ParseJargon: Personalized Real-time Jargon Support in Online Meetings

Yifan Song[1]*, **Wing Yee Au**[2], **Hon Yung Wong**[2], **Brian P. Bailey**[1], **Tal August**[1]
[1]University of Illinois Urbana-Champaign
[2]Fujitsu Research of America
{yifan33,bpbailey,taugust}@illinois.edu
{wau,awong}@fujitsu.com

Abstract

Effective interdisciplinary communication is frequently hindered by domain-specific terms. These terms, or jargon, are dependent on a listener's background, and rarely do listeners seek explanations due to distraction and social concerns. To address these concerns, we built ParseJargon, an interactive LLM-powered system providing real-time personalized jargon support tailored to users' individual backgrounds in online meetings. We first evaluated the effectiveness of personalization in a controlled setting with human participants. By comparing ParseJargon against baseline (no support) and general-purpose (non-personalized) conditions, we found that ParseJargon provided more precise jargon identification, and enhanced participants' comprehension, engagement, and appreciation of colleagues' work. We then evaluated the potential for using ParseJargon in real-time meetings through a latency test.

1 Introduction

In workplaces, professionals frequently face challenges to convey specialized knowledge to colleagues from other disciplines (Nickerson, 1999; Jeffres et al., 2011; Keelawat, 2023; Weirup and Taylor, 2024). For instance, a machine learning engineer might struggle to communicate concepts like "embedding" to a compliance officer concerned with data privacy, while healthcare professionals might face challenges describing "quasi-experimental designs" to policymakers without medical expertise. Such gaps in communication caused by domain-specific jargon limit interdisciplinary innovation and collective problem-solving (Daniel et al., 2022; Fiset et al., 2024).

Recent advances in speech-to-text technologies and large language models (LLMs) have the potential to overcome these limitations with automated, real-time jargon support. Prior research has explored computational techniques for jargon identification and explanation (Neumann et al., 2019; August et al., 2022; Huang et al., 2022; Lucy et al., 2023), and developed augmented interfaces that enhance comprehension during meetings through interactive transcripts or captions (Chandrasegaran et al., 2019; Li et al., 2021; Chen et al., 2023; Liu et al., 2023). However, existing systems typically neglect two critical aspects for effective knowledge support in meetings: real-time support and personalization. Most prior systems target only static text content (Abekawa and Aizawa, 2016; Head et al., 2021; August et al., 2023). While more recent research explores the design space of jargon support over video content, their system fails to consider user-specific background knowledge by providing uniform assistance to all users (Liu et al., 2025). This can overwhelm users with irrelevant or excessive information, especially in real-time settings, by reducing trust and user engagement (Chen et al., 2023; Aghahoseini et al., 2024).

To address these gaps, we introduce ParseJargon[1][2], a personalized jargon support system for real-time online meetings. To isolate and evaluate the performance of our personalization approach, we conducted a controlled within-subjects experiment with 7 participants who watched each other's presentation with three conditions (7×6=42 participant-presentation pairs, 14 per condition): (I) a baseline without jargon support, (II) a generic support (i.e., defining the same terms for all participants), and (III) a personalized support provided by ParseJargon. Results showed that while generic jargon support improved comprehension compared to the baseline, it negatively affected engagement by overwhelming participants with excessive jar-

*Work partially done during an internship at Fujitsu Research of America

[1]To download and install ParseJargon: https://github.com/yifansong98/ParseJargon

[2]ParseJargon stands for Personalized Assistant for Real-time Support in Explaining Jargon

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 615–625

gon explanations. In contrast, personalized support significantly improved comprehension, perceived value of peer's work, and maintained participants' engagement by more accurately predicting relevant jargon based on individuals' backgrounds. To examine ParseJargon's real-time capability, we conducted a latency test for the system using data collected in the controlled experiment, and also discussed usability constraints for a real-world deployment, such as the reliance on the quality of speech-to-text services.

In summary, this demo paper makes the following contributions:

- ParseJargon, a personalized jargon support system for real-time online meetings, powered by LLMs in identifying jargon tailored to each audience's background.
- Evidence from a controlled evaluation that personalized jargon support—even with minimal personalization based on a single-sentence profile—significantly enhances comprehension and sustains engagement.

2 Related Work

2.1 Jargon Support Technologies

Advancements in language technologies have enabled computational support for identifying and explaining jargon. A core task in this space is complex word identification with early benchmarks introduced by Shardlow (2013). Recent methods have applied LLMs to measure jargon complexity (Lucy et al., 2023) and adapted identification models to specialized domains such as biomedical research (Guo et al., 2021) or specific jargon usage like acronyms (Pouran Ben Veyseh et al., 2021). Jargon explanation has also been studied from the perspective of definition extraction (Veyseh et al., 2020), definition generation (August et al., 2022), or hybrid approaches (Huang et al., 2022). Closely related to jargon explanation, the task of text simplification transforms complex content into simpler language (Martin et al., 2020; Van et al., 2020).

