OSJ | OPEN SCIENCE JOURNAL

CASE REPORT

Improving Conversations with Digital Assistants through Extracting, Recommending and Verifying User Inputs

Sarah Burke[1*], Shauna Logan[2], Larissa Maksi[2]

[1]Raytheon RTX Andover, USA
[2]Raytheon RTX Tucson, USA

*Corresponding author: Sarah Burke: sarah.burke@rtx.com

Abstract:

Digital assistants, including chat bots and voice assistants, suffer from discrepancies and uncertainty in human text and speech inputs. Human dialogue is often varied, ambiguous, and inconsistent, making data entry prone to error and difficult for digital assistants to process. These difficulties are amplified in high security environments such as the Department of Defense (DoD).

This paper describes a series of input logic codifiers that form a corrective method to overcome errors and ambiguity typical of voice and text inputs. When users make a common mistake, the digital assistant can bridge the gap by recommending the most similar data that is available. The assistant measures the delta between the user's utterance and valid entries using fuzzy logic to identify the closest and next closest data that relates to the unstructured text. Furthermore, there are endless ways to denote dates, locations, etc., making it difficult for digital assistants to extract accurate and relevant data from the user's natural language. The desired data format or type is inferred using fuzzy extraction methods, such as fuzzy date extraction, to isolate the desired data format from the unstructured text. This extracted information is then verified or confirmed by the user to maintain data accuracy and avoid downstream data quality issues.

Finding and extracting pertinent information from unstructured user inputs improves and expands the use of digital assistants on any platform. By confirming data entries and providing relevant recommendations when invalid information is provided, the digital assistant enables the use of natural language and introduces a higher degree of flow into the conversation. Implementing these codifiers allows highly restricted industries, such as the DoD, to utilize digital assistants to a higher degree.

OPEN ACCESS

Citation: Burke S., Logan S., Maksi L. .(2023) Improving Conversations with Digital Assistants through Extracting, Recommending and Verifying User Inputs. Open Science Journal 8(2)

Received: 11th June 2023

Accepted: 23rd August 2023

Published: 20th September 2023

Funding: The author(s) received no specific funding for this work

Competing Interests: The authors have declared that no competing interests exists.

Keywords: Digital assistant, Fuzzy logic, Voice assistant, Data entry, Chat bot, Natural language processing, NLP

Introduction

Standard processes in the commercial industry, and specifically in the Department of Defense (DoD), are often complex and require a high degree of user interaction. Historically, the DoD has used outdated manual processes, but it is now working towards updating its processes through the DoD Software Modernization Strategy (1). Many of these processes are run on computer-based systems that involve multiple user inputs to complete the given task. Each of these inputs is required to be in a specific format. If the input is given in an unexpected format, the software will be unable to interpret the data and it will not have the information it needs to complete the task (2).

There are two primary options for collecting user input: Graphical User Interfaces (GUI) and Voice User Interfaces (VUI). GUIs are commonly used by the DoD for input because they are digital and consistent from person to person. It is easy to restrict input to the required format, which avoids unstructured data inputs and leads to fewer errors. Despite this benefit, GUIs can also be complicated to use, resulting in user frustration and time inefficiencies (3). GUIs that use touch screen devices have the added disadvantage of being difficult to use outside in adverse weather such as sun and rain. Touch screen devices also cannot be used if the user is wearing gloves or has gotten dirt / grease on their hands, which is a common occurrence for those in the DoD performing maintenance tasks. Not only does the user lose time having to take off their gloves to use the software, but the act of context switching between performing the task at hand and searching through a GUI can cost up to 40% of the user's productive time (4).

The alternative to GUIs is to use a VUI instead, which uses human speech input. VUIs enable the user to be hands free while they complete the task at hand. They do not have to worry about adverse weather or taking off their gloves to complete a digitized form. This saves the user time from context switching and from typing, which takes longer than using natural language (almost three times longer than speaking (5)). These benefits of VUIs has led to over 135 million people using voice assistants just in the United States (6).

