A Study of Audio-to-Text Conversion Software and Its Applications to Meeting Minutes and Slide Annotation Generation Tools

March, 2026

Amma Liesvarastranta Haz

Graduate School of
Environmental, Life, Natural Science and Technology

(Doctor's Course)
OKAYAMA UNIVERSITY

Dissertation submitted to
Graduate School of Environmental, Life, Natural Science and Technology
of
Okayama University
for
partial fulfillment of the requirements
for the degree of
Doctor of Philosophy.

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by
Professor Satoshi Denno
and
Professor Yasuyuki Nogami

Okayama University, March 2026.

To Whom It May Concern

We hereby certify that this is a typical copy of the original doctor thesis of
Amma Liesvarastranta Haz

Signature of
the Supervisor

Seal of

Prof. Nobuo Funabiki

Graduate School of Environmental,
Life, Natural Science
and Technology

Abstract

With the rapid growth of online meetings and presentations, their recorded videos have become important sources for getting information in academic and professional environments. The images and voices in these videos contain valuable explanations, discussions, and contexts that may not be captured from visual documents only. Besides, recent advances in *Automatic Speech Recognition (ASR)*, *Natural Language Processing (NLP)*, and *Large Language Models (LLMs)* provide new opportunities to automatically convert verbally spoken contents into structured and accessible textual information.

Raw audio transcriptions often produce lengthy, redundant, and incoherent texts that are difficult to interpret and analyze. For creating meaningful meeting minutes and slide annotations from recorded videos, additional processing steps will be necessary, including summarization, keyword extraction, semantic alignment, and *Optical Character Recognition (OCR)* for slide text extractions. Existing commercial tools that attempt to address these needs can be typically proprietary and offer limited transparency and flexibility, restricting their adaptability for research, customization, and continuous improvements.

In this thesis, I present a design of *Audio-to-Text Conversion Software* and investigate its performance. Then, as its applications, I implement two software tools for automatically generating meeting minutes and slide annotations from recorded meetings and online presentations. The objective of this study is to develop an open-source, multimodal framework that integrates *ASR*, summarization, keyword extraction, *OCR*, and semantic analysis to help users review and comprehend recorded videos of online meetings and presentations more efficiently.

As the first contribution of this thesis, I present an *Audio-to-Text Conversion Software*. This software adopts *Whisper* models from *OpenAI* that support multiple languages and are available in several sizes that differ in accuracy and computational requirements. To make the transcription process accessible through a simple interface, I implement an *ASR* web application using the *Streamlit Python* framework.

The software is evaluated using two categories of metrics. The first category includes the *ASR* related metrics, such as *Word Error Rate (WER)* and transcription speed. The second category includes the web-application performance metrics, such as loading time, response time, and stability under heavy workloads. The second category was measured using the *Locust* stress-testing framework to simulate many concurrent users.

The evaluation results show that the *small.en* model of *Whisper* provides a good balance between transcription accuracy and computational efficiency, producing transcriptions with a *WER* of less than 10%, which indicates strong accuracy for general audio inputs. Through *Locust* stress testing, the web application was also shown to handle up to 180 simultaneous users, with an average response time of 309 milliseconds and a throughput of 471.5 requests per second. These results demonstrate that both the transcription model and the web-based deployment perform efficiently and reliably, making this system a suitable foundation for the subsequent stages of meeting

minutes generation and slide annotation.

As the second contribution of this thesis, I present a *meeting minutes generation tool* from recorded videos. The workflow utilizes the audio-to-text conversion output produced in the previous contribution. The workflow includes summarization, keyword extraction, and system integration. To improve the readability of long *ASR* transcripts, I generate concise summaries using the sequence-to-sequence architecture called *Bidirectional and Auto-Regressive Transformers (BART)* model. For keyword extractions, I investigate several automatic methods, including *KeyBERT*, *RAKE*, *YAKE*, *TextRank*, and *RaKUn2*, to identify approaches that are suitable for different transcription lengths. After constructing these components, I integrate summarization, keyword extraction, audio-to-text conversion, and *Optical Character Recognition (OCR)* into a single open-source meeting minutes generation tool. I adopt an open-source and modular design to ensure that the system remains transparent, accessible, and easy to improve, allowing individual modules to be updated or replaced as newer models and techniques become available.