Building on these techniques, researchers have designed interactive systems for jargon support, especially within asynchronous reading interfaces (Lo et al., 2024; Fok et al., 2024). For example, ScholarPhi (Head et al., 2021) provides an automatically generated glossary for important scientific terms, while Paper Plain (August et al., 2023) offers in-situ definitions of unfamiliar terms and plain language summaries. Other recent work augments medical progress notes (Kambhamettu et al., 2024) or explores how user-generated analogies can support jargon understanding during scientific reading (Bao et al., 2025).

Closely related to our work, Liu et al. (2025) explored the design space of real-time LLM-based knowledge assistance during technical videos using a design probe, StopGap. Their findings offer valuable insights into user preferences and interface design in knowledge support when watching videos, including a desire for personalization. We build on this work by implementing a prototype system for jargon support in online meetings with personalization and real-time capabilities.

2.2 Personalization in Jargon Support

Personalization plays a critical role in tailoring jargon support to users' prior knowledge. Early work adapted complex word identification and lexical simplification models to individual users, substituting unfamiliar terms based on personal vocabulary profiles (Lee and Yeung, 2018). Subsequent efforts demonstrated that modeling word complexity at the individual level significantly improved performance (Gooding and Tragut, 2022) and introduced approaches for generating personalized descriptions of scientific concepts (Murthy et al., 2021). More recent research showed that LLMs can serve as a baseline for personalized scientific jargon identification for researchers reading interdisciplinary articles (Guo et al., 2024).

Beyond algorithmic personalization, researchers have investigated how users perceive and interact with personalized language systems. For example, researchers have designed interfaces that adapt scientific information to users' expertise using rule-based templates (Oh et al., 2020). A recent study investigated the effects of adaptive plain language on diverse audiences and offered insights into using LLMs to generate summaries tailored to different levels of expertise (August et al., 2024). Other work has shown that even perceived personalization, such as user-controlled filtering, can influence people's trust, satisfaction, and comprehension of explanations (Calisto et al., 2025).

These studies provide the foundation for our approach, which incorporates audience background to deliver real-time personalized jargon support. To our knowledge, our work is novel in evaluating personalized jargon support in live meetings.

Table 1: Example filtering from generic to personalized glossary, showing tailored selection by audience background, **X** indicates that the term remains in the personalized glossary.

Glossary (generic audience)	For Machine Learning Engineer	For Earth Science Researcher
Benchmarking		
Foundation Models	X	X
Remote Sensing	X	
Pre-training		X
Satellite Data	X	
Self-supervised Learning		X

3 System Design

3.1 Example Usage Scenario

Consider a scenario in which a researcher presents a project involving deep learning applications in earth science to a business team for product development (Figure 1). The audience lacks expertise in both machine learning and earth science, making it challenging for them to follow key technical terms such as "segmentation" or "remote sensing", despite the speaker's effort to explain these terms briefly. The business team members are reluctant to interrupt the speaker or independently search for definitions, fearing social discomfort and potential distraction to miss other important points, leading to persistent confusion and potentially undervaluing the presented research.

With ParseJargon, the business team has real-time access to automatically generated explanations of unfamiliar jargon directly within their meeting interface. As the speaker presents, terms like "segmentation" and "remote sensing" are identified as jargon based on each audience member's specific background and appear in a glossary sidebar next to the main meeting window, offering concise and accessible definitions. Audience members no longer need to actively search for terms or hesitate about interrupting the flow; instead, they seamlessly access essential explanations.

With the same presentation, two other listeners: a software engineer with some ML background but limited earth science knowledge, and a senior earth science researcher with limited AI experience, have more knowledge about the project compared to the business team. However, they still need some jargon support from the generic glossary (used for business team) to complement their expertise as shown in Table 1. Personalized jargon support reduced unnecessary terms and allowed each listener to focus on information most relevant to them.

3.2 System Architecture

Our system architecture (Figure 2) consists of two main components: an LLM-powered backend for real-time jargon identification, explanation, and personalization, and a user interface that displays jargon definitions within the meeting environment.

3.2.1 Backend Technology

The backend leverages the OpenAI GPT model to perform three interconnected tasks: *jargon identification*, *jargon explanation*, and *personalization*. These tasks are executed through prompting techniques; prompts and parameters are provided in the Appendix.

- **Jargon Identification & Explanation:** ParseJargon first fetches the live transcription generated by the service provided in online meeting platforms. Upon receiving the transcription, our backend identifies potential jargon and generates concise plain-language definitions for each term. This process uses a single combined prompt, analyzing each sentence of the meeting transcript sequentially. Each identified jargon term is defined only once throughout the meeting.