The wide adoption of this growing technology has its disadvantages (7). Human speech is varied, ambiguous, and inconsistent, which leads to errors in the software translating speech input to the desired data input (8). This problem is even more prevalent for users with accents or speech impediments, making it difficult for industries like the DoD to implement this technology (9). Background noise such as wind, traffic, and machinery also inflate this issue (10). One study has shown that the most widely used voice assistants like Amazon's Alexa or Apple's Siri only answer questions correctly 80% of the time (11). These commercial voice assistants have much lower security requirements than the DoD, allowing them to use an immense amount of data. Noise and reverberation resulting from the adverse conditions in which a military system must operate in can be detrimental to performance. The impact is more severe when security restrictions require system reliability of recognition tasks. A speaker recognition use case is described in, "Automatic Speaker Recognition System in Adverse Conditions – Implication of Noise and Reverberation on System Performance" (12). While our paper does not describe a speaker recognition system, the implications on performance impact and security hold. With the restrictions in place for secure environments like the DoD, it would be even more difficult to implement an accurate VUI with the currently available technology. Many of the best commercially available speech-to-text (STT) engines access commercial clouds (13), which are not secure for industries like the DoD. Those engines that are lightweight and able to be used in an air-gapped environment typically do not have sufficient computing power for high accuracy speech-to-text translations. These differences in technology lead to lower performing voice assistants in highly controlled industries.

These problems can be applied to all digital assistants, which are programs that can understand natural language and answer questions or perform tasks for a user. Examples of these include voice assistants and chat bots, a program that simulates human conversations (14).

To address these problems with digital assistants, we developed an input recommender and extractor to improve data input through natural language. We focused on improving the user's experience with digital assistants by developing input logic codifiers that allow the user to converse naturally with the system.

