Meetalk: Retrieval-Augmented and Adaptively Personalized Meeting Summarization with Knowledge Learning from User Corrections

Zheng Chen[1] , Futian Jiang[2] , Yue Deng[1] , Changyang He[3] , Bo Li[1]
[1] Computer Science and Engineering, Hong Kong University of Science and Technology
[2] Data Science and Artificial Intelligence, The Hong Kong Polytechnic University
[3] Max Planck Institute for Security and Privacy
zchenin@connect.ust.hk, alexft@connect.hku.hk,
ydengbi@connect.ust.hk, changyang.he@mpi-sp.org, bli@cse.ust.hk

Abstract

We present **Meetalk**, a retrieval-augmented and knowledge-adaptive system for generating personalized meeting minutes. Although large language models (LLMs) excel at summarizing, their output often lacks faithfulness and does not reflect user-specific structure and style. Meetalk addresses these issues by integrating ASR-based transcription with LLM generation guided by user-derived knowledge. Specifically, Meetalk maintains and updates three structured databases, Table of Contents, Chapter Allocation, and Writing Style, based on user-uploaded samples and editing feedback. These serve as a dynamic memory that is retrieved during generation to ground the model's outputs. To further enhance reliability, Meetalk introduces hallucination-aware uncertainty markers that highlight low-confidence segments for user review. In a user study in five real-world meeting scenarios, Meetalk significantly outperforms a strong baseline (iFLYTEK ASR + ChatGPT-4o) in completeness, contextual relevance, and user trust. Our findings underscore the importance of knowledge foundation and feedback-driven adaptation in building trustworthy, personalized LLM systems for high-stakes summarization tasks.

1 Introduction

Large Language Models (LLMs) have shown impressive capabilities in performing summarization and generation tasks across a wide range of domains. However, a fundamental question remains: How do LLMs utilize knowledge, unstructured, in real-world applications, and how can we ensure that this knowledge is personalized, accurate, and faithful? This question is especially critical in the context of automated meeting minutes generation, where information needs to be not only complete and concise but also aligned with domain-specific writing norms and user preferences.

Although existing approaches have used LLM to generate abstractive meeting summaries, they often fall short in two key areas: (1) the inability to adapt to user-specific structural and stylistic knowledge and (2) the tendency to produce hallucinated or generic outputs due to weak grounding. Furthermore, traditional systems lack mechanisms for learning from user feedback, leading to repeated errors and suboptimal long-term performance in repetitive meeting contexts.

In this work, we propose **Meetalk**, an adaptive meeting minutes generation system that addresses these challenges by tightly integrating *retrieval-augmented generation* (RAG), user-driven knowledge modeling, and hallucination-aware design. Specifically, Meetalk builds and updates structured knowledge bases, including chapter allocation mappings and writing style templates, learning from user-provided examples and edits. At inference time, these personalized knowledge modules are retrieved and injected into LLM prompts to guide faithful and stylistically consistent generation. In addition, we incorporate uncertainty indicators such as "[Not Sure]" labels to make the confidence of the model interpretable to users, thus enabling human-AI collaboration in mitigating hallucinated content.

To evaluate Meetalk, we conducted a controlled user study in five real-world meeting scenarios. Compared to a strong baseline (iFLYTEK ASR + ChatGPT-4o), Meetalk consistently improves output completeness, contextual relevance, and user trust, while significantly reducing time and cognitive load. Our findings suggest that adaptively modeling and utilizing user-specific knowledge not only enhances generation quality, but also provides a promising paradigm for deploying trustworthy, personalized LLM-based systems in professional workflows.

Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM), pages 94–110

2 Backgrouond

2.1 Text-to-Minutes: Evolution from Extractive Methods to Large Language Models

Early research on meeting summarization primarily employed extractive methods (Tur et al., 2008) (Riedhammer et al., 2008) (Tixier et al., 2017), though studies indicated a human preference for abstractive summaries in conversational content (Goyal et al., 2022) (Murray et al., 2010). The rise of LLMs has brought strong semantic capabilities to tasks like meeting minutes generation (Cao et al., 2024), but factual consistency remains a key issue. Studies show that nearly 30% of summaries generated by seq2seq models contain inaccuracies (Cao et al., 2018)(Kryściński et al., 2019). LLMs also face challenges in adapting to subjective preferences, crucial for meeting minutes. Biermann et al.(Biermann et al., 2022) found that users prefer tools that align with their writing styles, but Ippolito et al.(Ippolito et al., 2022)(Lin et al., 2024) noted LLMs struggle to maintain organizational or individual style and format, further complicating their use in this context. Therefore, to develop an accurate and personalized meeting minutes tool, we propose leveraging the capabilities of LLMs while implementing strategic system designs to enhance accuracy and adapt to personal preferences.

2.2 Adaptively Personalized Minutes: RAG and Summary-based Prompt Engineering

User preference modeling plays a crucial role in understanding and adapting to user preferences, thereby enabling the generation of personalized meeting minutes. Researchers have applied machine learning-based user preference modeling in various specific domains. Yang et al. proposed a kernel probability model for color theme evaluation (Yang et al., 2024). Ma et al. introduced CRNN-SA for extracting user music preferences from listening history (Ma et al., 2022). Ma et al. developed SmartEye, a deep learning system that generates real-time photo composition suggestions based on users' previous photos and feedback (Ma et al., 2019).

Recent advancements in LLMs have highlighted the potential of Retrieval-Augmented Generation (RAG) in user preference modeling (Lewis et al., 2020). RAG enhances LLM performance by providing relevant external information, reducing hallucinations, and improving response accuracy. Summary-based prompt engineering for adaptive personalization leverages the power of text summarization to create dynamic, user-tailored prompts (Ait Baha et al., 2023). This approach abstracts essential information conveniently without capturing sensitive details (Friedman et al., 2013). While users often struggle to distill key features to refine their prompts, employing LLMs to extract these features and automatically incorporate them into subsequent prompts offers a convenient solution (Ait Baha et al., 2023). Recent studies have shown that such adaptive systems can significantly improve engagement and satisfaction in various applications, from recommendation (Lyu et al., 2023) systems to personalized learning platforms (Ait Baha et al., 2023).

In the context of personalized meeting minutes, RAG and summary-based prompt engineering can be employed for retrieving users' sample meeting minutes and learning from user modifications on the minute's output.

3 System Design

3.1 Design Goals (DGs)

Motivated by the findings of formative study and existing research, we aim to design an adaptively personalized meeting minutes generation tool with the following **design goals (DGs)**:

DG1. To ***improve minutes quality*** while ***reducing time*** spent on meeting minute generation.

DG2. To integrate users' ***personal preferences*** in meeting minute formats and writing styles.

DG3. To leverage an adaptive approach that streamlines the process for repetitive meeting tasks, ***improving efficiency over time***.

DG4. To enhance the ***visualizations for trustworthiness***, increasing user confidence in the generated minutes.

3.2 Overall Workflow

Meetalk's workflow can be visualized in Figure 1, beginning with the user uploading a sample meeting minutes file and the meeting audio to be processed. The system analyzes the sample file to *suggest* three key components: the Table of Contents (ToC), Chapter Allocation Database, and Writing Style Database. These components serve as adaptive references for the subsequent processing steps, allowing Meetalk to tailor its output to each user's specific needs and preferences.