- **Personalized Filtering:** To tailor jargon support to individual audience members' expertise, the system applies a second filtering step. Using a separate prompt, the system takes a user profile and removes identified terms that the user likely already knows. We test with a simple profile that invites the listener to enter a one-sentence description of their education background and/or job role (e.g., "*I am a Physics PhD, working as a research intern in the Quantum Computing team*").

We intentionally adopt a two-stage design of first identifying and explaining, then filtering based on a minimal one-sentence profile to study whether

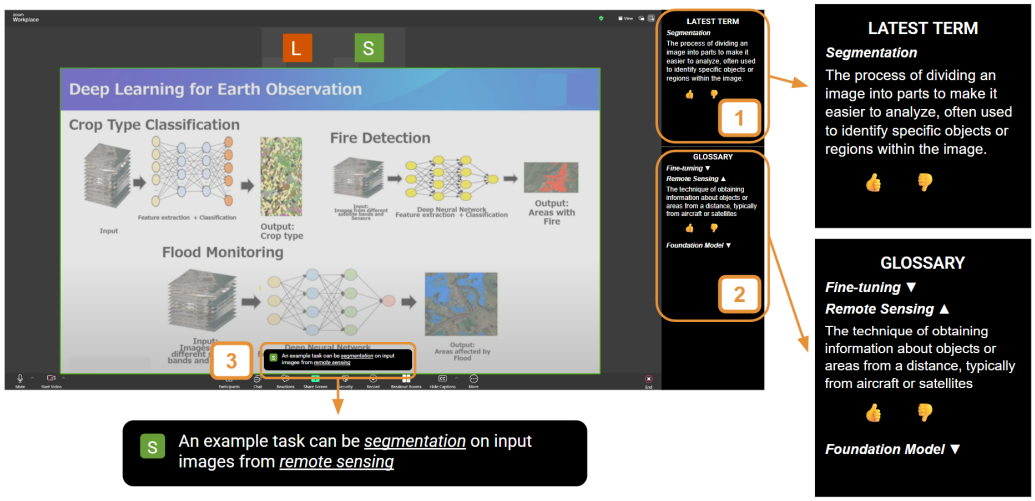


Figure 1: In an online meeting, the Speaker is presenting a project about deep learning in earth science and screen sharing the slides. The Listener uses ParseJargon with three interface components: 1) the latest jargon definition in concise plain language; 2) glossary for all jargon terms appeared in the meeting for revisiting; 3) real-time caption with highlighted jargon terms.

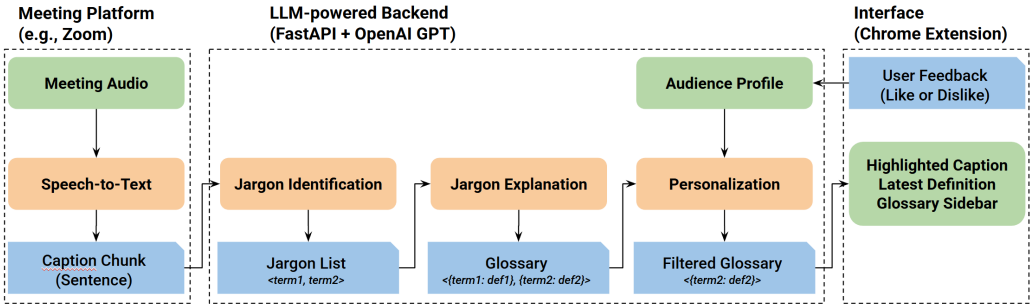


Figure 2: Workflow pipeline of ParseJargon, from speech input to personalized glossary output.

even simple personalization can make a meaningful impact in jargon support. The two-stage approach also allows us to isolate personalization from jargon identification with explanation.

3.2.2 Interface Design

The user interface of ParseJargon (Figure 1) integrates into standard online meeting platforms (currently supports only Zoom web version). The primary components include:

- **Real-time captions with highlighted terms:** Live captions are generated by the meeting platform's transcription service. ParseJargon highlights jargon terms for easy recognition.
- **Latest term definition:** The definition for the latest jargon term appears in real time, enabling quick glanceable explanations. Users can provide feedback by indicating their preference for each identified term ("like" or "dislike"). The preference list is then added to the system backend via the personalization filtering prompt to further improve personalization accuracy during the meeting.
- **Persistent glossary sidebar:** All identified jargon terms accumulate in a persistent glossary list, allowing listeners to revisit terms and definitions at any point during the meeting.

Each identified term is displayed in the sidebar until the next jargon term is identified with a minimum display of 7 seconds, hard-coded based on average reading speed (Brysbaert, 2019) and internal pilot testing. If a new term is identified before the 7 second minimum of an existing definition, both will appear in the glossary list and the first term will be replaced after 7 seconds, a mechanism informed by prior research (Liu et al., 2025).