To evaluate the second contribution, I conducted two sets of experiments to assess both the individual components and the integrated workflow for generating meeting minutes. The first evaluation used three voice recordings with different durations to measure the performance of summarization and keyword extraction. For summarization, I compared the generated summaries with manually created summaries using *ROUGE* metrics and obtained average scores of 50.02% for *ROUGE-1*, 24.04% for *ROUGE-2*, and 43.14% for *ROUGE-L*. For keyword extraction, I compared the extracted keywords with manually selected keywords and calculated an average cosine similarity of 54.09%.

The second evaluation assessed the integrated system using ten recorded presentation videos and showed that the audio-to-text component produced transcriptions with a low *WER*, while the summarization and keyword extraction components worked consistently within the workflow. User feedback also suggested that the generated minutes were usable, although precision and relevance could be improved in several cases. The results confirm that the proposed meeting minutes generation workflow is effective and provides a reliable foundation for automatic annotations of recorded presentations.

As the third contribution of this thesis, I address limitations identified in the previous contribution and improve the precision and coherence of the generated content by the *meeting minutes generation tool*. The modular summarization and keyword extraction components lacked consistency and were unable to capture the full context of the spoken explanations. These limitations become more apparent at the slide level, where presenters often verbally introduce additional details, context, or examples that are not written on the slides. Without slide-level information, users may still find it difficult to follow the flow of the presentation.

To address this need, I implement a *slide annotation generation tool* that applies an integrated approach using *Large Language Models (LLMs)*. The tool begins by detecting slide boundaries and extracting slide text using *Optical Character Recognition (OCR)*. It then converts the narration into text using the *Audio-to-Text Conversion Software* in the first contribution. Instead of relying on modular summarization and keyword extraction, I use a multimodal *LLM* that can jointly interpret the slide images and the narration transcript to produce concise, coherent, and context-aware annotations. By combining visual and textual understanding in a single model, the system is able to generate more accurate slide-level information and provide annotations that support faster and more meaningful review of recorded presentations.

To evaluate the third contribution, I tested the slide annotation generation system using several recorded presentation videos to measure segmentation accuracy, annotation quality, and overall usability. The slide boundary detection component achieved an F_1 score of 0.879 ($SD = 0.024$),

and the annotation alignment reached an accuracy of 90.0%, which indicates highly accurate segmentation across the tested videos. The multimodal *LLM* produced coherent and context-aware annotations, and presenters who reviewed the results reported that the annotations accurately reflected the intended explanations and provided useful support when revisiting the material. I also conducted a usability study with 37 participants, and the result obtained a *System Usability Scale (SUS)* score of 80.5 ($SD = 6.7$), which reflects a high level of user satisfaction. These results demonstrate that the integrated *LLM*-based approach performs effectively and improves both the accuracy and usefulness of slide-level information, confirming that this contribution successfully overcomes the limitations observed in the previous meeting minutes workflow.

In future works, I will focus on improving the accuracy of generated annotations by the *slide annotation generation tool* through incorporating a phoneme-aware transcription correction step. The current tool occasionally suffers from semantic errors caused by *ASR* confusions, which occur when similar-sounding terms are transcribed incorrectly and lead to inaccurate slide annotations. Integrating phonetic information such phoneme-based descriptors may help the system detect and correct these confusions before annotation generation, thereby improving the annotation precision. I also plan to further refine the multimodal *LLM* and explore a fully integrated, end-to-end pipeline that unifies transcription, correction, and annotation to provide a more seamless solution for reviewing recorded presentations.

Acknowledgements

I would like to express my profound gratitude to my supervisor, Professor Nobuo Funabiki, whose exceptional guidance and steadfast encouragement have been central to the completion of this dissertation. His insightful advice, intellectual rigor, and consistent support have shaped both my scholarly development and personal growth. I hold deep respect for his dedication to cultivating an enriching academic environment and for the genuine care he extends to his students. It has been a privilege and an honor to pursue my doctoral studies under his supervision.

I am also sincerely thankful to my co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for their thoughtful feedback, constructive suggestions, and careful examination of my work. Their expertise and commitment have significantly strengthened the quality of this thesis.