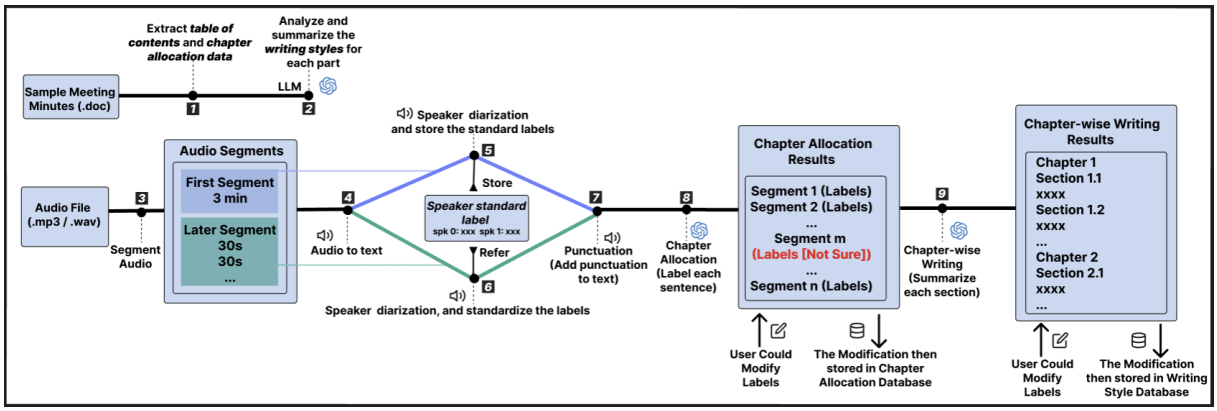

Figure 1: Meetalk's main process: Steps 1-2 sample file data analysis and suggestion, 3-7 transcribe audios, 8 allocates chapters, and 9 involves chapter-wise writing.

Next, Meetalk processes the meeting audio using ASR, dividing it into segments for *transcription, speaker diarization, and punctuation*. As each segment is processed, Meetalk performs *chapter allocation* by referring to the Chapter Allocation Database, labeling the text according to the existing ToC or creating new chapters or sections as necessary. If users find the chapter allocation results inaccurate, they can pause the process and *modify* the chapter allocation labels as easily as editing text. By confirming the modifications, these *changes are incorporated* into the Chapter Allocation Database for future reference.

Once all audio is processed and allocated, Meetalk *generates content for each section* based on the Writing Style Database and the allocated text. Similarly, if users are unsatisfied with the generated results, they can directly modify the content. By confirming the modifications, Meetalk will analyze the modified parts at a high level and *update the Writing Style Database* accordingly.

Throughout this process, Meetalk offers two LLM options: OpenAI ChatGPT API (supporting all available versions), and a locally hosted LLAMA3:8b. This flexibility allows users to choose their preferred LLMs for various needs, balancing factors such as performance, privacy, and cost.

3.3 Databases

Meetalk features three core databases. First, the **Table of Contents Database (ToC)** is responsible for storing the organizational structure of meeting records, specifically the chapters and sections. Second, the **Chapter Allocation Database** archives historical associations between contents and specific chapters and sections. Third, the **Writing Style Database** establishes guidelines and stores details for diverse writing styles. The writing styles for different sections are displayed alongside their corresponding sections in the ToC. These three components can be populated through three methods. *The first method is Referencing to Sample Files*: Meetalk processes reference documents by examining their chapter layouts, recognizing the content within, and summarizing the writing style characteristics. *The second method is Manual User Input*, where users can manually enter data for all three databases. *The third method involves Learning from User Modifications*, Meetalk learns and updates the databases based on user modification on the output. These processes will be explained in detail in a subsequent section.

3.3.1 Chapter Allocation Database

The Chapter Allocation Database is organized into three columns: content, Label A, and Label B. Both label columns follow the format "Chapter xx, Section xx," serving to denote the hierarchical chapter and section to which the content belongs. The inclusion of two labels is based on our rigorous testing results. We conducted a systematic study using a random sample of 200 sentences from meeting transcripts, analyzed in conjunction with their corresponding table of content. Two editors independently labeled each sentence, considering its contextual placement (Cohen's Kappa = 0.92, agreement ratio = 96.5%). Analysis of these labels revealed that 78% of sentences corresponded to a single section, while 22% belonged to two different sections. Therefore, we've included a second label column to accommodate these dual-labeled sentences. For content requiring more than two labels, users can split the same content item into

multiple rows for input, allowing additional labels to be assigned to that content. For example, a content item needing 4 labels can be entered in two rows, with 2 labels assigned to each row.

To allow users conveniently update the databases, as shown in Figure 2, Meetalk provides users with the flexibility to edit table contents, add new entries, and modify chapter allocation outputs. While minor changes need to be made, our database can keep track on the preferences based on these changes. After editing and confirming the edits, users could click the "Save Data" button to upload the edited database and save it as the current chapter allocation database.

Figure 2: Chapter Allocation Databases, with buttons to get chapter allocation data, add rows, delete rows and save data.

The content column of the Chapter Allocation Database is stored as embeddings (referred to as "content embeddings" below). For each entry of the "content" column, an embedding is generated using the *large multilingual E5 text embedding model* (referred to as *"multilingual-e5-large"* below). The multilingual-e5-large model supports 93 languages, primarily English, enabling Meetalk to process meeting minutes in multiple languages, thereby enhancing its global applicability. These embeddings are crucial as they provide a mathematical representation of the text, facilitating later comparison and retrieval.

3.3.2 Writing Style Database

To empower users with the capability to utilize and preserve precise and tailored writing tags, we have defined eleven indicators categorized into three main types: Five for **Writing Context**, five for **Summary Variables**, and one **Difference**.

Figure 3: Writing Style Databases, with 11 columns and buttons to get writing style data, add rows, delete rows, and save data.

Writing Context encompasses the foundational elements necessary for creating the writing piece. These indicators were derived from a comprehensive formative study on meeting minutes requirements across diverse industries, including: **Input:** The scenario of the meeting. **Participant:** The individuals or groups involved in the meeting, including their roles and relevance to the discussion topics. **Writing Goal:** The primary objective of the meeting minutes, such as informing, decision-making, or action planning. **Writing Format:** The required structure or style of the meeting minutes, such as paragraphs, bullet points, or numbers. **Your role:** The viewpoints or roles that need to be represented in the minutes.

Summary Variables are mainly derived from LIWC 2022 definitions, including: **Analytical Thinking:** Measures logical and hierarchical thinking patterns. **Clout:** Reflects social status, confidence, or leadership abilities. **Authenticity:** Indicates honesty, personal disclosure, and genuineness. **Emotional Tone:** Assesses overall emotional tone of the writing. **Language:** *English, Spanish, Traditional Chinese, etc.*

And finally, the **Difference** variable is created to store comparisons between user modifications and original text.

The Writing Style Manager interface includes three main buttons to interact with the writing style data. The *"Get Writing Style Data"* button retrieves the current tag data from the database. Users may then add, delete, or edit rows, uploading their changes using the *"Save Writing Style Data"* button. With this approach, we enable dynamic and iterative improvements in writing style prediction and generation.

