



PDF Download
3726302.3730146.pdf
10 February 2026
Total Citations: 0
Total Downloads: 1689

 Latest updates: <https://dl.acm.org/doi/10.1145/3726302.3730146>

DEMONSTRATION

InstInfo: A Just-in-Time Literature Recommendation System for Presentations

KEVIN ROS, University of Illinois Urbana-Champaign, Urbana, IL, United States

RAHUL SURESH, University of Illinois Urbana-Champaign, Urbana, IL, United States

CHENGXIANG ZHAI, University of Illinois Urbana-Champaign, Urbana, IL, United States

Open Access Support provided by:

University of Illinois Urbana-Champaign

Published: 13 July 2025

[Citation in BibTeX format](#)

SIGIR '25: The 48th International ACM
SIGIR Conference on Research and
Development in Information Retrieval
July 13 - 18, 2025
Padua, Italy

Conference Sponsors:
SIGIR

InstInfo: A Just-in-Time Literature Recommendation System for Presentations

Kevin Ros
kjros2@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Rahul Suresh
rsuresh4@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

ChengXiang Zhai
czhai@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Abstract

The efficient discovery of academic literature is critical for research progress, yet many researchers have difficulties in finding literature. This work proposes InstInfo: a novel just-in-time literature recommendation system for presentations. InstInfo transcribes audio in real-time and recommends literature according to the ideas being discussed, thereby helping researchers ground presentations in academic literature while saving them the time of having to manually search. Informal usability studies show that InstInfo is easy to use and that researchers find value in the recommendations. InstInfo can be accessed at <https://instinfo.com>.

CCS Concepts

• **Information systems** → *Specialized information retrieval; Search interfaces; Recommender systems.*

Keywords

just-in-time; recommendation; literature discovery; spoken dialogue

ACM Reference Format:

Kevin Ros, Rahul Suresh, and ChengXiang Zhai. 2025. InstInfo: A Just-in-Time Literature Recommendation System for Presentations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3726302.3730146>

1 Introduction

The ability to discover literature is a critical aspect of any researcher’s workflow. This ability not only provides foundational knowledge for shaping the direction of research projects, but it also minimizes duplicate work, avoids unknown unknowns, and strengthens foundational knowledge. The need to discover literature is also time sensitive. Ideally, researchers are made aware of relevant literature as early as possible in a new project to minimize any negative consequences. Despite this importance, the ever-increasing number of published papers makes literature discovery challenging, especially in a timely manner.

From preliminary analysis, it was found that researchers frequently report wanting to discover literature related to the topics of real-time presentations but give up the search process (or do

not start it at all) due to various barriers. These barriers mainly stem from the synchronous nature of real-time presentations: if the researcher focuses their attention on discovering literature, then they might miss important aspects of what is being said, they might delay or hold up others, or they might appear distracted or uninterested.

As a first step towards mitigating these barriers, this work proposes InstInfo (a portmanteau of **I**nstant and **I**nformation), a novel just-in-time recommendation system [20] to help researchers discover literature related to the topics of spoken presentations. InstInfo provides a user interface where one or more researchers can record and transcribe audio, and this transcription is then used to retrieve related research literature. The corresponding literature references are then provided back to the researchers through the UI in real-time, which helps alleviate the need to spend time discovering the literature themselves. The transcriptions and recommendations are persisted, so that researchers can also follow up with the discovered literature post-presentation for future research.

2 Related Work

This work could be considered an instance of a just-in-time information retrieval agent (JITIR) [20], which aims to proactively predict and resolve user information needs within a local context without requiring any explicit actions (e.g., queries). There have been many instances of these agents, from early work such as Watson [5] and Remembrance Agent [21] to the recent use of spoken conversations as a way to predict and resolve information needs. Andolina et al. proposed SearchBot [2], a proactive search support system that provides web search results based on entities extracted from conversations between two people. In a similar fashion, Liu et al. [14] proposed Visual Captions, an online visual meeting add-on that provides images based on the topics that the participants are discussing, such as recent vacations or locations. InstInfo is different from these lines of work in that InstInfo focuses on the specific use case of retrieving research literature similar to ongoing presentations, and that it also persists the literature for follow-up research goals.