My appreciation extends to Dr. Htoo Htoo Sandi Kyaw at Okayama University, whose valuable discussions and dependable assistance have supported the progress of my research. I am equally grateful to Dr. Sritrusta Sukaridhoto from Politeknik Elektronika Negeri Surabaya, Indonesia, for his continuous guidance and generous academic support throughout my studies.

I wish to convey my sincere gratitude to my wife, Dr. Evianita Dewi Fajrianti, whose patience, understanding, and unwavering support have sustained me throughout the challenges of this doctoral journey. Her encouragement has been a constant source of strength, and I am deeply thankful for her presence in every stage of this pursuit.

I gratefully acknowledge the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT)-JASSO for supporting my doctoral studies at Okayama University. I also extend my sincere appreciation to Okayama University for granting me the full tuition exemption each term, which provided essential financial support throughout the course of my degree.

I also extend my appreciation to the members of the Distributed System Design Laboratory. I wish to acknowledge Mrs. Keiko Kawabata and Ms. Safira Kinari for their administrative assistance, and I say thanks to Dr. Yohanes Panduman, and Mr. Komang Candra Brata for the collaborative discussions and shared research activities. I further acknowledge many other colleagues whose support and companionship enriched my time in the laboratory. Although it is not possible to mention everyone by name, I hold the experiences we shared in great regard.

Lastly, I am grateful to my friends in Okayama, particularly those who pursued their studies far from home. Your kindness, solidarity, and encouragement have provided emotional resilience and comfort throughout this journey. I consider myself fortunate to have walked this path alongside you.

Amma Liesvarastranta Haz
Okayama, Japan
March 2026

List of Publications

Journal Papers

1. **Amma Liesvarastranta Haz**, Yohanes Yohanie Fridelin Panduman, Nobuo Funabiki, Evianita Dewi Fajrianti, and Sritrusta Sukaridhoto, "Fully Open-Source Meeting Minutes Generation Tool," Future Internet, vol. 16, no. 11, pp. 429, MDPI, 2024.

2. **Amma Liesvarastranta Haz**, Komang Candra Brata, Nobuo Funabiki, Htoo Htoo Sandi Kyaw, Evianita Dewi Fajrianti, and Sritrusta Sukaridhoto, "A Slide Annotation System with Multimodal Analysis for Video Presentation Review," Algorithms, vol. 19, no. 2, pp. 110, MDPI, 2026.

International Conference Papers

3. **Amma Liesvarastranta Haz**, Evianita Dewi Fajrianti, Nobuo Funabiki, and Sritrusta Sukaridhoto, "A Study of Audio-to-Text Conversion Software Using Whisper Model," in Proc. of the Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE), pp. 268–273, IEEE, 2023.

4. **Amma Liesvarastranta Haz**, Nobuo Funabiki, Evianita Dewi Fajrianti, and Sritrusta Sukaridhoto, "A Study of Summarization and Keyword Extraction Function in Meeting Note Generation System from Voice Records," in Proc. of the 12th International Conference on Networks, Communication and Computing (ICNCC), pp. 106–112, 2023.

5. **Amma Liesvarastranta Haz**, Nobuo Funabiki, Htoo Htoo Sandi Kyaw, Evianita Dewi Fajrianti, and Sritrusta Sukaridhoto, "A Study of Slide Annotation Generating System from Online Presentations," in Proc. of the Eighth International Conference on Vocational Education and Electrical Engineering (ICVEE), pp. 203-208, IEEE, 2025.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

The widespread adoption of the Internet has driven digital transformations across various aspects of human activities, particularly in education and research [1]. Along this trend, academic meetings and discussions have gradually shifted from traditional face-to-face interactions to virtual communications, as online meeting platforms and video presentations have become widely used [2,3]. The shift to online work has made video conferencing essential, leading to a significant increase in online presentations at conferences and meetings.

However, the excessive use of online meetings has increased fatigue among participants [4, 5]. Consequently, it has become common practice to record academic presentations for later review, allowing students and researchers to take time focusing on specific slides or sections of interest when they have the time [6].