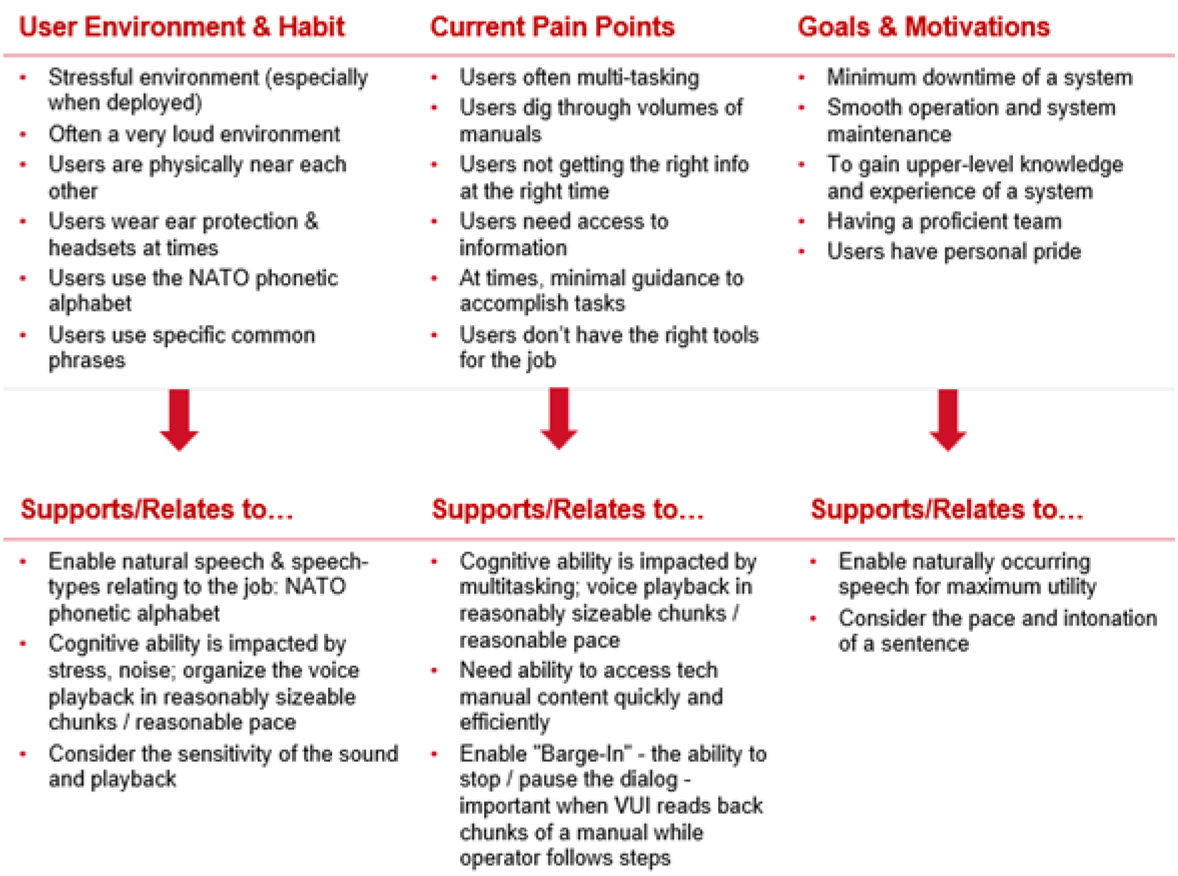

Figure 1: Observation notes after qualitative interviews with users and SMEs on how they would use speech and language in their environment

Specifically, this technology can be applied to operators working on the factory floor. Voice assistants can help workers navigate work instructions and find relevant documents seamlessly while working on the shop floor. They can also enter data verbally, streamlining data collection. With these applications, many manual tasks and associated lost productivity from context switching could be reduced or eliminated. Human-in-the-loop (HIL) errors would also be reduced due to increased access to information, leading to time savings from less factory rework. With current out-of-the-box commercial-off-the-shelf (COTS) technologies, it would be difficult to implement a high-fidelity voice assistant that would lead to these increased efficiencies. COTS technologies would not contain the domain specific language used by the DoD, specifically the terminology of our Army customer base. As described in, “Conversational AI over Military Scenarios Using Intent Detection and Response Generation” (15), it is difficult to obtain domain-specific datasets and most conversational assistants are based on commercial applications. It becomes necessary to perform entity extraction with retrieval validation. Because of these shortcomings, our team developed the logic codifiers described in this paper.

The custom approach described below includes the creation of original skills with intent files created for our user base. An intent file includes the keywords or phrases that trigger the activation of a skill. Intent detection is the technology that matches the semantics in the intent file to the user’s utterance to activate the appropriate skill. Slot filling is also included where appropriate in the intent files to allocate semantic tags for entity recognition within the user’s request. For example, if the user’s utterance includes an ID or location that information is extracted from the utterance as a slot within the intent file (16).

Approach

Design Thinking – User Experience (UX) Research

Design Thinking is a process for solving problems by prioritizing the consumer’s needs above all else. It relies on observing with empathy, how people interact with their environments, and employs an iterative, hands-on approach to creating innovative solutions (17). In order to ensure we were tailoring the digital assistant design to meet the specific needs of its users, we needed to understand who would be using it, the environment it would be used in, as well as text-to-speech (TTS) and speech-to-text (STT) considerations. TTS is a type of assistive technology that reads digital text aloud (18). STT is speech recognition software that enables the recognition and translation of spoken language into text through computational linguistics (19).

Our team conducted multiple rounds of qualitative interviews with subject matter experts (SMEs) and stakeholders to understand how the audience uses speech and language in their environment. From this data, we created four user personas representing different military roles and a list of sample speech commands and phrases, along with an explanation as to why these are valid voice commands to consider. The product owner provided a list of 11 actions matched with voice commands they wanted the voice assistant to execute. After talking with and listening to approximately 12 representatives working in these kinds of maintenance tasks in the field for the Department of Defense, we were able to match these desired actions with a more accurate match of voice commands to trigger the action. For all 11 actions we identified more than 28 voice commands that a user would use to commonly trigger these commands. We also added two more actions and commands – those being “pause” and “continue”, due to the nature of their work, the environment they work in, and how they work. The various reasons for choosing each voice command included natural speech, cognitive load limits, and cultural speech patterns (the use of the NATO Phonetic alphabet, in this case).

Additionally, we considered the persona of the voice assistant. Research shows that a female voice is preferred over a male voice, because a female chatbot voice is perceived as more human and likely to consider unique needs (20). This lead us to choose a female voice for the VUI.

