

Primitive	Description
`all`	all blocks
`cyan, red, brown, orange`	colors
`except`(B)	all blocks except B
`leftmost`(B)	leftmost block of B
`rightmost`(B)	rightmost block of B
`add`(B, C)	add color C to B
`remove`(B)	remove B

- **Effort**: How hard did you have to work (mentally and physically) to accomplish your level of performance?
- **Frustration**: How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?
- **Mental Demand**: How much mental and perceptual activity was required? Was the task easy or demanding, simple or complex?
- **Performance**: How successful were you in performing the task? How satisfied were you with your performance?

Figure 3: NASA-TLX Scale Descriptions

The evidence for the benefits of NLIs in these domains compared to specialized programming languages or GUIs is strong. However, we aim to formally examine whether truly "natural" language is desirable. In doing so, we build on previous work designing effective dialogue systems [5, 9].

SHRDLURN: INTERACTIVE LANGUAGE LEARNING GAME

Our testbed is SHRDLURN [15], an interactive language learning game where a human and computer develop a language to manipulate blocks from start to goal configurations in an artificial world (Figure 1). The game is inspired by classic work on language interfaces and interaction (SHRDLU; [16]). Following Wang et al. [15], we study this game as it shares the complex and compositional nature of many real-world tasks. Additionally, the underlying logical language of the computer is hidden from the user, which mimics current NLIs where users are encouraged to *give [any] command*.[1]

Formalization

Formally, let $\mathcal{Y}$ be the set of possible configurations of blocks. In each SHRDLURN level, human and computer are presented with a *start* state $s \in \mathcal{Y}$, while only the human observes a *goal* state $t \in \mathcal{Y}$. The task of the human is to transform the *start* configuration into the *goal* configuration with a sequence of language commands $[x_1, \ldots, x_n]$.

For each utterance x_i, the computer constructs a set of possible *actions* corresponding to the meaning of the utterance, $Z = [z_1, \ldots, z_k] \subseteq \mathcal{Z}$, ranked by their plausibility, where $\mathcal{Z}$ is the set of actions in the game. It then computes the set of successor states $Y = [y_1, \ldots, y_k] \in \mathcal{Y}$, where $y_i = z_i(s)$ is the result of applying action z_i to the current state s. The human chooses the successor state y_i corresponding to the meaning of the utterance x_i. The state then updates $s = y_i$ and the computer updates its parameters with the known utterance-state pair (x_i, y_i). The level ends when $s = t$. Note that the human never observes the actions Z and instead must build a mental model of the system's capabilities, similar to existing natural language interfaces whose underlying mechanisms are hidden to the user. Figure 1 depicts the user interface for this process.

The action space $\mathcal{Z}$ is defined by a simple grammar corresponding to adding and removing blocks of color $C \in \{$`red`, `orange`, `cyan`, `brown`$\}$ to the top of a set of stacks of blocks B, where B is recursively specified with operators such as `all`, `leftmost`(B), and `except(B)` (Table 1).

The semantic parser in SHRDLURN is a simple log-linear model over logical forms given lexical features of an utterance (e.g. bigrams, trigrams). It initially knows nothing, but is designed to learn quickly via efficient gradient descent updates. Generally, the system only needs one or two uses of an utterance such as *remove the red blocks* to learn the mapping to `remove(red)`. Thus, users are generally much quicker at completing later levels of a stage than earlier levels, and consistent and efficient use of language can greatly increase task performance. There are 27 levels in the game, with

[1] Amazon Echo launch trailer; youtube.com/watch?v=FQn6aFQwBQU

later levels depending on abstractions learned in the earlier levels (examples in Figure 2). For full details, see Wang et al. [15].

METHOD

We recruited 16 participants (14 male; 2 female), all graduate students and recent college graduates from the US and UK. We evenly divided these participants into two conditions:

- In the *unrestricted* setting, users were instructed to communicate to the system using any language of their choice;
- In the *restricted* setting, user utterances were restricted to contain only combinations of the 11 tokens corresponding to the logical SHRDLURN primitives: `all`, `cyan`, `red`, `brown`, `orange`, `except`, `leftmost`, `rightmost`, `add`, `remove`, plus the connective `to`. Utterances with other tokens were rejected and users were asked to try again via the prompt in Figure 1.

Measures

For each utterance, the computer produces a list of possible interpretations ordered by likelihood, and the user must *scroll* through interpretations to find the correct state (the computer is *correct* if the user does not have to scroll at all). We thus used the *number of positions scrolled* as a performance metric: the fewer positions scrolled, the smaller the gap between the user's intended meanings and the computers' inferred interpretations. This serves as a proxy for the *surprisal* of the system [15].[2]

To measure subjective experiences in the task, we use the Raw NASA-TLX survey post-experiment [6], using the **Effort**, **Frustration**, **Mental Demand**, and **Performance** scales (Figure 3), dropping the Temporal and Physical scales as they were unrelated to the task.