While these recorded videos serve as valuable materials for review, the effective processing requires advanced technological interventions. *Natural Language Processing (NLP)* is a field of study that focuses on the interactions between computers and human language. It combines the power of artificial intelligence and computational linguistics to enable machines to understand, interpret, and generate human language in a way that is both accurate and natural. The advances in *NLP* have been driven by the developments of the new *Transformers* architecture [7, 8]. *Transformer*-based models have demonstrated state-of-the-art performance in various *NLP* tasks, including *audio-to-text* transcription [9, 10].

At the same time, sophisticated architecture alone is insufficient. In machine learning, it is well understood that achieving superior results relies on having both sufficient quantity and quality of data [11]. Building upon this principle, OpenAI's *Whisper* model leveraged massive-scale weak supervision to achieve state-of-the-art results in audio-to-text tasks. It demonstrates performance closely matching human professional transcribers in the English [12, 13]. These advancements provide the technological foundation necessary to address the challenges of reviewing recorded online meetings.

Despite the availability of recorded materials and advanced *AI* models, analyzing such recordings to extract specific information faces significant limitations [14]. Since the videos are often lengthy and contain both spoken narrations and visual contents, manual review becomes inefficient and time-consuming [15, 16]. Participants struggle to process all the information during live sessions, and reviewing raw recordings without navigational aids is equally laborious.

From a technical perspective, transforming these recordings into useful information is not a single task but a multi-layered challenge involving accessibility, flexibility, and contextualization.

The first hurdle lies in the practical accessibility of foundational models. Although state-of-the-art architectures like *Whisper* offer high accuracy, their utilization is often hindered by complex installation steps and dependency management that require intricate technical knowledge. Without a user-friendly interface, such as a web application, it remains difficult for general users to deploy and visualize these powerful machine learning models effectively [17].

Achieving accessible transcription is the initial step. The subsequent challenge lies in the flexibility of the *Information Extraction (IE)* tools used to process that data. While *AI* can capture key information [18], relying on existing proprietary systems often limits adaptability. These "black box" tools prevent researchers from customizing or integrating the latest open-source innovations. Consequently, constructing a system that effectively combines multiple models, such as *audio-to-text*, *summarization*, and *keyword extraction*, requires a modular approach that proprietary solutions typically do not support.

Finally, a critical limitation persists, the loss of multimodal context. Most automated methods process the entire recording as a single text sequence, merging different topics across various slides into one [19]. As a result, these pipelines often miss the contextual relationships between spoken narrations and the visual elements on the slides. Unlike manual review, where the viewer sees the slide while hearing the explanation, prior extraction-based pipelines generate detached summaries that are separated from the visual source. This separation forces users to mentally map the text back to specific visual elements, reducing the clarity and usability of the extracted information.

1.2 Contributions

To address the challenges of accessibility, system flexibility, and multimodal context, this thesis presents a design of *Audio-to-Text Conversion Software* and its applications. The objective is to develop an open-source, multimodal framework that integrates *Automatic Speech Recognition (ASR)*, *summarization*, *Optical Character Recognition (OCR)*, and *Large Language Models (LLMs)* to help users review and comprehend recorded videos of online meetings and presentations more efficiently. The research is structured into three main contributions.

1.2.1 Audio-to-Text Conversion Software

The first contribution directly addresses the challenge of accessibility by implementing *Audio-to-Text Conversion Software*. While high-performance models like *Whisper* exist, they often lack the usability required for widespread adoption. In this contribution, I design and implement an *Audio-to-Text Conversion Software* using OpenAI's *Whisper* models, focusing on bridging the gap between raw model performance and end-user utility.

Specifically, this study involves the development of an accessible interface by merging the *Whisper* model with a user-friendly software tool using the *Streamlit* web application framework. This implementation effectively eliminates complex installation dependencies for the end user. Furthermore, the research identifies the optimal *Whisper* models by systematically comparing their transcription speed, accuracy, and resource consumption to find the best balance for real-time application. Finally, the study establishes a reliable foundation for subsequent data processing stages by evaluating the stability and response time of the web-based deployment under varying loads.

1.2.2 Meeting Minutes Generation Tool

Building upon the transcription foundation, the second contribution addresses the challenge of flexibility and information overload. Recognizing that raw transcripts are often too lengthy for efficient review, this contribution develops a fully open-source meeting minutes generation tool with modular workflow that can match proprietary black box systems in generating concise meeting minutes.