More generally, InstInfo can be considered an instance of a citation recommendation system [3, 7, 8, 10, 11, 13, 15]. In the vast majority of prior literature, citation recommendation approaches have focused on either finding literature related to a full paper (global) or to a specific sentence or paragraph (local). In contrast, InstInfo provides citations to *transcript* segments, which are semantically different than academic paper segments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR '25, Padua, Italy
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1592-1/2025/07
<https://doi.org/10.1145/3726302.3730146>

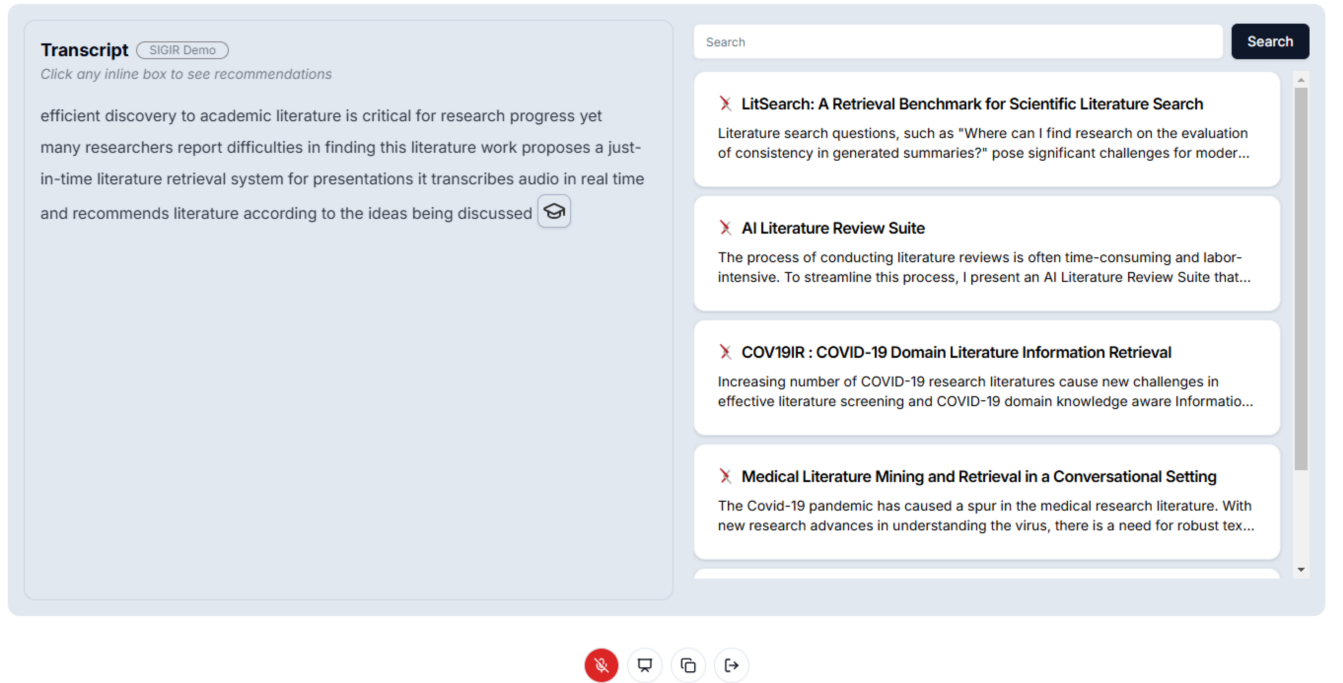


Figure 1: The user interface of InstInfo during a presentation. As the audio is transcribed, it is displayed on the left-hand side of the screen. Periodically, recommendations are presented as inline citations and fully displayed on the right-hand side of the screen.

3 System Overview

This section describes the user interface and infrastructure of InstInfo. At a high level, InstInfo works by recording audio, transcribing audio in real-time, and providing literature recommendations based on the transcriptions back to the user.

3.1 User Interface

The homepage of InstInfo provides the user with the ability to "Create a New Session" or to "Join an Existing Session". Once selected, the user will be shown a screen similar to that of Figure 1. This screen is divided into two main components: the transcript (left) and the recommended literature (right). The text of the transcribed audio appears in the transcript section in real time. As depicted, at certain points in the conversation, small academic icons will be inserted into the transcript text. These icons represent new literature recommendations, and the corresponding links are displayed on the right-hand side of the screen automatically. Should the user want to revisit an old recommendation, they are able to click the respective academic icon in the transcript, and the feed will display the corresponding recommendations. The user can also use the search bar, located at the top of the recommendation feed, to manually query the literature.