4 Knowledge Integration and Utilization in Meetalk

In the era of large language models (LLMs), the ability to effectively ground generation on structured and personalized knowledge is crucial to enhancing output accuracy and trustworthiness. Meetalk addresses this challenge by incorporating a retrieval-augmented and user-adaptive knowledge pipeline into its summarization workflow. This section details how Meetalk constructs, retrieves, and updates knowledge to enable personalized, faithful, and hallucination-aware meeting minutes generation.

4.1 Knowledge as Structured Memory

We conceptualize knowledge in Meetalk as a structured memory composed of three user-specific databases: the Table of Contents (ToC) database, the Chapter Allocation database, and the Writing Style database. These databases are derived from user-provided sample minutes or previous interactions, and encode the organizational structure, topical segmentation, and preferred linguistic style for each meeting domain. Unlike static templates, these knowledge modules dynamically evolve as users revise system outputs.

4.2 Retrieval-Augmented Prompting

To ensure faithful and stylistically consistent generation, Meetalk employs retrieval-augmented generation (RAG) techniques at multiple stages. During chapter allocation, each segment of transcribed audio is embedded and matched against prior content in the Chapter Allocation database to suggest contextual labels. Similarly, in the writing stage, the system retrieves style exemplars from the Writing Style database to construct section-specific prompts. These retrieved signals act as grounding knowledge, guiding the LLM to produce outputs aligned with both the user's structural expectations and domain-specific discourse.

4.3 Knowledge Updating via User Feedback

To support long-term adaptability, Meetalk treats user modifications as implicit knowledge updates. After each editing action—whether modifying chapter boundaries or rewriting section texts—the system summarizes the difference and updates the corresponding database entry. In doing so, Meetalk implements an interactive knowledge editing loop that enables continual refinement of the structured memory without requiring explicit reprogramming or prompt engineering from the user.

4.4 Hallucination Awareness and Uncertainty Markers

To further enhance trust and mitigate hallucinations, Meetalk integrates a lightweight hallucination-aware mechanism. When the system detects uncertain or low-confidence segment-label mappings—based on retrieval inconsistencies or model disagreement—it marks them with a "[Not Sure]" tag in the interface. This allows users to prioritize checking potentially unreliable content, offering a human-AI collaboration path for factuality verification. These uncertainty annotations can also be logged for future benchmarking or fine-tuning, supporting broader efforts in hallucination detection and correction in knowledge-intensive generation tasks.

In summary, Meetalk transforms user interactions into a dynamic knowledge lifecycle: acquiring knowledge from user examples, injecting it via retrieval-augmented prompting, refining it through feedback, and regulating output trustworthiness through uncertainty cues. This design provides a concrete pathway for realizing knowledgeable, user-aligned LLM applications in high-stakes domains such as meeting documentation.

5 Evaluation

To evaluate the effectiveness of Meetalk in supporting the generation of meeting minutes, we conducted a *within-subject study* comparing Meetalk with the conventional approach to automate meeting minutes. As our baseline, we selected *iFLYTEK real-time ASR combined with ChatGPT-4o*. Participants were asked to complete **two tasks**, using the baseline method and Meetalk respectively.

To validate the optimization of our system for handling the repetitive nature of meetings, participants in each task processed **three meeting audios** from a specific scenario, generating meeting minutes in a consistent format. To assess the generalizability of Meetalk, we selected **five different scenarios** and invited participants who were familiar with these scenarios to complete the tasks.

Through these comparisons, we seek to evaluate whether Meetalk outperforms the baseline method in addressing the design goals derived from literature and the formative studies.

Eighteen (N=18) participants are invited to this study, with five different real-world scenarios included: **legal consultations**, **study abroad counseling**, **academic discussions**, **mock interviews**, and **company pitches**. Participants generally span moderate to high levels of expertise within their respective fields.

It is noteworthy that all participants demonstrate high frequency of meetings and substantial usage of language models in their professional contexts, underscoring the relevance of this study to contemporary professional practices. If the audio contains private conversations, any mentions of real names have been cut out beforehand, and this removal does not affect the main content of the meeting.

5.1 The Baseline Method

The baseline method combines two powerful tools: Using iFLYTEK ASR to generate transcript from the meeting audios, and using OpenAI's ChatGPT-4o to write meeting minutes from transcripts. This approach requires users to switch between two separate tools and incurs significant costs.

5.2 Study Procedure

A remote study session for each participant lasted up to 3 hours, divided into three parts: **the pre-study survey, the main study itself, and the post-study interview.** Participants accessed Meetalk via a web browser on a researcher-provided computer through remote control software. Simultaneously, the experimenter communicated with the participants via Zoom or Lark video conferencing.

As for the main process, initially, communicate with the participant to ensure they understand the relationship and purpose of the above materials, as well as the workflow of using Meetalk. Then, proceed with two tasks while recording the time taken for each: ***Task 1:*** Using the three transcripts produced by iFLYTEK, create meeting minutes similar to the sample meeting minutes file using ChatGPT4o for each transcript. Instruct participants to pay close attention to the format and writing style, aiming to match the sample meeting minutes as closely as possible. Participants are allowed to use various tools within ChatGPT4o to accomplish this task. ***Task 2:*** On Meetalk, upload the sample meeting minutes file and click "suggest". Allow participants to freely modify the suggested database. Then, instruct them to click "submit" for chapter allocation, again allowing free editing. Finally, have them click "write" and permit further modifications as needed. Remind participants that their edits will be saved to the database, which may influence the processing of subsequent audio files.

6 Results

In this section, we analyzed objective and subjective results by combining the final study minutes, post-study questionnaires, and screen recordings captured during the process. The subjective ratings on minutes' quality and the ML GUI Heuristics are presented in Figure 4.

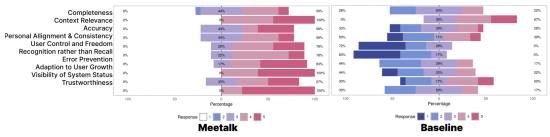

Figure 4: **Ratings for Meetalk(left) and the baseline(right) method** of the subjective 5-point Likert rating results on minutes' quality and user experience based on the ML GUI Heuristics.

6.1 Q1: Meetalk *improves writing quality* while *reducing time*

We recorded the time taken by 18 participants to complete two distinct tasks in this study. The average time for each scenario was calculated and visualized in Figure 5. Overall, Meetalk consistently utilizes less time than the baseline method across all scenarios, (p = 0.0169, Cohen's d = 1.7629), with an average time reduction of 33.9%.

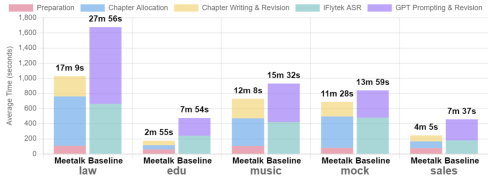

Figure 5: **Meetalk and the baseline method average time comparison** in five studied scenarios. Within a bar, the different colors show the specific time proportion of each stage. It is evident from the figure that the total time used by Meetalk is lower than that of the baseline method.

As shown in 1, the Flesch-Kincaid Reading Ease scores indicate that Meetalk's output has no significant difference on the readability level compared to the baseline's output (p = 0.0688, lower scores indicating easier-to-read text. Regarding word count, Meetalk consistently demonstrates a higher word count percentage across all domains compared to the baseline (p = 0.0114, Cohen's d = 1.9815). This substantial difference in word count percentages indicates that Meetalk consistently produces more extensive content than the baseline, suggesting more detailed or comprehensive responses in each domain.