This contribution entails designing a modular workflow that integrates distinct open-source components, including scene-change detection, audio-to-text conversion, and summarization. It involves a systematic selection of the most appropriate open-source models, such as *BART* for summarization, to ensure the system remains adaptable and transparent. The effectiveness of this approach is demonstrated through a comparative evaluation, which shows that the proposed open-source integration achieves higher accuracy in transcription and competitive performance in summarization and keyword extraction when compared to existing proprietary commercial systems.

1.2.3 Slide Annotation Generation Tool

The third contribution addresses the critical limitation of multimodal context loss. To overcome the problem of "detached summaries" identified in the previous workflow, where text is separated from visual aids, this contribution implements a *slide annotation system* that performs robust multimodal analysis to anchor information directly to the slides.

A key aspect of this contribution is the development of a hybrid two-stage segmentation strategy that prioritizes *OCR* for precision while employing a visual change fallback to handle occlusions and ensure accurate temporal alignment. Additionally, the system integrates a multimodal *LLM* to generate concise annotations that are spatially anchored to specific visual elements, effectively fusing the spoken narration with the visual slide content. The value of this integrated approach is validated through user studies, which confirm that the system significantly streamlines the information retrieval process compared to standard video playback or text-only summaries.

1.3 Contents of This Dissertation

The remainder of this thesis is organized to present the progression of the research from foundational technologies to advanced multimodal applications.

Chapter 2 presents the Literature Review and Theoretical Background, establishing the academic context for the research. It covers the evolution of *Automatic Speech Recognition (ASR)* systems, the development of *Natural Language Processing (NLP)* techniques for *summarization*, and the recent advancements in *Large Language Models (LLMs)* and multimodal analysis. This chapter also discusses existing methodologies for meeting analysis and identifies the specific gaps that this thesis aims to address.

Chapter 3 details the first contribution of the thesis, *Audio-to-Text Conversion Software*, focusing on the accessibility of audio-to-text technologies. It describes the design and implementation of a web-based application that integrates the OpenAI *Whisper* model, designed to lower the barrier of entry for using state-of-the-art transcription tools. The chapter presents the system architecture, the methodology for model selection, and the evaluation results regarding transcription accuracy, speed, and system stability under load.

Chapter 4 presents the second contribution, meeting minutes generation tool, which addresses the challenge of processing lengthy transcripts into usable meeting minutes. This chapter intro-

duces a meeting minutes generation tool built upon a modular, fully open-source workflow that integrates *ASR* with *summarization* and *keyword extraction*. It details the comparative analysis of various open-source models against proprietary systems and evaluates the quality of the generated minutes through both quantitative metrics and user feedback.

Chapter 5 introduces the third contribution, the *slide annotation generation tool*. This chapter describes the advanced methodology used to overcome the loss of context in detached summaries. It explains the hybrid two-stage segmentation strategy for slide detection and the implementation of a multimodal *LLM* to generate context-aware annotations. The chapter concludes with a comprehensive evaluation involving technical performance benchmarks and a user study assessing the system's usability and effectiveness.

Chapter 6 presents the discussion and comparative analysis of proposed frameworks. This chapter synthesizes the findings across the developed software and tools to demonstrate the evolution of the research. It provides a comparative analysis of the modular workflow versus the integrated multimodal approach, discussing the trade-offs between component flexibility, computational efficiency, and contextual precision.

Finally, Chapter 7 concludes the thesis by summarizing the key contributions and limitations of the research. It also outlines potential directions for future work, suggesting pathways for further refining automated annotation tools and expanding their application in digital communication.

Chapter 2

Literature Review and Theoretical Background

2.1 Overview

This chapter establishes the theoretical framework for the proposed Audio-to-Text Integration system by reviewing the diverse bodies of literature that underpin this thesis. The review is organized into three primary thematic areas, textual analysis, visual processing, and multimodal integration.

First, surveys were conducted on *Audio-to-Text* conversion and *Speech Recognition* alongside *Text Summarization* techniques and *Keyword Extraction* methods to identify effective strategies for handling spoken language.

Second, investigations were conducted on *Optical Character Recognition (OCR)* and Video Processing and Slide Segmentation to understand how to accurately align temporal video data with static slide content.