Beneath the transcription and the recommendation main components are the session controls. The far-left button lets the user mute and unmute themselves. The far-right button lets the user end or leave the recording session. And the middle two buttons are intended to assist the user with collaboration. As previously

mentioned, users have the ability to "Join an Existing Session" - this is done by pasting the session ID, which can be copied by session participants via the third (copy) button and sent to others. All users in a single session see the same transcript and recommendations as the presentation progresses. The second button allows the creator of the session to toggle between presentation and community mode. In presentation mode, only the creator of the session can unmute and talk - all other members are muted. In community mode, any member of the session can unmute and speak. This functionality is intended to provide system flexibility between in-person and online presentations.

Once the session is ended, then it is saved in the user's "Past Sessions". Here, the user can browse the transcript, recommendations, and any queries and clicks that were made by users in the session. Sessions can also be resumed, which will load the session transcript to the interface as presented in Figure 1 and allow the user to continue recording.

3.2 System Architecture

3.2.1 Overview. The system architecture for InstInfo is presented in Figure 2. Starting at the top-left of the diagram, a session is initiated by the user via the Website, which sets up a socket connection between the Website and the Backend Manager server. When the recording begins (i.e., the user is unmuted), Spoken Audio is sent in two-second chunks from the Website to the Backend Manager. The manager then submits each chunk as a transcription job to the Transcription Queue. The Transcription Queue is monitored by

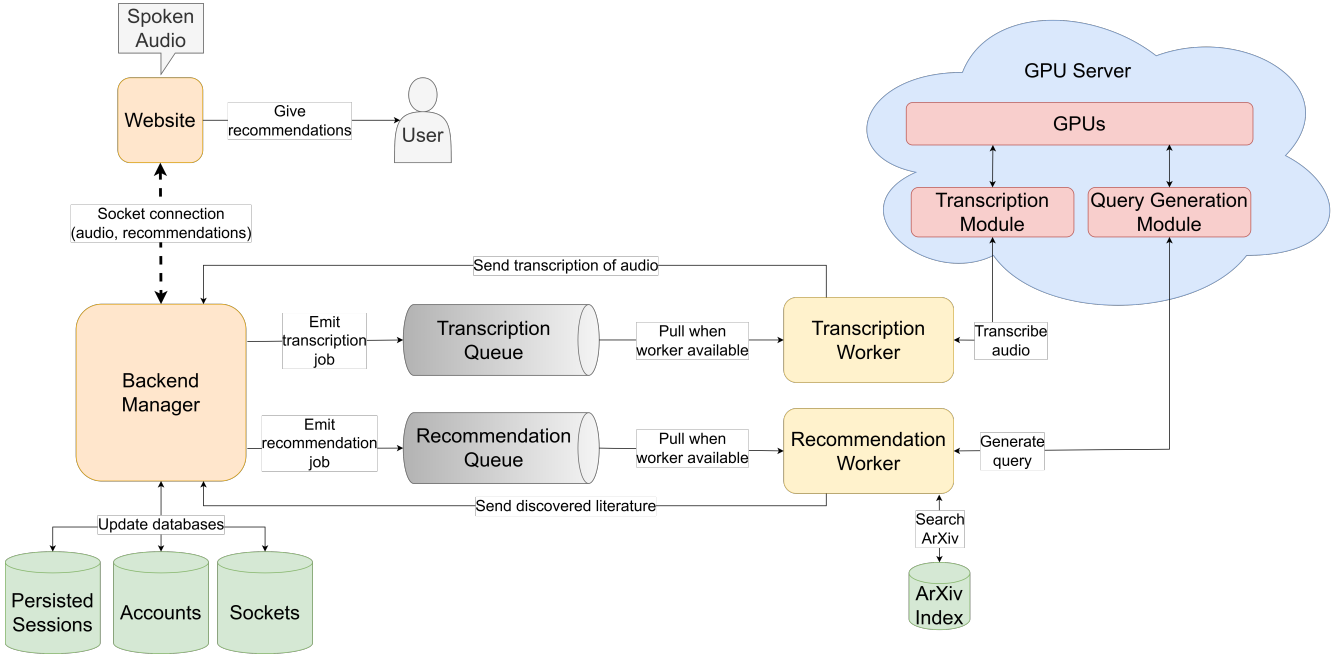


Figure 2: The system architecture of InstInfo. The system is designed to scale while maintaining a transcription round-trip time of around two seconds.

the Transcription worker, which pulls chunks from the queue and transcribes the audio using the Transcription Module, which runs on a GPU server. The Transcription Worker sends the transcription of the audio to the Backend Manager, which then sends it via the socket connection back to the Website. Concurrently, the Backend Manager then submits a recommendation job to the Recommendation Queue. This job is processed when a Recommendation Worker is ready. Here, the Recommendation Worker examines the recent transcript chunks uses the Query Generation Module to (a) determine if literature should be recommended at this point, and (b) if so, construct a query using the transcript. Then, it uses this query to search the ArXiv Index. The Recommendation Worker sends the discovered literature to the Backend Manager, which then sends it via the socket connection to the Website. The Backend Manager also manages the databases to update and persist sessions, accounts, and socket connection information. As it is important to minimize delays, both the Workers and the GPU models can be scaled with increased demand, so that requests across various users can be processed in parallel.