3.3 Implementation Details

ParseJargon is implemented as a Chrome browser extension integrated into web version of Zoom.

The frontend interface is developed with `React.js`. The backend server, developed using `Python FastAPI`, manages calls to OpenAI API. In the following controlled experiment and latency analysis, we tested with `gpt-4o` model with the server deployed on `Heroku` connected to a `PostgreSQL` database. The system also supports other GPT-family models and can be deployed locally with JSON data storage.

4 Controlled Experiment

To evaluate how ParseJargon and its personalization approach affects online meeting experience, we conducted a controlled experiment with the hypothesis that personalized jargon support provides more precise jargon identification and is more effective than generic support in improving comprehension, engagement, and perceived value of others' work during online meetings.

Our system depends on live captions provided by online meeting platforms' speech-to-text engines (e.g., Otter.ai for Zoom, Azure Speech-to-Text for Teams). Because these transcriptions are not always reliable (Picovoice, 2023), we used pre-recorded videos with manually verified transcriptions in this controlled experiment. This allowed us to isolate and evaluate the system's backend, especially personalized filtering, as a proof-of-concept without interference from transcription inaccuracies or latency issues.

4.1 Methods

We recruited seven participants from a technology company who worked on different projects from different teams. This selection was aimed for minimal prior knowledge about each other's projects. Participant profiles, including education and job role, are included in Appendix Table 5. Each participant prepared a 10-minute presentation with slides about their ongoing project. Presentations were recorded, and transcripts were manually verified for correctness.

We created three experimental conditions using the recorded and transcribed presentations:

- **Generic Support:** Recordings were processed without the personalized filter. (Appendix Figure 3a)
- **Personalized Support:** Recordings were processed by the complete pipeline. (Appendix Figure 3b)
- **No-support Baseline:** Original recordings with no jargon support.

We employed a within-subject design where each participant viewed all presentations from the other six participants. Participants watched recordings individually in two sessions (∼45 minutes each), viewing three presentations per session (one per condition). Sessions were spaced at least one day apart to minimize fatigue. Conditions were counterbalanced across participants and order positions, which yielded 14 unique viewing experiences per condition and 42 participant-presentation pairs.

We evaluated effectiveness with a mix of self-report measures and jargon identification precision:

Self-reported measures After each presentation, participants completed a short survey with the following 5-point Likert-style items (1 = not at all, 5 = very): (I) Comprehension, (II) Engagement, and (III) Perceived Value; detailed survey questions in Appendix B.2.

Jargon identification precision After watching the presentation (generic & personalized condition only), participants reviewed all identified jargon terms and labeled each term as *helpful* or *not helpful*. We treated these as gold-standard labels and computed the jargon identification precision as the proportion labeled helpful.

We also asked participants to note down terms they still didn't understand while watching the presentations. Because participants varied in whether note-taking felt natural or distracting, and because these notes mixed several cases: missed terms, insufficient definitions, and terms shown disappeared too quickly, these notes were therefore not a good representation in measuring recall.

4.2 Findings

Our controlled experiment results support our hypothesis, showing that the personalized jargon support was significantly more precise, and improved participants' comprehension and perceived value of the presented work. Tables 2 report means and standard deviations for self-report measures.

Personalized support improved self-reported comprehension and perceived value while avoiding information overload. As shown in Table 2, both the generic and personalized conditions increased self-reported comprehension compared to the baseline, with the personalized condition showing greater improvement. While both experimental

Table 2: Self-reported measures by condition (mean ± SD, 5-point Likert scale). Statistical significance (* vs. Baseline; † vs. Generic) indicates Holm–Bonferroni corrected $p<.05$ using Wilcoxon signed-rank tests (N=14 per condition). More detail in Appendix B.3.

Condition	Comp.	Eng.	Val.
Baseline	3.07±0.62	3.93±0.83	3.57±0.65
Generic	3.79±0.70*	3.64±1.01	3.93±0.62
Personalized	4.29±0.61*†	4.29±0.73	4.43±0.51*†

conditions improved participants' perceived value of others' presented work, only the personalized condition showed significance in rating improvement to the baseline condition. However, participants reported lower engagement in the generic condition than baseline, whereas personalized support maintained engagement with a higher score than both baseline and generic support. Follow-up interviews suggested participants felt overwhelmed by the excessive number of definitions (6 participants). Some participants even described this as *"annoying"* (P4) or even *"offensive... (because) the system treats me like I know nothing"* (P3).

Table 3: Average precision (number of helpful jargon terms over total terms identified) per participant-presentation.

Condition	Helpful / Total Terms	Precision
Generic	10.29 / 22.57	47.03%
Personalized	7.64 / 9.71	77.51%

Personalized jargon support identifies fewer terms and is significantly more precise than generic support. Table 3 suggested that the personalized glossary identified on average fewer terms (9.71 vs. 22.57), increasing jargon identification precision from 47.03% to 77.51%. Showing fewer but more relevant entries helps limit on-screen distractions during presentation flow, helping participants stay with the speaker rather than triaging definitions. This increased precision in the personalized condition was associated with higher comprehension and perceived value compared to the generic support.