Finally, examinations were conducted on the emergence of *Multimodal Large Language Models (LLMs)*, discussing their potential to overcome the limitations of modular pipelines by unifying text and image processing.

2.2 Audio-to-Text Conversion and Speech Recognition

Audio-to-text conversion, formally known as *Automatic Speech Recognition (ASR)*, is the process of decoding audio signals into textual data [20]. While various architectures have been employed for this task, *Transformer*-based models have recently become dominant in the *Natural Language Processing (NLP)* field. This success is largely attributed to the transformer's *attention mechanism*, which captures unique information and models relationships between words across long-range dependencies without the significant training slowdowns associated with previous architectures [8].

To further enhance *ASR* performance, researchers have proposed hybrid and optimized transformer architectures. For instance, Karita et al. [21] combined *Transformers* with *Recurrent Neural Networks (RNNs)* using *Connectionist Temporal Classification (CTC)*, successfully reducing the *Word Error Rate (WER)* on standard benchmarks like Wall Street Journal [22, 23] and TED-LIUM [24].

Similarly, Shi et al. [25] introduced *Weak-Attention Suppression (WAS)* to improve robustness against long, noisy inputs, achieving consistent *WER* reductions on the LibriSpeech benchmark [26].

Parallel to these architectural optimizations, self-supervised learning has emerged as a powerful paradigm. Models such as *Wav2Vec 2.0* [27] and *HuBERT* [28] leverage unlabeled data to perform effectively even for low-resource languages. Building on this, the *Massively Multilingual Speech (MMS)* model scales self-supervised learning to support thousands of languages, though this extensive scope incurs higher computational costs [29].

Among these state-of-the-art approaches, the *Whisper* model, proposed by Radford et al. [30], stands out for its robustness in multilingual and noisy environments. Trained on a massive dataset of 680,000 hours of weakly supervised audio data, *Whisper* demonstrates capabilities comparable to professional human transcribers, achieving *WER* scores of 8.81% and 7.61% respectively.

While it occasionally suffers from "hallucinations" that generates incorrect words or phrases [30], its ability to handle diverse acoustic conditions makes it a superior candidate for real-world applications. Consequently, to ensure state-of-the-art transcription accuracy, this study employs the *Whisper* transformer-based architecture for the proposed audio-to-text web application.

2.3 Text Summarization Techniques

Text summarization is generally categorized into extractive and abstractive approaches [31]. Extractive summarization identifies and concatenates significant sentences from the source text without modification. While this method preserves the original wording, it often requires iterative manual processing to ensure the final output flows coherently.

In contrast, abstractive summarization generates novel words, phrases, and sentence structures to convey the essence of the original text, mimicking human writing styles [32]. Comparative evaluations highlight the superior capability of abstractive methods for producing cohesive summaries. For instance, Giarelis et al. [33] demonstrated that abstractive models consistently outperform extractive approaches on standard metrics such as *ROUGE-n* and *BLEU*. Consequently, to improve the comprehension of voice record transcriptions without requiring additional human intervention, this study utilizes abstractive summarization.

Recent advancements in abstractive summarization rely heavily on *Transformer*-based architectures. Early approaches utilized general-purpose language models. For example, Ramina et al. [34] employed *BERT* for topic-level summaries, while Goloviznina et al. [35] demonstrated that *T5* outperformed GPT-3 on specific language datasets. However, general-purpose models can sometimes result in sub-optimal performance on specialized tasks compared to dedicated architectures.

To address this, *Google* introduced *PEGASUS*, a model designed explicitly for abstractive summarization which typically achieves state-of-the-art results [36]. However, its specialized nature can limit generalization across unseen topics.

In contrast, the *Bidirectional and Auto-Regressive Transformers (BART)* model strikes a balance between accuracy and generalization [37]. Shiraly [38] demonstrated that fine-tuned *BART* models achieve performance comparable to premium *GPT-3* models but with significantly lower implementation costs. Given the diverse range of topics covered in lecture recordings, *BART* was selected as the optimal architecture for this system due to its proven robustness and cost-efficiency [39, 40].