Based on the participants' ratings of minutes quality, Meetalk-generated minutes generally outperformed those produced by the baseline method on **completeness** (Mean = 3.56 > 2.94, p = 0.0022, Cohen's d = 1.1093), **Context Relevance** (Mean = 4.44 > 3.94, p = 0.0244, Cohen's d = 0.7864), and **Accuracy**(Mean = 4.00 > 2.89, p = 0.0010,

Domain	Law		Edu		Music		Mock		Sales	
	Meetalk	Baseline	Meetalk	Baseline	Meetalk	Baseline	Meetalk	Baseline	Meetalk	Baseline
Flesch-Kincaid Reading Ease **Mean (SD)**	14.03 (0.23)	13.26 (0.55)	16.47 (1.72)	15.20 (2.08)	15.18 (2.83)	10.81 (1.69)	15.88 (1.92)	16.38 (2.87)	18.00 (1.44)	13.36 (2.27)
Word Count Percentage	3825/17945 =21.32%	869/17945 =4.84%	774/1500 =51.6%	476/1500 =31.73%	3566/13012 =27.41%	873/13012 =6.71%	841/4406 =19.09%	151/4406 =3.42%	1480/3595 =41.17%	551/3595 =15.32%

Table 1: Comparison of Meetalk and Baseline across **Flesch-Kincaid Reading Ease** (The lower, the easier to read) and the **word count**

Cohen's d = 1.1971).

• **Completeness:**

Meetalk's score generally outperforms the Baseline's score. Even though both methods cover the main idea, Meetalk provides more details and in-depth explanations, resulting in more comprehensive and complete content. This difference can be attributed to the different approaches Meetalk takes in processing the long transcript. Meetalk accurately extracts all sentences related to a specific section. Then, in a single LLM process, it focuses only on these sentences and rewrites them. By focusing on a specific section, Meetalk can provide richer, more relevant content within a limited generation space. In comparison, the baseline method adopts a full-text summarization, although it touches on the solution part, but only provides an overall summary. It is constrained by the token limit of the LLMs, resulting in limited space allocated to the solution part in the summary.

• **Conext Relevance:** Likert results show that Meetalk consistently achieves slightly higher context relevance scores compared to the baseline method. The low relevance in the baseline method may be attributed to overgeneralization. When processing large amounts of text, language models often attempt to synthesize broad summaries, resulting in vague or generic statements that lack specific, pertinent details (Liu & Lapata, 2019). This tendency towards overgeneralization leads to output that, while broadly related to the input, fails to address the nuances of the given query, significantly reducing its relevance and utility to the user.

• **Accuracy:** Users rated Meetalk's accuracy slightly higher than the baseline's. User feedback indicated that while both methods generally handle explicit numerical data well, the baseline often introduces logical errors that reduce overall accuracy. This issue likely stems from the limitations of traditional summarization techniques in handling long-form content (Liu & Lapata, 2019), which adapted by the baseline method.

6.2 Q2: Meetalk *allows user-driven customization to address personal preferences*

Two metrics in the ML GUI heuristics framework showed noteworthy results. The **Personal Alignment & Consistency** metric showed a positive trend favoring Meetalk over the baseline method (Mean = 3.72 vs. 2.83, $p = 0.1887$, Cohen's d = 0.4472), although the difference was not statistically significant. More compellingly, the **User Control and Freedom** metric demonstrated a highly significant advantage for Meetalk (Mean = 4.11 vs. 1.83, $p < 0.0001$, Cohen's d = 1.9031). These results strongly suggest that Meetalk effectively empowers users to tailor their reading experience according to individual preferences and habits, particularly in terms of providing enhanced control and freedom.

Through user-driven customization, the system achieves alignment consistency by ensuring that formats and writing styles are consistent with both sample files and user preferences. By allowing users to define their own formats and create writing style tags, the system maintains a seamless alignment with users' desired outcomes and expectations. By allowing users to edit or delete suggested content, and to modify Meetalk's output as needed, Meetalk ensures a high degree of user control and freedom.

6.3 Q3: Meetalk *streamlines repetitive meeting tasks with adaptive learning*

Given the repetitive nature of meeting minutes tasks, it's crucial for a system to leverage this characteristic to enhance efficiency. Our study revealed that Meetalk significantly outperformed the baseline method in three critical areas: **recognition rather than recall** (Mean 3.94 > 1.67, $p < 0.0001$, Cohen's d = 2.5820), **error prevention** (Mean 4.33 > 2.72, $p < 0.0001$, Cohen's d = 1.9437), and **adaptation to user modification** (Mean 4.67 > 2.61, $p < 0.0001$, Cohen's d = 2.2194). These results strongly indicate that Meetalk effectively empowers users to tailor their meeting minutes experience, leveraging

the repetitive nature of the tasks to enhance overall efficiency and user satisfaction.

Function **suggestion** significantly enhances this aspect. P3 noted, *"Meetalk's ability to suggest table of contents and writing styles from sample files is incredibly helpful. I don't have to remember everything or keep separate databases."* P12 added, *"It's much easier than the baseline where we had to maintain our own databases and then figure out how to prompt ChatGPT correctly."* This automated suggestion feature allows users to focus more on minutes creation and understanding rather than tedious memorization and retrieval.

6.4 Q4: Meetalk *enhances visualizations for trustworthiness*

The trustworthiness of meeting minutes generation is paramount to its usefulness. Meetalk outperformed the baseline in both **Visibility of System Status** (Mean 4.00 > 3.28, p = 0.0198, Cohen's d = 0.8165) and **Trustworthiness** (Mean 4.78 > 2.89, $p < 0.0001$, Cohen's d = 2.2356), according to participant ratings.

• **Visibility of System Status:** Meetalk provides visibility into three key areas: databases, progress of the chapter allocation process, and final results. P5 commented, *"With Meetalk, I can see everything from the databases being used to how far along the process is. It's so much more transparent than just seeing input and output like with the baseline."* P11 added, *"Being able to track the chapter allocation process in real-time gives me a sense of control and understanding that I didn't have with ChatGPT."* The high degree of visibility allows to reduce uncertainty about system behavior. Users are better able to anticipate and adjust processes, resulting in greater efficiency and accuracy, making participants more confident and proactive in using LLMs.

• **Trustworthiness:** The enhanced visibility of system status, coupled with Meetalk's ability to indicate uncertainties, fosters a true collaboration between human and AI. P2 noted, *"I appreciate that Meetalk shows me what it's unsure about. It feels like we're working together, rather than me just correcting a finished product."* P14 elaborated, *"The constant feedback during the process makes me trust Meetalk more. It's not just a black box spitting out results."*

This approach to transparency and collaboration significantly increases trustworthiness. As P8 summarized, *"With Meetalk, I feel like I'm part of the process, not just an end-user. That makes me trust the results much more than I did with the baseline system."*

7 Conclusion and Discussions

Meetalk addresses the challenges of long meeting minutes generation through innovative chunking and adaptive personalization. By performing ASR on 30-second audio segments and labeling transcribed content for section allocation, Meetalk enhances completeness and relevance, allowing users to review and modify labels in real-time. This process reduces input length for LLMs, improving the quality of summaries. Additionally, the system's flexibility accommodates various data types and user preferences through RAG and summary-based prompt engineering, enabling natural adaptation to user behavior. Meetalk's design also includes an authenticity assessment mechanism that boosts user trust with feedback labels like "[Not Sure]." Overall, Meetalk's approach and principles can be generalized to other AI-driven applications beyond meeting note-taking, enhancing user engagement and facilitating multimodal processing tasks.