5 Latency Analysis

5.1 Methods

We measure ParseJargon's backend latency as an indicator of real-time capability using the transcripts from our controlled experiment. We report *pipeline latency per caption chunk*: the elapsed time from when the transcription service prepares a caption chunk to be displayed on screen to when the system's response for that chunk is displayed. For the generic condition, this includes jargon identification and explanation. For the personalized condition, this also includes the personalization filter when at least one jargon term is identified. We report medians (50th percentile) and tails (95th percentiles), as the tail latency most strongly affects perceived responsiveness in interactive services (Delimitrou and Kozyrakis, 2018). To capture the timeliness of jargon presentation, we compare latency with the length of time that caption chunks were displayed on screen (*chunk duration*). We did not measure end-to-end UI latency (from meeting-platform captioning to on-screen glossary rendering), which depends on device/network factors beyond our control.

There were in total 787 caption chunks across seven recordings. In the personalized condition, each transcript was measured twice with two participants' profiles (each video was watched by two participants in their unique personalized condition), yielding 1574 chunks. As the latency may change depending on API traffic and internet speed, we ran the test twice at different days and times under the same network environment (800 Mbps Wi-Fi) and report the metrics based on the two runs (787×2=1574 chunks for generic condition, 1574×2=3148 chunks for personalized condition).

5.2 Findings

Table 4: Pipeline latency (seconds) per caption chunk. Median and tail (95 percentile) for all chunks.

Condition	Subset	N	Median	Tail
Generic	All chunks	1574	0.32	1.36
	No jargon	1350	0.31	0.53
	≥1 terms	224	1.11	2.25
Personalized	All chunks	3148	0.42	1.73
	No jargon	2980	0.41	1.35
	≥1 terms	168	1.60	3.45

As shown in Table 4, the median and tail of backend latencies were 0.32/1.36 s (generic) and 0.42/1.73 s (personalized). While personalization introduces a modestly higher latency, these values remained well below the *chunk duration* range with a median of 4.15 s and a tail of 6.62 s, suggesting that definitions arrived well before the captions con-

taining referenced terms disappeared. As expected, latencies are higher on chunks that emit terms due to definition generation and filtering (generic: median 1.11 s, tail 2.25 s; personalized: median 1.60 s, tail 3.45 s). We also observed that the personalized condition produced a smaller fraction of chunks containing terms (5.34%) than the generic condition (14.23%), consistent with the controlled experiment findings of fewer, more precise identified terms with reduced information load.

6 Limitations and Future Work

First, our study involved seven participants from a single technology company, which limits generalizability across organizations, disciplines, and expertise distributions. Future studies should evaluate ParseJargon with larger and more diverse groups, especially in meetings with stronger cross-domain knowledge gaps.

Second, our current personalization uses a one-sentence profile and treats jargon selection as a binary filtering task. In practice, terms differ in conceptual difficulty and may have different meanings across domains. Future systems should use meeting context, domain cues, and user feedback to improve personalization accuracy and adapt explanation depth. For example, a foundational term may need a brief definition, while a more abstract concept may require an example or analogy.

Finally, the current prototype is implemented as a Chrome extension for Zoom Web and depends on external speech-to-text and LLM services. Broader deployment will require integrations with native meeting clients or platform APIs, as well as robustness to caption errors. We did not systematically benchmark prompt variants, different models, or longer/richer profiles. Future work should evaluate other model families (e.g., Gemini or DeepSeek), small/local models, and profile representations. Moreover, our system processes each caption chunk individually without evaluating gating strategies (e.g., user-triggered or signal-based invocation) or batching/caching that could reduce computational cost and latency.

7 Conclusion

We introduced ParseJargon, a system toward real-time, personalized jargon support in online meetings. We provided empirical evidence that even simple personalization (via a one-sentence audience profile and an identify–then–filter pipeline) significantly improved meeting experience. Looking ahead, while we used `gpt-4o` in our controlled experiment and latency analysis, newer models with more advanced performance (e.g., `gpt-5.5`) or higher speed (e.g., `gpt-realtime`) can be tested. Future systems can also integrate richer profiling, such as accumulated usage history or background inference, thereby enhancing precision. Taken together, we position ParseJargon as a practical path toward personalized language support that helps diverse audience follow—and appreciate—work across domains.

Acknowledgments

We thank our colleagues in Fujitsu's Converging Technologies Lab for their constructive feedback. We also thank our study participants who generously shared their time for this research, and the anonymous reviewers for the comments.