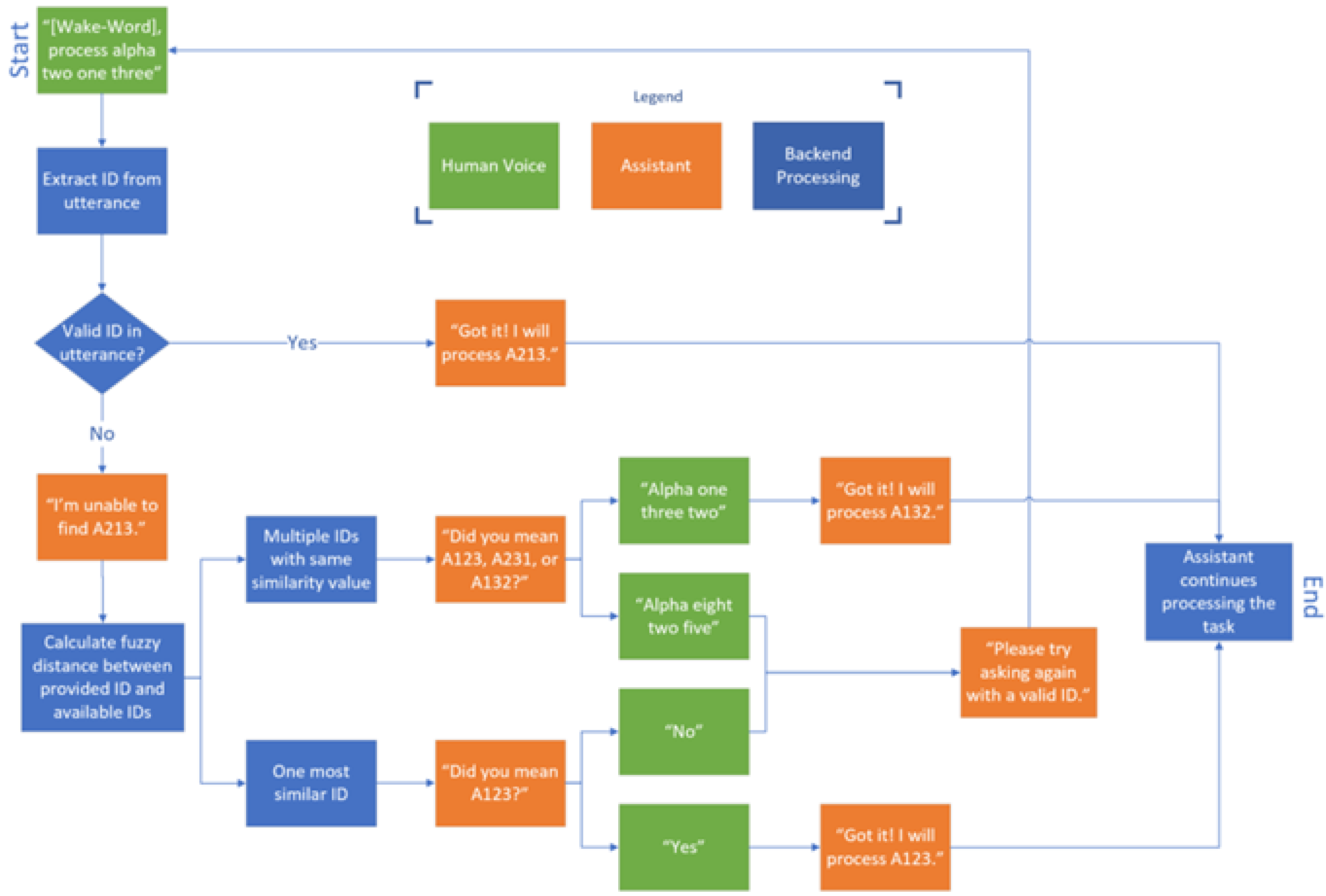

Figure 2: Example input recommender process flow map

From the qualitative interviews, we learned several things that helped to tailor the design in a way that would support the user base and enable them to use natural speech patterns they are accustomed to, as summarized in Fig. 1.

Technical approach

As a result of the User Experience research conducted, it became clear that we required a flexible system that could easily adapt to changing input parameters. This research was the basis for the development of the input logic codifiers to overcome errors and ambiguity.

The first of these input codifiers is an input recommender to keep conversations going in the event a user makes a mistake or forgets data, such as an identifier. The assistant may check a data source, such as a database, to check the validity of the provided input. In the case that the identifier is not found, the assistant can provide a recommendation of a similar identifier using fuzzy logic. Data quality is ensured by verifying all inputs exist in the data source and verifying entries with the user.