3.2.2 Transcription. One of the challenges of building a system like Instinfo is ensuring that there is minimal delay between the spoken audio and when the transcription appears on the screen. If the transcription time takes too long, then the recommendations will not be relevant to the current topic. To address this, audio is sent by the Website in two-second chunks. For each audio chunk, the Transcription Worker uses FFmpeg [23] to remove background noise and detect silence. If full silence is not detected, then the chunk is sent to the Transcription Module. The Transcription Module uses whisper-large-v3-turbo [19], a state-of-the-art model for audio

transcription. Following the transcription, Aeneas is used for forced alignment [18] to avoid transcribing boundary words cut off by the ends of the two-second stream interval. The entire round-trip time for transcription, once a chunk is received, is approximately 1.5 seconds, meaning that the time between hearing the audio live and seeing the transcript appear on the screen is between 1.5 to 3.5 seconds.

3.2.3 Recommendation. Another challenge of building this system is to be able to test various recommendation algorithms. One benefit of the recommendation design described below is that it is modular: a new algorithm can be added or removed without any changes to other portions of the system. This is achieved through the Backend Manager periodically emitting a recommendation jobs. This job is pulled by the Recommendation Worker, and includes the last N seconds of the spoken transcript. The Recommendation Worker uses the Query Generation Module to decide if a recommendation here is appropriate and, if so, what the search query should be. Currently, the Query Generation Module uses Llama-3-8B [1], and the prompt instructs the model to do the following: (1) summarize the transcript segment, (2) predict whether or not any academic concepts, such as papers, methodologies, techniques, or findings are mentioned, and (3) if so, generate a keyword query.

If the prediction is negative, then no action is taken. Otherwise, the query is used to search the ArXiv Index. The ArXiv Index is an OpenSearch [16] index of the ArXiv metadata dataset hosted by Kaggle [12], consisting of around 2.6M arXiv paper abstracts and metadata. The search formula used is the default instantiation of BM25 [22]. The top three to five matching papers are returned to

the Backend Manager and then pushed to the Website for display to the user.

3.2.4 Queueing. A third challenge of the system design is ensuring parallelization, which is achieved by building the both the Transcription Queue and the Recommendation Queue on top of RabbitMQ [4], a messaging and streaming broker. The Backend Manager connects to these queues and passes jobs to queues. The Transcription Worker and the Recommendation worker subscribe to the respective queues, and pull and process the jobs when ready. Note that the response from the Workers to the Backend Manager is done via a direct HTTP request.

3.2.5 Infrastructure. The Website, Backend Manager, Workers, and Queues are all hosted on a Microsoft Azure virtual machine. MongoDB Cloud is used to manage the databases. A separate private lab GPU server is used to host the Transcription Module, the Query Generation Module, and the ArXiv Index. Each module on the Azure server and the GPU server is running within a Docker container [9] and exposed by either Gunicorn [6] or Flask [17].

4 Evaluation

A handful of informal early usability interviews have been conducted where researchers use InstInfo in scenarios simulating literature discovery during and following a conference presentation. While it is too early to draw any strong conclusions, preliminary observations and feedback indicate that participants generally dislike having to manually search for literature during live conference presentations due to the potential of missing information. With InstInfo, participants used the provided literature recommendations and the post-presentation aggregations to help guide future search query refinements. A larger full-scale usability study is planned, and its findings will be reported in a future paper.

5 Demonstration

During the conference, we will show a live demonstration of InstInfo. We will have a laptop with a microphone where conference attendees can pretend to present their work or have a conversation about their research to see the resulting recommended literature. Moreover, we also plan to make the platform publicly accessible to any conference attendee so that they can use it themselves during any of the scheduled paper presentations.