In conclusion, this study introduces Meetalk, an innovative adaptive AI system for personalized meeting minutes generation. By addressing key challenges in automated minute-taking, including effectively adapting to personal preferences, Meetalk represents a significant advancement. The system's unique features, such as chapter allocation, chapter-wise writing, and adaptive learning from user modifications, offer a flexible and user-centric approach to generate meeting minutes. Our comprehensive user study across diverse real-world scenarios demonstrates Meetalk's effectiveness in producing high-quality, personalized minutes while enhancing user experience and trustworthiness. These findings validate Meetalk's practical applicability, and further contribute valuable insights to the broader domain of personalized AI-assisted text processing and summarization. As organizations continue to rely heavily on meetings for information exchange and decision-making, systems like Meetalk have the potential to significantly improve productivity and communication effectiveness. Future research can build upon this foundation, further exploring the integration of adaptive personalization in various professional contexts and expanding the capabilities of AI-assisted documentation systems.

Limitations

One limitation for our work is that we chose personal computers as the primary device for Meetalk in the user study, since we consider their common use as meeting minute tools. However, we believe that one of Meetalk's core functionality, namely converting speech into structured meeting minutes, can be applicable to other devices, particularly smartphones, which might offer more convenience in audio recording and uploading. Nevertheless, using the system on smaller screens may require UI adjustments, and the user experience could differ. For instance, content review and manual editing might face more challenges, which potentially increases the need for automated support.

Another limitation lies in the failure to use locally deployed LLMs for the user study. Although we include LLAMA3:8b in the design of Meetalk, we still used GPT-4o in our user study in order to be consistent with the most commonly used methods mentioned by the participants in the formative study.This choice, while facilitating a direct comparison of the results, also limits our understanding of how the localized models perform in real-world applications. Future research could explore similar user studies using localized models such as LLAMA3:8b to validate the effectiveness of our proposed approach in real privacy-constrained environments.

Furthermore, we employed the Flesch-Kincaid Reading Ease score and word count percentage as objective measures to assess the quality of the meeting minutes produced. While readability is a crucial aspect of meeting minutes and it provides valuable insights, it does not include all dimensions of content quality. Additionally, word count percentage could somehow refelect the completeness of Meetalk's generated minutes, but we did not measure the quality of the large word counts. Both these two measures provide extra results to triangulate with the subjective assessment of text quality.

Lastly, Meetalk, as a research prototype, has inherent limitations. Our system relies on advanced LLMs like LLAMA3:8b or ChatGPT-4, both requiring significant computing resources. In our experiments, we either deployed LLAMA3:8b locally on a 24GB NVIDIA RTX 4090 GPU or used the ChatGPT-4o API. What's more, the 8k limit of one-sentence summarization, might lead to information gap, in concluding the meeting scenarios. Additionally, the ASR component lacks an interactive learning process, which means the transcription errors can't be automatically corrected based on user modifications. Currently, the system doesn't support real-time audio input, only allowing for audio file uploads. Furthermore, while powerful, the LLM-based text generation is not 100% accurate and can occasionally produce hallucinations or inaccuracies in the generated content.

References

Tarek Ait Baha, Mohamed El Hajji, Youssef Es-Saady, and Hammou Fadili. 2023. The power of personalization: A systematic review of personality-adaptive chatbots. *SN Computer Science*, 4(5):661.

Oloff C Biermann, Ning F Ma, and Dongwook Yoon. 2022. From tool to companion: Storywriters want ai writers to respect their personal values and writing strategies. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference*, pages 1209–1227, New York, NY, USA. ACM.

Yupeng Cao, Zhi Chen, Qingyun Pei, Prashant Kumar, KP Subbalakshmi, and Papa Momar Ndiaye. 2024. Ecc analyzer: Extract trading signal from earnings conference calls using large language model for stock performance prediction. *arXiv preprint arXiv:2404.18470*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Batya Friedman, Peter H Kahn, Alan Borning, and Alina Huldtgren. 2013. Value sensitive design and information systems. *Early engagement and new technologies: Opening up the laboratory*, pages 55–95.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Daphne Ippolito, Ann Yuan, Andy Coenen, and Sehmon Burnam. 2022. Creative writing with an ai-powered writing assistant: Perspectives from professional writers. *arXiv preprint arXiv:2211.05030*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Susan Lin, Jeremy Warner, JD Zamfirescu-Pereira, Matthew G Lee, Sauhard Jain, Shanqing Cai, Piyawat Lertvittayakumjorn, Michael Xuelin Huang, Shumin Zhai, Björn Hartmann, et al. 2024. Rambler: Supporting writing with speech via llm-assisted gist manipulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–19.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Shuai Ma, Zijun Wei, Feng Tian, Xiangmin Fan, Jianming Zhang, Xiaohui Shen, Zhe Lin, Jin Huang, Radomír Měch, Dimitris Samaras, et al. 2019. Smart-eye: assisting instant photo taking via integrating user preference with deep view proposal network. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12.

Xichu Ma, Yuchen Wang, and Ye Wang. 2022. Content based user preference modeling in music generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 2473–2482, New York, NY, USA. Association for Computing Machinery.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th international natural language generation conference*.

Korbinian Riedhammer, Daniel Gillick, Benoit Favre, and Dilek Hakkani-Tür. 2008. Packing the meeting summarization knapsack. In *INTERSPEECH*, pages 2434–2437.

Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Proceedings of the workshop on new frontiers in summarization*, pages 48–58.

Gokhan Tur, Andreas Stolcke, Lynn Voss, John Dowding, Benoît Favre, Raquel Fernández, Matthew Frampton, Michael Frandsen, Clint Frederickson, Martin Graciarena, et al. 2008. The calo meeting speech recognition and understanding system. In *2008 IEEE Spoken Language Technology Workshop*, pages 69–72. IEEE.

Bailin Yang, Tianxiang Wei, Frederick W. B. Li, Xiaohui Liang, Zhigang Deng, and Yili Fang. 2024. Color theme evaluation through user preference modeling. *ACM Trans. Appl. Percept.*, 21(3).

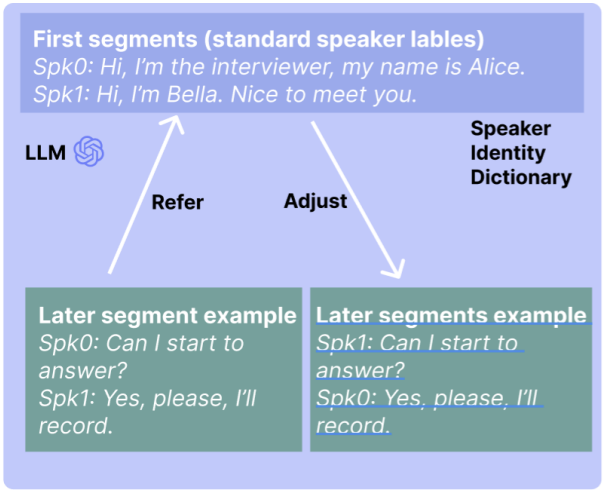

Figure 6: Meetalk's speaker diarization example in an interview scenario: The initial segment identifies Speaker 0 as the interviewer and Speaker 1 as the interviewee, storing their utterances in a **Speaker Identity Dictionary**. In later segments, even if speakers are initially mislabeled due to isolated analysis, the system corrects these labels by referencing the Dictionary, ensuring consistent speaker identification throughout the interview.