References

Takeshi Abekawa and Akiko Aizawa. 2016. SideNoter: Scholarly Paper Browsing System based on PDF Restructuring and Text Annotation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 136–140, Osaka, Japan. The COLING 2016 Organizing Committee.

Pouya Aghahoseini, Millan David, and Andrea Bunt. 2024. Investigating the Role of Real-Time Chat Summaries in Supporting Live Streamers. In *Proceedings of the 50th Graphics Interface Conference*, GI '24, pages 1–12, New York, NY, USA. Association for Computing Machinery.

M. Aickin and H. Gensler. 1996. Adjusting for multiple testing when reporting research results: the Bonferroni vs Holm methods. *American Journal of Public Health*, 86(5):726.

Tal August, Kyle Lo, Noah A. Smith, and Katharina Reinecke. 2024. Know your audience: The benefits and pitfalls of generating plain language summaries beyond the "general" audience. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Tal August, Katharina Reinecke, and Noah A. Smith. 2022. Generating scientific definitions with controllable complexity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317, Dublin, Ireland. Association for Computational Linguistics.

Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A. Hearst, Andrew Head, and Kyle Lo. 2023. Paper

Plain: Making Medical Research Papers Approachable to Healthcare Consumers with Natural Language Processing. *ACM Trans. Comput.-Hum. Interact.*, 30(5):74:1–74:38.

Calvin Bao, Yow-Ting Shiue, Marine Carpuat, and Joel Chan. 2025. Words as bridges: Exploring computational support for cross-disciplinary translation work. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 1598–1623, New York, NY, USA. Association for Computing Machinery.

Marc Brysbaert. 2019. How many words do we read per minute? a review and meta-analysis of reading rate. *Journal of Memory and Language*, 109:104047.

Francisco Maria Calisto, João Maria Abrantes, Carlos Santiago, Nuno J. Nunes, and Jacinto C. Nascimento. 2025. Personalized explanations for clinician-ai interaction in breast imaging diagnosis by adapting communication to expertise levels. *International Journal of Human-Computer Studies*, 197:103444.

Senthil Chandrasegaran, Chris Bryan, Hidekazu Shidara, Tung-Yen Chuang, and Kwan-Liu Ma. 2019. TalkTraces: Real-Time Capture and Visualization of Verbal Content in Meetings. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–14, New York, NY, USA. Association for Computing Machinery.

Xinyue Chen, Shuo Li, Shipeng Liu, Robin Fowler, and Xu Wang. 2023. MeetScript: Designing Transcript-based Interactions to Support Active Participation in Group Video Meetings. *Proc. ACM Hum.-Comput. Interact.*, 7(CSCW2):347:1–347:32.

Kristy L. Daniel, Myra McConnell, Anita Schuchardt, and Melanie E. Peffer. 2022. Challenges facing interdisciplinary researchers: Findings from a professional development workshop. *PLOS ONE*, 17(4):e0267234. Publisher: Public Library of Science.

Christina Delimitrou and Christos Kozyrakis. 2018. Amdahl's law for tail latency. *Commun. ACM*, 61(8):65–72.

John Fiset, Devasheesh P. Bhave, and Nilotpal Jha. 2024. The Effects of Language-Related Misunderstanding at Work. *Journal of Management*, 50(1):347–379. Publisher: SAGE Publications Inc.

Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. 2024. Qlarify: Recursively expandable abstracts for dynamic information retrieval over scientific papers. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA. Association for Computing Machinery.

Sian Gooding and Manuel Tragut. 2022. One Size Does Not Fit All: The Case for Personalised Word Complexity Models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 353–365, Seattle, United States. Association for Computational Linguistics.

Yue Guo, Joseph Chee Chang, Maria Antoniak, Erin Bransom, Trevor Cohen, Lucy Wang, and Tal August. 2024. Personalized Jargon Identification for Enhanced Interdisciplinary Communication. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4535–4550, Mexico City, Mexico. Association for Computational Linguistics.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated Lay Language Summarization of Biomedical Scientific Reviews. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):160–168. Number: 1.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2021. Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–18, New York, NY, USA. Association for Computing Machinery.

Jie Huang, Hanyin Shao, Kevin Chen-Chuan Chang, Jinjun Xiong, and Wen-mei Hwu. 2022. Understanding Jargon: Combining Extraction and Generation for Definition Modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4004, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Leo Jeffres, David Atkin, and Hanlong Fu. 2011. Knowledge and the knowledge gap: Time to reconceptualize the "content". *Open Communication Journal*, 5.

Hita Kambhamettu, Danaë Metaxa, Kevin Johnson, and Andrew Head. 2024. Explainable notes: Examining how to unlock meaning in medical notes with interactivity and artificial intelligence. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Panayu Keelawat. 2023. NBGuru: Generating Explorable Data Science Flowcharts to Facilitate Asynchronous Communication in Interdisciplinary Data Science Teams. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '23 Companion, pages 6–11, New York, NY, USA. Association for Computing Machinery.