The second input codifier handles inconsistencies in data formats. Generally, a GUI restricts the data entry by enforcing the use of widgets, like a date input box. Without any restriction, there are endless ways a user may denote an input type, or not at all. The assistant attempts to extract an anticipated data type from a user's free-form input using fuzzy extraction methods (21). Error handling is used in the event a user does not provide an expected input type. For example, a location is expected, but a date is provided. When a fuzzy extraction method does detect the desired format type, it will verify with the user that the extracted information is correct. This check is performed to maintain data quality and ensure only accurate data is used to process a task.

The details of these methodologies and example workflows will be described in the next section

Details

Input recommender

Users commonly make mistakes or forget data, such as part numbers, codes, or other identification numbers. These identifiers are often required as keys to process a request or complete a task. Examples include processing a purchase requisition, looking up personnel information, or resolving an open ticket item. We present a workflow that verifies entries against a data source, makes recommendations, and confirms with users prior to preceding with a task. A generic process flow map is presented in Figure 2.

The user initiates a request by providing an identifier with the command to activate a skill or trigger an action (i.e. "Show me...[admin number]", "Tell me about...[ID]"). The assistant will extract the identifier from the user's utterance or typed command. The identifier is extracted by looking for the format that matches the data source, such as an alpha character followed by three numeric characters.

After the identifier is extracted, it will be compared against the available data in the source of record to see if there is a match. If a valid ID is provided, the assistant will repeat the identifier and proceed with the transaction. However, if a valid ID is not provided, the assistant will inform the user that she was unable to find a match and will repeat the identifier provided by the user.