6 Conclusion and Future Work

This demonstration presents InstInfo: a novel just-in-time literature recommendation system for presentations. With InstInfo, researchers can ground academic presentations in related literature with minimal effort. The architecture of InstInfo helps maintain efficiency and responsiveness while allowing the flexibility for scale in the future. InstInfo presents an exciting development towards real-time proactive retrieval agents, and it opens up new opportunities studying how such systems can be implemented, tested, and validated.

There are numerous directions for future technical improvements. Possible directions include expanding the coverage of search results, improving the recommendation algorithm, and refining the search algorithm to leverage the transcript as additional context.

In addition to this, there are many research challenges that need to be addressed to increase the utility of InstInfo. One major challenge is understanding user preferences for the interface, the speed, and the important metrics, which we hope to address in our aforementioned full-scale usability study. Second, properly finding and inserting citations into transcripts is an open challenge, especially in real time. And a third challenge is measuring the effectiveness of tools like this, as traditional measurements of relevance or click-through rates may not indicate the actual utility provided by the system.

7 Acknowledgements

This work is supported in part by the National Science Foundation and the Institute of Education Sciences under Grant DRL-2229612 and by the SRI program at the University of Illinois Urbana-Champaign.

References

- [1] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
- [2] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. Investigating proactive search support in conversations. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 1295–1307.
- [3] Steven Bird, Robert Dale, Bonnie J Dorr, Bryan R Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R Radev, Yee Fan Tan, et al. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *LREC*.
- [4] Broadcom. 2025. *RabbitMQ*. <https://www.rabbitmq.com/>
- [5] Jay Budzik and K Hammond. 1999. Watson: An Infrastructure for Providing Task-Relevant, Just-In-Time Information. In *AAAI 1999 Workshop on Intelligent Information Systems (Orlando FL USA)*. Citeseer.
- [6] Benoit Chesneau and Paul J. Davis. 2025. *gunicorn*. <https://github.com/benoit/gunicorn>
- [7] Tao Dai, Li Zhu, Yaxiong Wang, and Kathleen M Carley. 2019. Attentive stacked denoising autoencoder with bi-lstm for personalized context-aware citation recommendation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2019), 553–568.
- [8] Ying Ding, Guo Zhang, Tamy Chambers, Min Song, Xiaolong Wang, and Chengxiang Zhai. 2014. Content-based citation analysis: The next generation of citation analysis. *Journal of the association for information science and technology* 65, 9 (2014), 1820–1833.
- [9] Docker. 2025. *Docker*. <https://www.docker.com/>
- [10] Travis Ebesu and Yi Fang. 2017. Neural citation network for context-aware citation recommendation. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 1093–1096.
- [11] Wenyi Huang, Zhaohui Wu, Chen Liang, Prasenjit Mitra, and C Giles. 2015. A neural probabilistic model for context based citation recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.
- [12] Kaggle. 2025. *arXiv Dataset*. <https://www.kaggle.com/datasets/Cornell-University/arxiv>
- [13] Yicong Liang and Lap-Kei Lee. 2023. A Systematic Review of Citation Recommendation Over the Past Two Decades. *International Journal on Semantic Web and Information Systems (IJSWIS)* 19, 1 (2023), 1–22.
- [14] Xingyu "Bruce" Liu, Vladimir Kirilyuk, Xiuxiu Yuan, Alex Olwal, Peggy Chi, Xiang "Anthony" Chen, and Ruofei Du. 2023. Visual captions: augmenting verbal communication with on-the-fly visuals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.
- [15] Zoran Medić and Jan Šnajder. 2020. Improved local citation recommendation based on context enhanced with global information. In *Proceedings of the first workshop on scholarly document processing*. 97–103.
- [16] OpenSearch. 2025. *OpenSearch*. <https://opensearch.org/>
- [17] Pallets. 2025. *Flask*. <https://github.com/pallets/flask>
- [18] Alberto Pettarin. 2025. *Aeneas Forced Alignment*. <https://github.com/readbeyond/aeneas>
- [19] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. doi:10.48550/ARXIV.2212.04356
- [20] Bradley James Rhodes and Pattie Maes. 2000. Just-in-time information retrieval agents. *IBM Systems journal* 39, 3.4 (2000), 685–704.

- [21] Bradley J Rhodes and Thad Starner. 1996. Remembrance Agent: A Continuously Running Automated Information Retrieval System.. In *PAAM*, Vol. 96. 487–495.
- [22] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 232–241.
- [23] Suramya Tomar. 2006. Converting video formats with FFmpeg. *Linux Journal* 2006, 146 (2006), 10.