John Lee and Chak Yan Yeung. 2018. Personalizing Lexical Simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 224–232, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Daniel Li, Thomas Chen, Albert Tung, and Lydia B Chilton. 2021. Hierarchical summarization for long-form spoken dialog. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 582–597, New York, NY, USA. Association for Computing Machinery.

Xingyu 'Bruce' Liu, Vladimir Kirilyuk, Xiuxiu Yuan, Peggy Chi, Alex Olwal, Xiang 'Anthony' Chen, and Ruofei Du. 2023. Experiencing Visual Captions: Augmented Communication with Real-time Visuals using Large Language Models. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23 Adjunct, pages 1–4, New York, NY, USA. Association for Computing Machinery.

Yuhan Liu, Aadit Shah, Jordan Ackerman, and Manaswi Saha. 2025. Exploring the design space of real-time llm knowledge support systems: A case study of jargon explanations. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, and 36 others. 2024. The semantic reader project. *Commun. ACM*, 67(10):50–61.

Li Lucy, Jesse Dodge, David Bamman, and Katherine Keith. 2023. Words as Gatekeepers: Measuring Discipline-specific Terms and Meanings in Scholarly Publications. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6929–6947, Toronto, Canada. Association for Computational Linguistics.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Constantin Mircioiu and Jeffrey Atkinson. 2017. A Comparison of Parametric and Non-Parametric Methods Applied to a Likert Scale. *Pharmacy: Journal of Pharmacy, Education and Practice*, 5(2):26.

Sonia K. Murthy, Daniel King, Tom Hope, Daniel S. Weld, and Doug Downey. 2021. Towards personalized descriptions of scientific concepts.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Raymond Nickerson. 1999. How we know—and sometimes misjudge—what others know: Imputing one's own knowledge to others. *Psychological Bulletin*, 125:737–759.

Changhoon Oh, Jinhan Choi, Sungwoo Lee, SoHyun Park, Daeryong Kim, Jungwoo Song, Dongwhan Kim, Joonhwan Lee, and Bongwon Suh. 2020. Understanding user perception of automated news generation system. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA. Association for Computing Machinery.

Picovoice. 2023. Speech-to-text benchmark.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Walter Chang, and Thien Huu Nguyen. 2021. MadDog: A Web-based System for Acronym Identification and Disambiguation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 160–167, Online. Association for Computational Linguistics.

Matthew Shardlow. 2013. The CW corpus: A new resource for evaluating the identification of complex words. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 69–77, Sofia, Bulgaria. Association for Computational Linguistics.

Hoang Van, David Kauchak, and Gondy Leroy. 2020. AutoMeTS: The Autocomplete for Medical Text Simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1424–1434, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Amir Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9098–9105.

Amanda Weirup and Phylicia Taylor. 2024. What Do You Mean? Developing Jargon Literacy for the Workplace. *Management Teaching Review*, page 23792981241266465. Publisher: SAGE Publications Inc.

A Prompts and Parameters

Parameters: temperature is set to 0.1, and maximum length is 1000.

A.1 Jargon Identification & Explanation

System Message `Your job is to help a listener understand speeches that might contain jargon terms they are unfamiliar with. You will be given the transcript snippet. For each snippet, the format will be "Transcript: [snippet]". Your task is to first identify any of those terms that the listener might not fully understand, then provide a definition for each term in concise plain language. Your output should be in the format of a list of term-definition pairs. Return only valid JSON in the format [{"term": "definition"}, ...]. Do not include additional commentary or text outside the JSON. Leave the list blank if you think all the terms in the input transcript are common words that don't need additional explanations. Do not include terms that are already in the previously defined term list.`

User Prompt Transcript: {*transcript*}, Previously defined terms: {*defined_terms*}

A.2 Personalization

System Message `You are given a glossary, a user profile, and a user preference list. Your job is to remove terms the user is likely to already understand based on their profile and preference list. The input glossary is provided in valid JSON format, where each item is structured as {"term": "definition"}. Examine only the terms (the keys in the JSON) and remove the terms the user is likely already familiar with from the glossary. Return only valid JSON structured exactly as: {"understood_terms": ["term1", "term2", ...], "refined_glossary": [{"term": "definition"}, ...]}. Do not include any extra commentary or text.`

User Prompt Glossary: {*glossary*}, User Profile: {*profile*}, User preference: {*preferences*}

B Controlled Experiment

B.1 Participant Profiles

Table 5: Controlled experiment participant profiles, hiding participant index and randomizing the order for anonymity

Education	Job Role
Statistics PhD	Machine Learning Researcher
Computer Science Master	Research Engineer
Applied Mathematics PhD	Oceanography Researcher
Computer Science Master	Data Engineer
Physics PhD	Quantum Researcher
Civil Engineering PhD	Earth Science Researcher
Computer Science Bachelor	Application Engineer

B.2 Self-reported measures survey

- *Comprehension confidence*: "How confident do you feel in your understanding of the presentation?"
- *Engagement*: "How engaged were you while following the presentation?"
- *Perceived value*: "How valuable do you think the presented work is?"