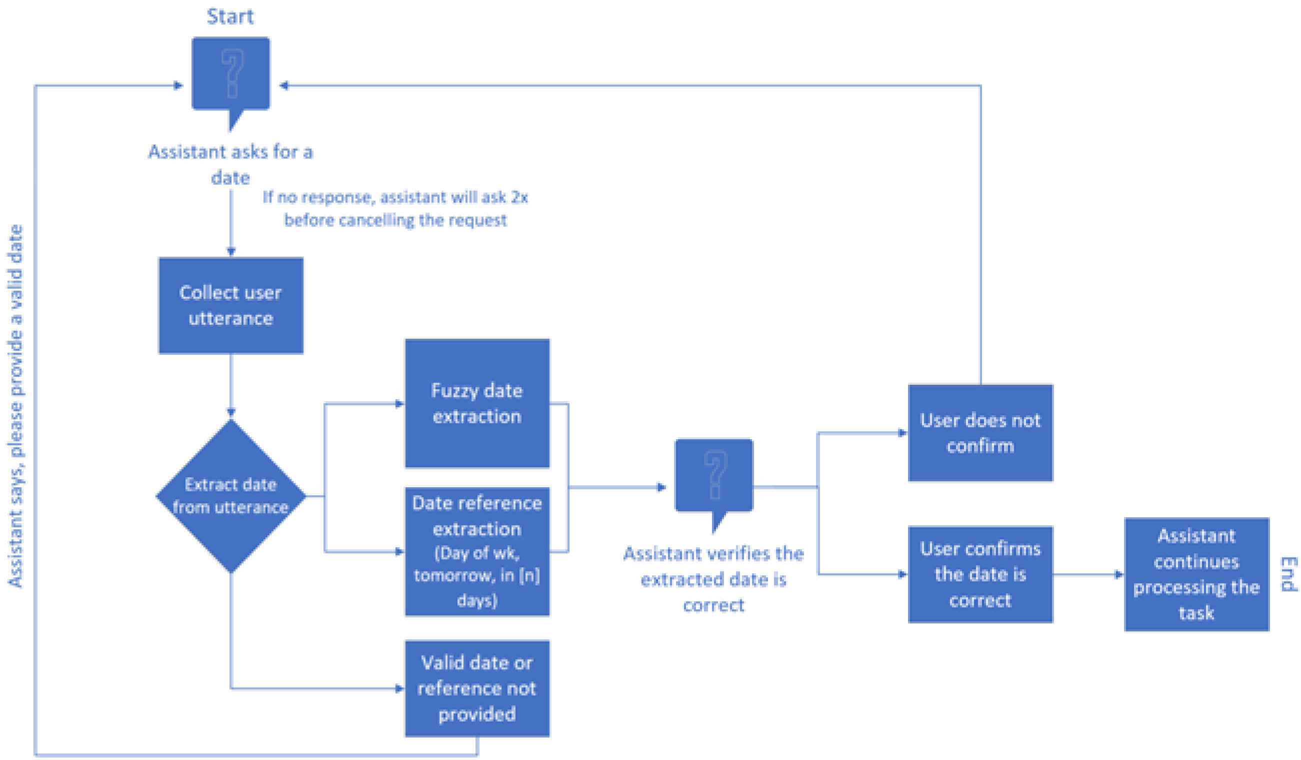

Figure 3: Date input extractor process flow map

Next, the assistant calculates a fuzzy distance metric comparing the provided identifier and the available IDs in the data source. This feature uses Jaro-Winkler (22), a character swap technique, to measure the delta of the input provided to the data currently available. Other fuzzy logic algorithms could be used in place of Jaro-Winkler, such as Levenshtein (23) or Jaccard (24). The authors selected Jaro-Winkler because it uses a prefix scale, which gives a more accurate answer when the strings have a common prefix. This correction term works well when users tend to accurately remember the first alpha numeric characters of the identifying string. The formulas for Jaro Similarity and Jaro-Winkler Similarity are the following:

$$Jaro\ Similarity = \frac{1}{3} * \left(\frac{Number\ of\ Matching\ Characters}{Length\ of\ Word\ 1} + \frac{Number\ of\ Matching\ Characters}{Length\ of\ Word\ 2} + \frac{Number\ of\ Matching\ Characters - Number\ of\ Transpositions}{Number\ of\ Matching\ Characters}\right)$$

$$Jaro - Winkler\ Similarity = (Jaro\ Similarity) + (Scaling\ Factor) * (Length\ of\ Matching\ Prefix) * (1 - Jaro\ Similarity)$$

Where the scaling factor is typically 0.1 and the length of the matching prefix is up to 4.

In the case that multiple IDs have the same similarity metric, or are very close in value, the assistant will list out those IDs and ask the user if one of them was the intended identifier. If the user responds with one of the listed IDs, the assistant repeats the identifier and proceeds with the transaction. Otherwise, the assistant will request the user try again with a valid ID.

When there is only one similar ID to the user's provided identifier, the assistant will ask the user to confirm the suggested ID and will expect a simple yes or no type of response ('yeah', 'yep', 'nope' are valid responses). If the user responds with an affirmative, the assistant repeats the identifier and proceeds with the transaction. Otherwise, the assistant will request the user try again with a valid ID.

Input extractor

Many transactions require a form be completed with specific information in a prescribed format based on the data type. With natural language there is no limit on what information is given or the format in which it will be provided. While this applies to many data types (temperatures, locations, entities) we provide an example below as it relates to dates. The workflow below walks through the collection of a date entry for processing a request (Fig. 3).

The assistant will initiate the data collection by requesting a date from the user. In the event the user does not respond to the assistant's question, she will ask twice before cancelling the request.

When a user responds to the digital assistant's question, she will attempt to extract a date from the user's response. To extract a date, the assistant first attempts a fuzzy date extraction where she looks for a date such as February 8th, 02/08, February eight, or February 8, 2022. The dateutil Python library's parser (25) is used to perform the fuzzy date search on the free-form text. In the case of a voice assistant, the free-form text is first derived from the speech-to-text or speech recognition system. Fuzzy parsing with the dateutil parser allows strings as the timestr input parameter, a string containing a date, such as "Today is January 1, 2047 at 8:21:00AM" to be parsed. It returns the date timestamp contained in the free form string.

If the assistant is unable to extract a date, she will look for a reference to a date. A date reference includes responses like next Monday, tomorrow, in five days. Mycroft's utility parser, mycroft.util.parse, (26) for datetime extraction is used to perform the date reference extraction and conversion to a date type. In regard to relative dates, the current local date time is used as the anchor date. A date timestamp is returned with the remaining leftover string separated. In the event the assistant is unable to identify a date or date reference, she will ask the user to provide a valid date and will again ask for a date.

Given the assistant can identify a date or reference, she will verify the extracted date with the user to ensure it is correct and maintain data accuracy. If the user confirms the extracted date is correct, the assistant will continue processing the request or collecting information to complete the form. However, if the user does not confirm the extracted date is correct, the assistant will once again request a date and the loop will continue until the user either doesn't respond or provides and confirms a date or reference to a date.