RESULTS

Participants played all 27 levels of SHRDLURN, taking 15 minutes on average. Across all participants, we collected 585 utterances, averaging 36.6 ± 4.18 utterances per participant.

Unsurprisingly, the languages developed in the restricted setting were simpler, with a mean length of 3.36 ± 1.10 tokens versus 5.12 ± 1.80 ($t(497.6) = -14.3, p < 0.0001$). Each restricted participant used all 11 tokens available, compared to the average 28.8 ± 7.17 distinct tokens used by unrestricted participants (Figure 4). These simpler languages seemed to translate to same or better task performance (Figure 5): restricted participants needed an average of 7.63 ± 4.11 scrolls per utterance, compared to unrestricted users' 12.9 ± 7.98 scrolls, but the difference was not significant ($t(10.5) = -1.66, p = 0.13$).

Figure 6 depicts the distribution of NASA-TLX question responses for users in the restricted and unrestricted conditions. For many questions, significant differences were seen in user responses: restricted users reported less required effort (48.8 ± 17.5 vs. 72.5 ± 15.4; Mann-Whitney $U = 9, n_1 =$

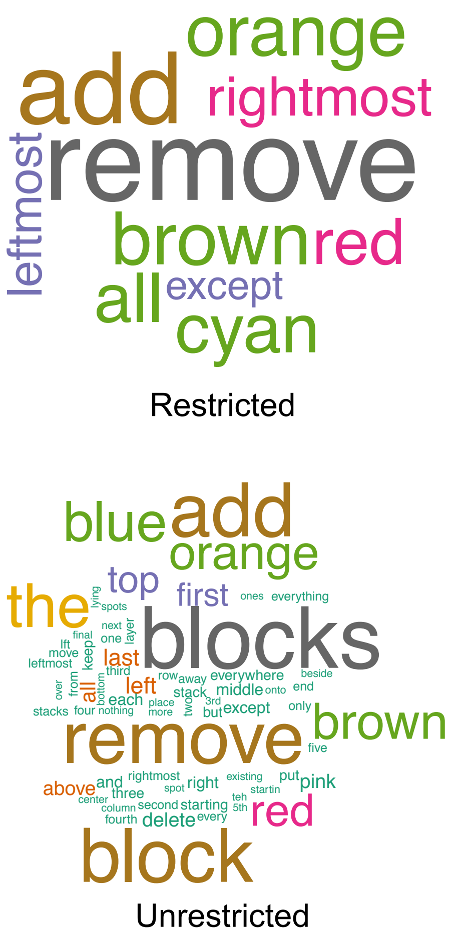

Figure 4: Unique tokens used by participants in restricted and unrestricted conditions. Size indicates frequency.

[2] Defined as the negative log likelihood of the logical forms given utterances under the model.

$n_2 = 8, p = 0.014)$[3] and *perceived* their task performance to be higher (66.9 ± 20.3 vs. 39.4 ± 16.6; $U = 57, p = 0.005$). Mental demand also appeared to be lower, though the difference was not quite significant (54.4 ± 20.8 vs. 75.6 ± 13.5; $U = 14, p = 0.06$). There was no significant difference between reported task frustration (49.4 ± 15.7 vs. 52.5 ± 19.5; $U = 29, p = 0.78$).

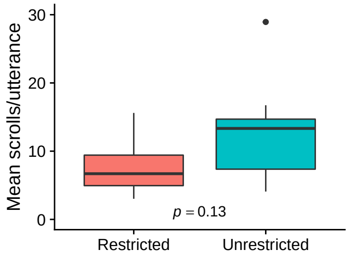

Figure 5: Mean scrolls/utterance for users in restricted and unrestricted conditions. Lower scores indicates less model surprisal and thus higher task performance.

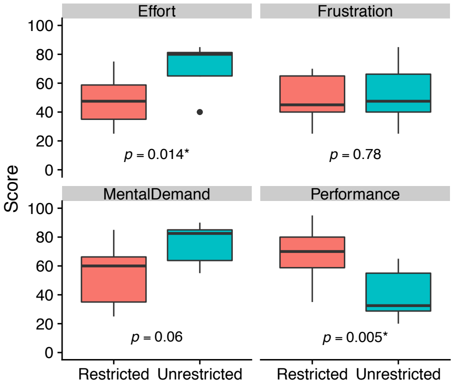

Figure 6: NASA-TLX scores for users in restricted and unrestricted conditions.