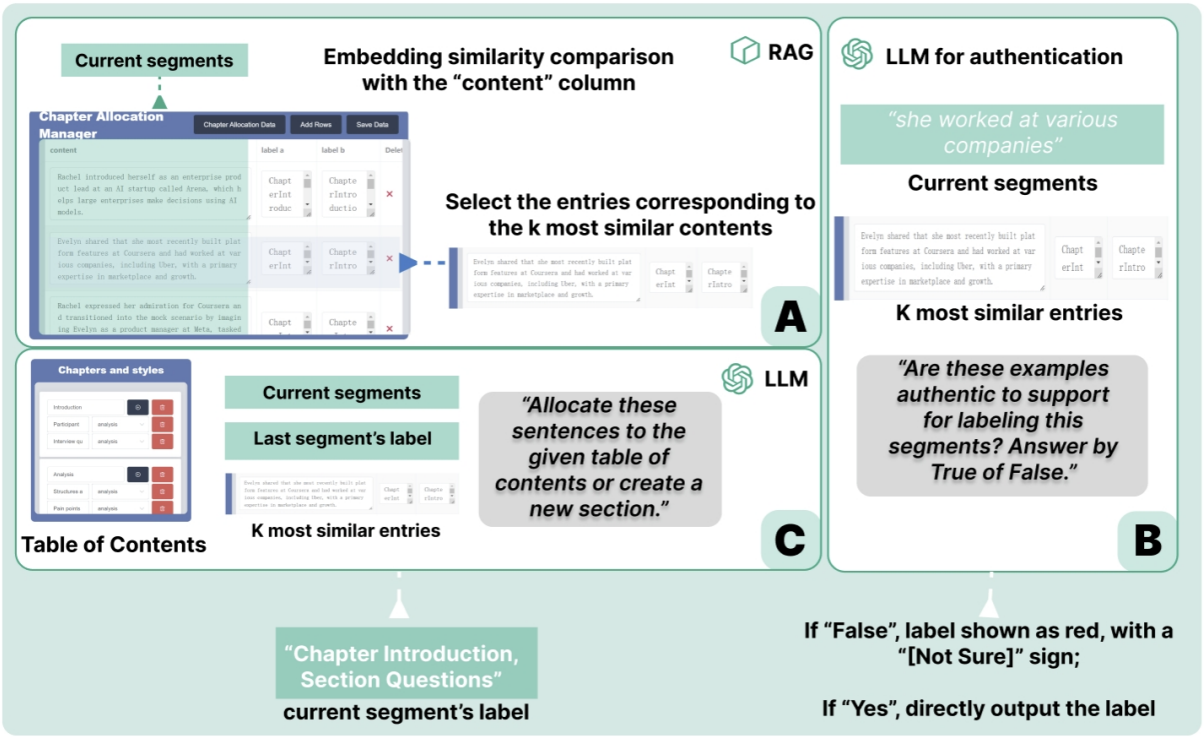

Figure 7: Chapter Allocation Procedure. **Step A**: Retrieve two entries with similar contents to the current segments. **Step B**. Leveraging an LLM to judge whether the retrieved two entries are authentic or not. If False, the label will be shown as red with a "[Not Sure]" sign. **Step C**. Request: Prompt ToC, current segments, last segment's label, and the retrieved two entries to an LLM, for generating the label for the current segment.

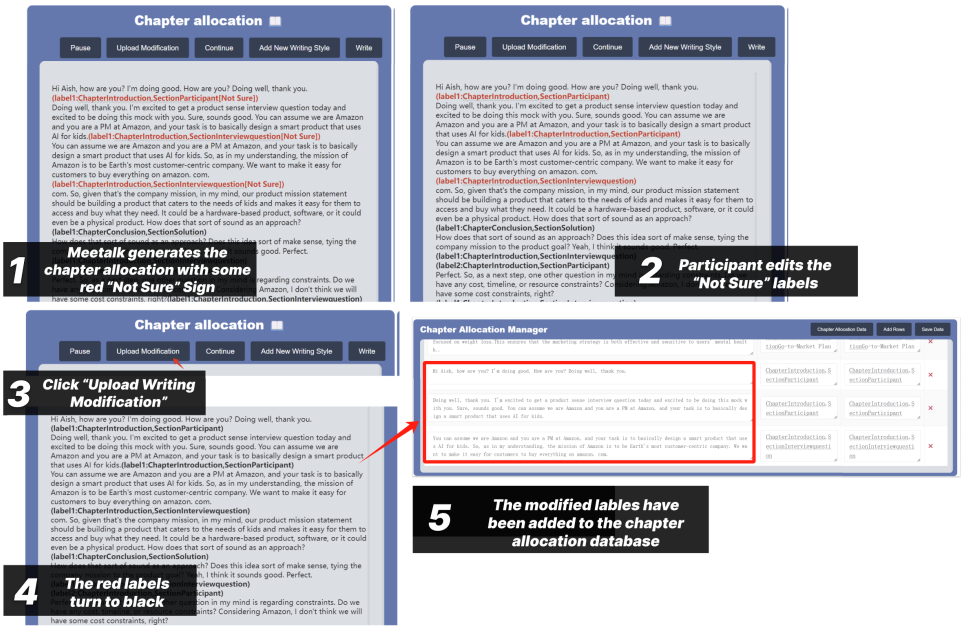

Figure 8: Chapter allocation modification procedure. Participants are notified with the unauthentic labels by a red "[Not Sure]" sign. By modifying these unauthenntic labels and clicking the "Upload Writing Modification" button, the modified labels turn black and been added to the chapter allocation database.

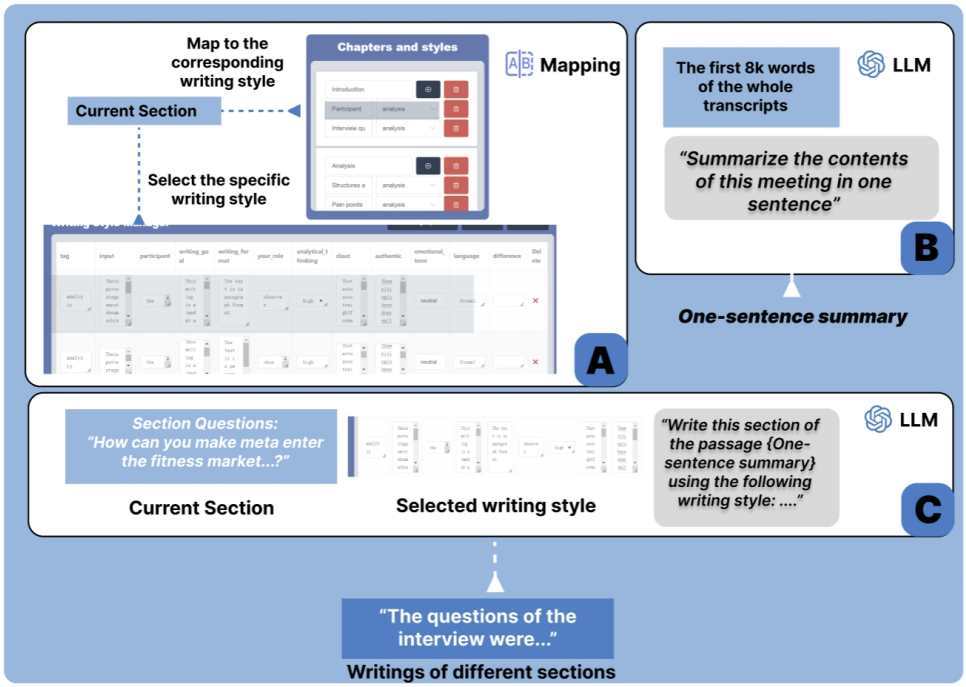

Figure 9: Chapter-wise Writing Procedure. **Step A**: Map the writing style with current section. **Step B**. Summarize the first 8k words (compatibility of the LLMs) of the whole transcripts with one sentence. **Step C**. Prompt current section and the writing style to an LLM for writing this section.