Results

Natural speech patterns and acoustics

Voice Command Priorities	Sample Speech / Voice Command(s)	Reasoning
• Open a document	• "Show me…[admin number]" • "Pull up [LTAMDS-20]" • "Bring up…" • "Open LTAMDS-20"	• Every system is given an admin no. This is how they identify a system • System TMs are labeled according to "manual type". Manual types are standard across the military • Natural speech • Remembering "specific numbers" or "sections" in a TM is impossible
• Go to Step / Jump to a desired step	• "Go to step One Alpha" (this means Step 1A) • "Open Work Package 3-0-0-6" • "Open LTAMDS Operations Manual Work Packet 3-0-0-6"	• Natural speech • NATO phonetic alphabet is used at times
• Repeat – Repeat the reading of the step	• "Say again" • "Repeat"	• "Say again" is common verbiage in military when a person has not heard a command Natural Speech
• Pause the voice playback	• "Pause" • "Stop"	• Natural speech • An user might not have cognitive load to remember large volume of content as it's being fed to them
• Continue the voice playback	• "Continue" • "Resume"	• Natural speech • An user is ready for the voice system to resume playback after a pause
• Close	• "Close"	• Natural speech
• Play/Show/display multimedia content	• "Show image…" • "Play Step 2 Alpha" (this is step 2A) • "Open video/image…"	• Natural speech • NATO phonetic alphabet is used at times

Figure 4: Vocabulary samples matched with digital assistant commands/skills based on UX research with users and SMEs

Based on the user experience research conducted we were able to tailor the VUI design so that its users can use their natural speech patterns when interacting with the system, including the use of the NATO phonetic alphabet. The VUI system can input any NATO phonetic phrase, translate the statement, and then reply to the user with a meaningful response in relation to their input.

Several considerations were made to train the system to understand various words and phrases that are often used for a single command or task (See Fig. 4). For example, a user may say "repeat" or "say again" to trigger the assistant to repeat her previous utterance.

The speed of the speech playback was set at a pace that was easy to follow and a wake word was trained to trigger the system even when there is background noise, so that users can access the system in their natural environment – which is often noisy. Barge-In was enabled to allow the user to pause the playback.

Based on user feedback from our UX research, input logic codifiers were created to improve the user experience and conversation flow. Through usability testing, qualitative interviews, and a soldier touchpoint demo, we were able to verify that the NATO phonetic alphabet with alpha numeric identifiers in conjunction with the implementation of the logic codifiers, has improved the VUI speech recognition and resulted in less errors due to accents or background noises. The input recommender was developed to help users with remembering specific identifiers, which are commonly numeric values, that are difficult to remember. The recommender creates a more enjoyable experience and saves the user time from having to look up information.

The data format input extractor enables the user to use their natural speech and results in improved data quality by confirming the extracted information with the user. Time savings are also realized from the user no longer needing to navigate a GUI or click multiple times to select an input, as is often the case with date widgets.

Overall, the development of the input logic codifiers in concert with the UX research conducted and a heuristic design review (27) has resulted in better data quality, improved time savings, and a better user experience.

Conclusion

Commercial digital assistants have improved data collection and streamlined manual processes. The use of voice assistants and chat bots have been widely adopted, but the technology has not yet been optimized. The technology is even more limited in restricted environments, like the Department of Defense. The Input Recommender and Input Extractor codifiers address these limitations and aim to enhance the existing STT engines to improve the digital assistant's natural language understanding. By implementing these codifiers, highly restricted industries will be able to utilize digital assistants to a higher degree, resulting in better data quality, time savings, and a better user experience.

References

1.K. H. Hicks, "Department of Defense Software Modernization," US Department of Defense, 02-Feb-2022. [Online]. Available: https://media.defense.gov/2022/Feb/03/2002932833/-1/-1/1/DEPARTMENT-OF-DEFENSE-SOFTWARE-MODERNIZATION-STRATEGY.PDF. [Accessed: 27-Jun-2022].

2.Kupzyk, K. A., & Cohen, M. Z. (2015). Data validation and other strategies for data entry. Western journal of nursing research, 37(4), 546-556.

3.Lazar, J., Jones, A., Hackley, M., & Shneiderman, B. (2006). Severity and impact of computer user frustration: A comparison of student and workplace users. Interacting with Computers, 18(2), 187-207.

4.Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part I. Basic mechanisms. Psychological review, 104(1), 3.

5.Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., & Landay, J. A. (2018). Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(4), 1-23.