B.3 Statistical Analysis

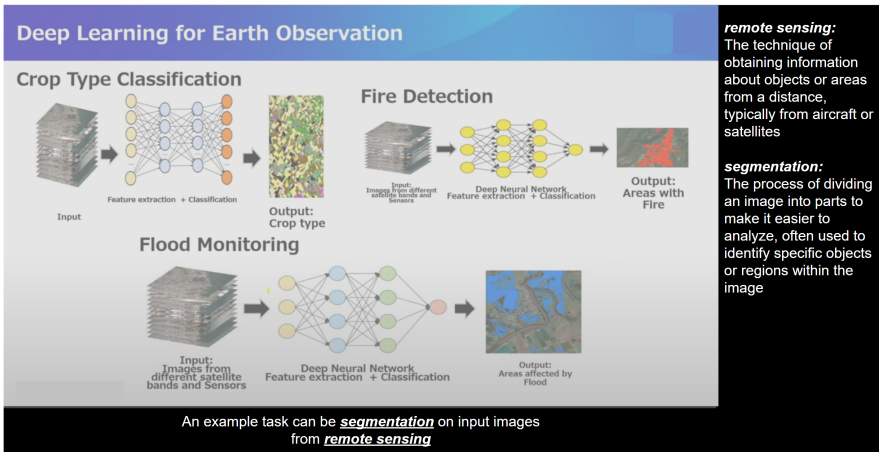
We report the statistical comparisons on the Likert-scale metrics between conditions using Wilcoxon signed-rank test. We chose this non-parametric approach given the ordinal nature of Likert-scale data and the relatively small participant sample size (Mircioiu and Atkinson, 2017). Effect sizes (Cohen's d) were calculated. Holm–Bonferroni corrections were applied to control for Type I errors due to multiple comparisons (Aickin and Gensler, 1996), and we report significance based on the corrected p-values. Full statistical test results are shown in Table 6.

B.4 Generic vs. Personalized Condition Example Figures

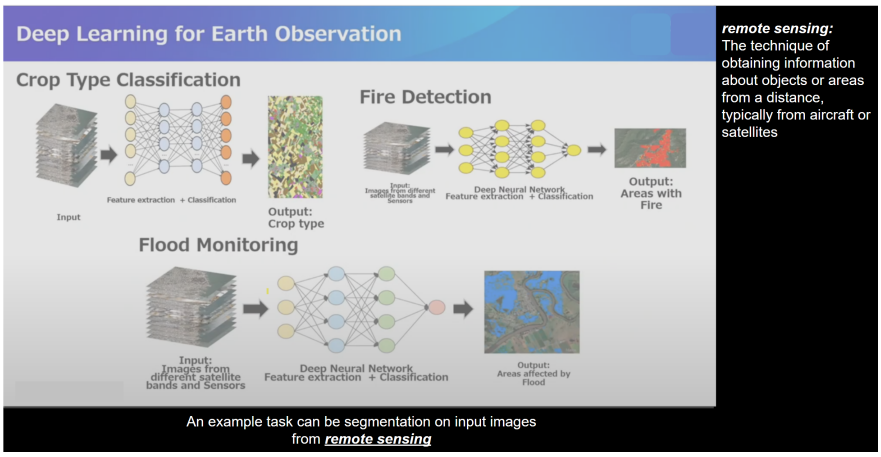
Figure 3a and 3b show a comparative example of the presentation recordings under generic and personalzied condition, respectively.

Table 6: Complete test statistics for all metrics, including w (Wilcoxon signed-rank test), corrected p-value for w, and Cohen's d. Holm–Bonferroni method was used for post-hoc correction.

	Generic vs Baseline			Personalized vs Baseline			Personalized vs Generic		
Metric	w	p_w	d	w	p_w	d	w	p_w	d
Comprehension	56.0	0.0294	0.6682	86.5	0.0047	1.2455	24.5	0.0294	0.5316
Engagement	26.0	0.7395	-0.1988	31.5	0.2733	0.2938	25.5	0.0710	0.5942
Value	48.0	0.0658	0.4242	72.5	0.0073	1.1127	21.0	0.0196	0.7687



(a) Generic Condition



(b) Personalized Condition

Figure 3: Generic vs. Personalized: (a) shows generic jargon support that assumes the listener has no expertise of any domains; (b) shows personalized jargon support in a scenario that the listener provides a profile indicating background in computer vision but not earth science, so ***segmentation*** is filtered out as the listener likely already knows and only ***remote sensing*** is displayed.