Figure 10: Chapter-wise writing revision procedure

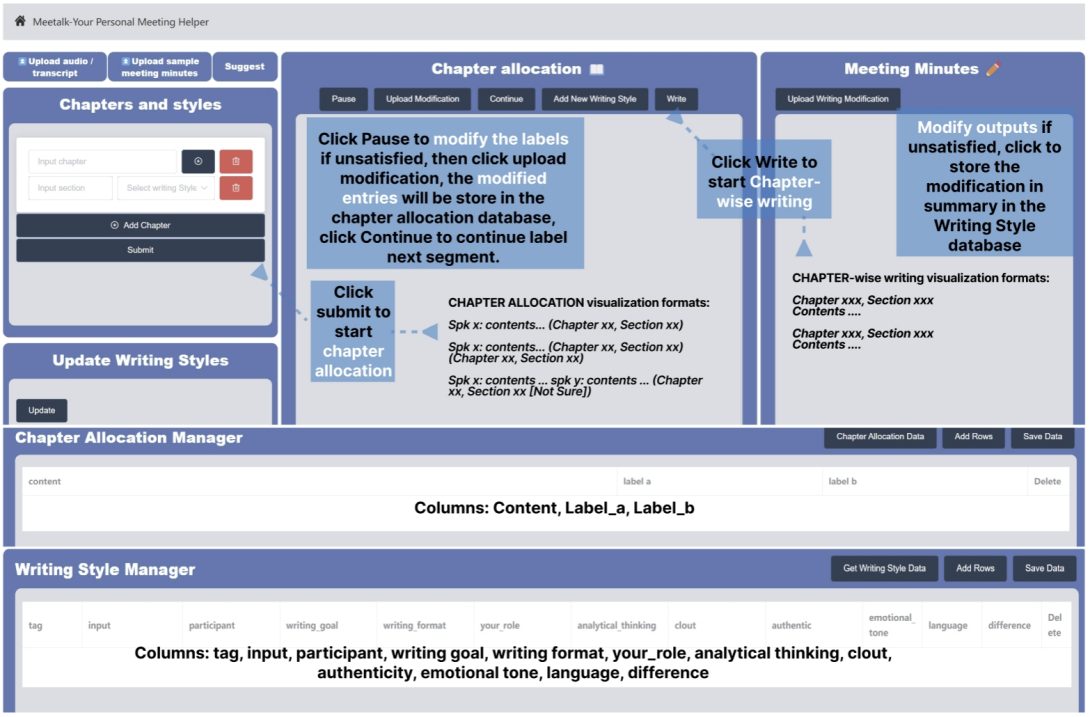

Figure 11: Meetalk, an adaptively personalized meeting minutes generation system. As illustrated in the Meetalk User Interface, after uploading meeting audio and a sample meeting file, Meetalk suggests a table of contents, chapter allocation data, and writing style data based on the sample file to personalize the meeting minutes. After that, Meetalk starts chapter allocation to label each segment according to the table of contents. Finally, meetalk write for each section to form a final meeting minutes. For both the chapter allocation and chapter-wise writing procedure, users could modify the outputs and Meetalk will learn the modifications to better adapt to user preferences.

Table 2: demographics, meeting frequency, and LLM usage of study participants

Scenario	ID	Age	Gender	Degree	Occupation	Meeting Freq.	LLM Usage
Legal consultations	P1	18-24	M	Bachelor	Lawyer trainee	Daily	Daily
	P2	18-24	F	Undergrad.	Law student	Weekly	Daily
	P3	25-34	M	Bachelor	Junior lawyer	Daily	Daily
Study abroad counseling	P4	25-34	F	Bachelor	Consultant	Weekly	Weekly
	P5	25-34	F	Master	Teacher	Weekly	Weekly
	P6	25-34	F	Bachelor	Teacher	Weekly	Weekly
	P7	25-34	M	Postgrad.	Senior postgraduate	Weekly	Daily
	P8	18-24	F	Postgrad.	Senior postgraduate	Monthly	Weekly
Academic discussions	P9	18-24	M	Undergrad.	Music major	Monthly	Daily
	P10	35-44	M	Ph.D.	Lecture tutor	Weekly	Daily
Mock Interviews	P11	25-34	M	Bachelor	HR intern	Weekly	Daily
	P12	25-34	M	Bachelor	HR intern	Weekly	Daily
	P13	25-34	F	Undergrad.	HR intern	Weekly	Daily
Company pitches	P14	35-44	M	Master	Sales manager	Weekly	Daily
	P15	24-34	M	Bachelor	Sales agent	Daily	Daily
	P16	35-44	F	Bachelor	Sales agent	Daily	Daily
	P17	35-44	F	Master	Venture Capital	Daily	Daily
	P18	35-44	M	Master	Venture Capital	Daily	Daily

Figure 12: **Meetalk's Databases UI**: This comprehensive view showcases Meetalk's suggestions following document parsing in the databases' UI. The left panel displays a suggested Table of Contents, while the right side presents a Chapter Allocation Database (top) and Writing Style Database (bottom). These AI-generated recommendations offer a strategic starting point, with full user customization available to tailor the content structure and style to specific needs.

Figure 13: **Meetalk's Full UI illustration**: S1-2, upload meeting audios or transcripts to proceed, and upload sampe meeting minutes file to be referred by the system. After clicking on the suggest button in S3, Meetalk analyzes the uploaded files to suggest Table of Contents, chapter allocation data, and writing style data, as shown in S4. In S5, three buttons in each database are provided to reveiw and revise the suggested data if needed. In S6, while submitting the data to start chapter allocation, and could pause to modify the labels and store the modifications in the chapter allocation database. In S8, users could add writing styles if they are not specified in the table of contents. In S9, click write to start chapter-wise writing, and again in S10, if users are not satisfied with the outputs, modification is allowed and will be summarized in high level to store in the writing style database.