Finally, we qualitatively analyze the communication strategies (in the unrestricted setting) that lead to good task performance. Table 2 displays utterances made by the most and least successful players in the unrestricted condition, with a restricted player for reference. The best player uses consistent language and logical specifiers remarkably similar to the restricted language setting. Although the player does not use the exact tokens corresponding to the logical language, the system still comfortably learns, for instance, to map *first* to `leftmost` and *last* to `rightmost`. The low-performance player uses nonsensical utterances (*move nothing*) and multiple specification systems (e.g. *all except blue* vs. *fifth*).

DISCUSSION

We investigated users playing an interactive language learning game in a toy blocks world. By imposing restrictions on how users communicate, we found that artificial languages tend to result in same or better task performance (Figure 5) without detriment to user experience: in fact, participants reported *less* effort and higher performance in the restricted condition (Figure 6). Qualitatively, we also showed that users that organically developed simpler languages tended to perform better (Table 2).

These results are partially unsurprising, since players in the restricted condition are forced to be perfectly consistent, which improves model learning. However, we also hypothesize that a guided, consistent language helps users understand the limitations of the system and, within these constraints, infer the abstractions needed to succeed in the task.

These preliminary results come with some limitations. First, these experiments are more relevant for text-based NLIs, since text input is more amenable to restriction and shorthand than voice communication. Second, they are limited to well-specified domains with finite action spaces, unlike open-ended tasks such as unrestricted question answering.

Indeed, our intent is not to suggest that full human-level language capabilities will never be desired in future NLIs. We instead argue that given the capabilities of current NLIs, we will see diminishing returns in user experience and performance by attempting to accommodate arbitrary natural language input, especially for repetitive, compositional tasks like SHRDLURN. More generally, rather than considering only two extremes—a specialized programming language or GUI versus a human-level language understanding system—designers should consider "restricted" language interfaces which trade off full expressivity for simplicity, learnability, and consistency. We have not fully explored the range of options in this work, but as our initial results show, such systems may not only be easier to build: they may even preserve the benefits of natural language interaction while helping users better understand the system, improving user experience as result.

[3] Ties broken with the *mid-rank* method.

Table 2: Sample utterances for the best and worse unrestricted players, with a restricted player for comparison.

Player 1/8 (Mean scrolls/utterance: 4.08)
add blue blocks to blue blocks
add red block on the last orange stack
remove last red block
remove top orange blocks
remove first red block
Player 8/8 (Mean scrolls/utterance: 28.9)
move nothing
move all but blue
move all but red
remove 5th
remove first
Restricted Player (Reference)
remove brown
add brown to cyan
remove all except leftmost brown
add rightmost red
add leftmost red

ACKNOWLEDGMENTS

We thank reviewers and Alan Blackwell for valuable feedback, and Sida Wang for helpful discussions on SHRDLURN. JM is supported by the Winston Churchill Foundation of the United States.

REFERENCES

[1] Matthew P Aylett, Per Ola Kristensson, Steve Whittaker, and Yolanda Vazquez-Alvarez. 2014. None of a CHInd: relationship counselling for HCI and speech technology. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. 749–760.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1533–1544.

[3] Benjamin R Cowan, Nadia Pantidi, David Coyle, Kellie Morrissey, Peter Clarke, Sara Al-Shehri, David Earley, and Natasha Bandeira. 2017. What can I help you with?: infrequent users' experiences of intelligent personal assistants. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 43:1–43:12.

[4] Sumit Gulwani and Mark Marron. 2014. NLyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In *Proc. 2014 ACM SIGMOD International Conference on Management of Data*. 803–814.

[5] Brian Hansen, David G Novick, and Stephen Sutton. 1996. Systematic design of spoken prompts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 157–164.

[6] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. 904–908.

[7] Gary G Hendrix. 1982. Natural-language interface. *Computational Linguistics* 8, 2 (1982), 56–61.

[8] Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science* 349, 6245 (2015), 261–266.

[9] Laurent Karsenty. 2002. Shifting the design philosophy of spoken natural language dialogue: From invisible to transparent systems. *International Journal of Speech Technology* 5, 2 (2002), 147–157.

[10] Nate Kushman and Regina Barzilay. 2013. Using semantic unification to generate regular expressions from natural language. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 826–836.

[11] Gierad P Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. Pixeltone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2194.

[12] Ewa Luger and Abigail Sellen. 2016. Like having a really bad PA: the gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5286–5297.

[13] Martin Porcheron, Joel E Fischer, Stuart Reeves, and Sarah Sharples. 2018. Voice Interfaces in Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 640:1–640:12.

[14] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *AAAI Conference on Artificial Intelligence*.

[15] Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2368–2378.

[16] Terry Winograd. 1971. *Procedures as a representation for data in a computer program for understanding natural language.* Technical Report. MIT.