6."U.S. Voice assistant users 2017-2022," Statista, 18-Mar-2022. [Online]. Available: https://www.statista.com/statistics/1029573/us-voice-assistant-users/. [Accessed: 27-Jun-2022].

7.Maedche, A., Legner, C., Benlian, A., Berger, B., Gimpel, H., Hess, T., ... & Söllner, M. (2019). AI-based digital assistants: Opportunities, threats, and research perspectives. Business & Information Systems Engineering, 61, 535-544.

8.Swerts, M., Litman, D. J., & Hirschberg, J. (2000, October). Corrections in spoken dialogue systems. In INTERSPEECH (pp. 615-618).

9.D. Pal, C. Arpnikanondt, S. Funilkul and V. Varadarajan, "User Experience with Smart Voice Assistants: The Accent Perspective," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944754.

10.Sokol, N., Chen, E. & Donmez, B (2017) "Voice-Controlled In-Vehicle Systems: Effects of Voice-Recognition Accuracy in the Presence of Background Noise". Driving Assessment Conference. 9(2017). doi: https://doi.org/10.17077/drivingassessment.1629

11.Munster, G., & Thompson, W. (2019). Annual digital assistant IQ test. Recuperado de https://loupventures. com/annual-digital-assistant-iq-test.

12.A Al-Karawi, K., H Al-Noori, A., Li, F. F., & Ritchings, T. (2015). Automatic speaker recognition system in adverse conditions—implication of noise and reverberation on system performance. International Journal of Information and Electronics Engineering, 5(6), 423-427.
13.Hasal, M., Nowaková, J., Ahmed Saghair, K., Abdulla, H., Snášel, V., & Ogiela, L. (2021). Chatbots: Security, privacy, data protection, and social aspects. Concurrency and Computation: Practice and Experience, 33(19), e6426.
14.Wellsandt, S., Hribernik, K., & Thoben, K. D. (2021). Anatomy of a digital assistant. In Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part IV (pp. 321-330). Springer International Publishing.
15.Chuang, H. M., & Cheng, D. W. (2022). Conversational AI over Military Scenarios Using Intent Detection and Response Generation. Applied Sciences, 12(5), 2494.
16.He, T., Xu, X., Wu, Y., Wang, H., & Chen, J. (2021). Multitask learning with knowledge base for joint intent detection and slot filling. Applied Sciences, 11(11), 4887.
17.Razzouk, R., & Shute, V. (2012). What is design thinking and why is it important?. Review of educational research, 82(3), 330-348.
18.Klatt, D. H. (1987). Review of text - to - speech conversion for English. The Journal of the Acoustical Society of America, 82(3), 737-793.
19.Trivedi, A., Pant, N., Shah, P., Sonik, S., & Agrawal, S. (2018). Speech to text and text to speech recognition systems-Areview. IOSR J. Comput. Eng, 20(2), 36-43.
20.Borau, S., Otterbring, T., Laporte, S., & Fosso Wamba, S. (2021). The most human bot: Female gendering increases humanness perceptions of bots and acceptance of AI. Psychology & Marketing, 38(7), 1052-1068.
21.Pereda, R., & Taghva, K. (2011, April). Fuzzy information extraction on OCR text. In 2011 Eighth International Conference on Information Technology: New Generations (pp. 543-546). IEEE.
22.Wang, Y., Qin, J., & Wang, W. (2017, October). Efficient approximate entity matching using jaro-winkler distance. In Web Information Systems Engineering–WISE 2017: 18th International Conference, Puschino, Russia, October 7-11, 2017, Proceedings, Part I (pp. 231-239). Cham: Springer International Publishing.
23.Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).
24.Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. Sci. Nat., 44, 223-270.
25.Read the Docs. "dateutil". Built with Sphinx using a theme provided by Read the Docs. © Copyright 2019, dateutil Revision 6b035517. Available: https://dateutil.readthedocs.io/en/stable/parser.html.
26.Read the Docs. "mycroft.util.parse". Built with Sphinx using a theme provided by Read the Docs. © Copyright 2017, Mycroft AI Inc. Revision a909fc8f. Available: https://mycroft-core.readthedocs.io/en/latest/source/mycroft.util.parse.html.
27.Wei, Z., & Landay, J. A. (2018). Evaluating speech-based smart devices using new usability heuristics. IEEE Pervasive Computing, 17(2), 84-96.