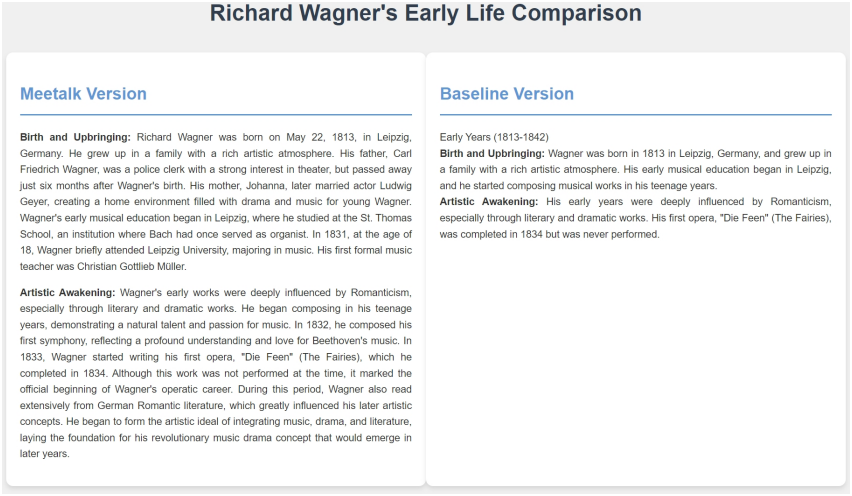

Figure 14: High Readability Example: Audio Musician3

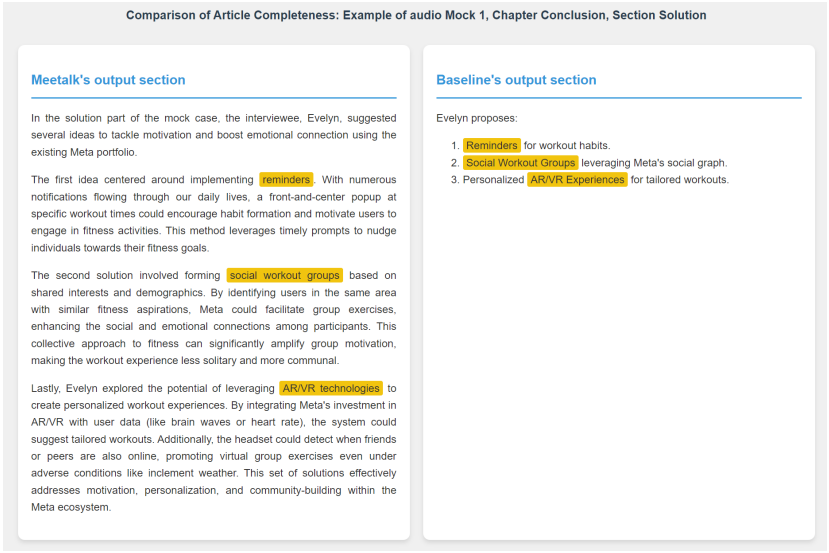

Figure 15: Comparison of **Article Completeness**: Example of audio Mock 1, Chapter Conclusion, Section Solution

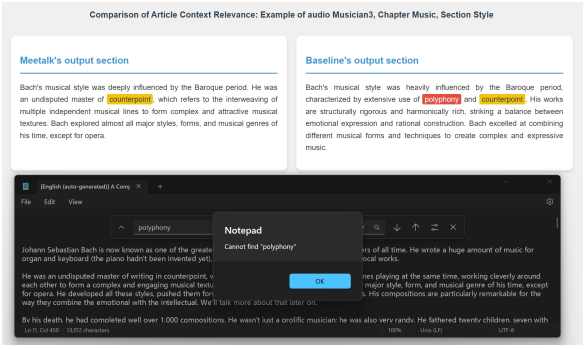

Figure 16: The image above is the **Comparison of Context Relevance** for Example of audio Mock 3, Chapter Music, Section Style. The screenshot below shows **no results for 'polyphony' in the audio Mock 3 transcript**, confirming its absence in the original text.

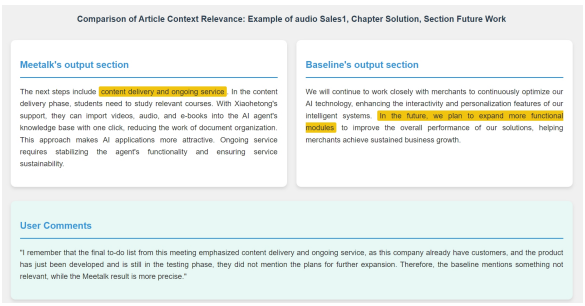

Figure 17: The image above is the **Comparison of Context Relevance** for Example audio Sales 1, Chapter Solution, Section Future work. The screenshot below shows **user comments**, proving the baseline results contain irrelevant information.

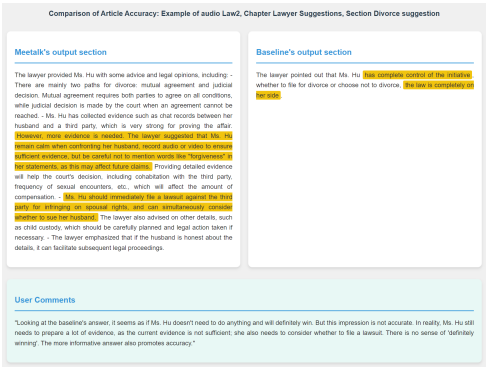

Figure 18: The image above is the **Comparison of Accuracy** for Example audio Law 2, Chapter Suggestion, Section Divorce suggestion. The screenshot below shows **user comments**, proving results of Meetalk are more accurate.

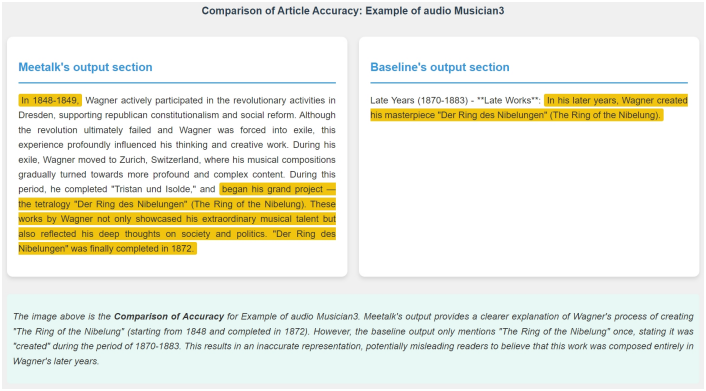

Figure 19: The image above is the **Comparison of Accuracy** for Example of audio Musician3.

Figure 20: Alignment and consistency comparison, with Meetalk got the correct narrative perspective but the baseline method does not.

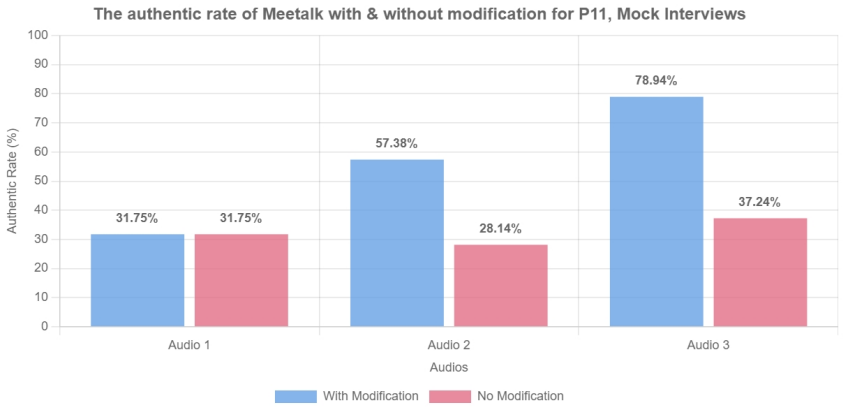

Figure 21: The authentic rate of Meetalk with & without modification for P11's mock interview audio tasks.