The application of automatic summarization to educational content has been widely explored

as a means to enhance learning efficiency. Haz et al. [41] applied abstractive summarization to automatically produce localized minutes for each slide, capturing detailed information. Similarly, Gonzalez et al. [42] employed *GPT-3* to summarize lecture videos, reporting that participants who accessed these summaries achieved better learning outcomes. More recently, multimodal pipelines like *SlideSpecs* [43] and *Retrieval-Augmented Generation (RAG)* frameworks have been proposed to extract text and visual elements for generating comprehensive lecture notes [44].

While these systems effectively generate summaries, a critical limitation remains. These systems predominantly function as extraction pipelines that output results as detached text reports, separate web pages, or side-by-side video interfaces. They lack ability to write generated insights directly back into the original working document. This disconnection forces users to constantly switch contexts between the summary and the source material. This work addresses this gap by generating summaries and anchoring them spatially within the original PowerPoint file as comments, transforming the static presentation into a self-contained, interactive learning artifact.

2.4 Keyword Extraction Methods

Keyword Extraction (KE) is the automated process of identifying the most representative words or phrases in a document to provide a concise summary of its content. Approaches to this task are generally categorized into statistical, embedding-based, and graph-based methods, each with distinct trade-offs between computational efficiency and semantic accuracy.

Statistical methods, such as *Rapid Automatic Keyword Extractio (RAKE)* and *Yet Another Keyword Extractor (YAKE)*, rely on word frequency and co-occurrence patterns. Maylawati et al. [45] demonstrated the utility of *RAKE* in chatbot applications for quick question matching, while Ramachandran et al. [46] found that *YAKE* achieved superior similarity values among statistical methods for document clustering. These algorithms are highly efficient and require no training data, but they often lack the contextual depth needed to capture complex semantic relationships [47, 48].

To address the lack of context, embedding-based methods utilize *Transformer* architectures. Khan et al. [49] evaluated *KeyBERT*, which employs *BERT* embeddings to rank candidate keywords based on their cosine similarity to the document embedding. Their results showed that *KeyBERT* significantly outperformed statistical baselines by capturing semantic nuance. However, this accuracy comes at the cost of high computational resource requirements [50].

Graph-based approaches, such as *TextRank* [51], model text as a network where edges represent co-occurrence. While effective, standard *TextRank* can struggle with frequent but semantically unimportant words.

Xiong et al. [54] proposed improving this by fusing *TextRank* with *BERT* semantic clustering, which outperformed standard unsupervised methods. Building on this graph-theoretical approach, Skrlj et al. [52] introduced *Rank-based Keyword Extraction via Unsupervised Learning (RaKUn)*, which utilizes meta-vertices and redundancy filters to improve ranking.

The updated version, *RaKUn2*, combines this graph-based structure with unsupervised learning to cluster similar words and apply load-centrality for ranking [53]. This mechanism significantly reduces the graph size, offering a scalable solution that balances the contextual accuracy of embeddings with the speed of statistical methods.

Given the diverse and technical nature of lecture content, a model that balances efficiency with semantic precision is required. Therefore, this study employs *RaKUn2* [52], as its hybrid graph-based approach is well-suited for managing the complexity of presentation transcripts without the excessive computational overhead of fully generative models.

In educational contexts, extracted keywords serve a critical function beyond simple summarization. They act as cognitive anchors that assist in indexing and retrieving specific information within dense lecture materials. When these textual cues are aligned with visual elements, they function as annotations that significantly enhance user comprehension [55].

Traditionally, generating these annotations relied on specialized extraction pipelines [33]. However, recent advances in *Large Language Models (LLMs)* have enabled unified systems capable of producing both summaries and reflective keywords simultaneously [56, 57].

Several studies have applied these capabilities to lecture videos. For instance, multimodal pipelines like *SlideSpecs* [43] and *RAG* frameworks [44] extract text and visual elements to generate searchable indices or comprehensive lecture notes.

Despite these advancements, a critical gap remains in how this information is presented to the learner. Existing systems predominantly function as extraction pipelines, outputting results as detached text reports, separate web pages, or side-by-side video interfaces. They lack the ability to write generated insights directly back into the original working document. This disconnect forces users to switch contexts between the keywords and the source material. This work addresses this limitation by using extracted keywords to spatially anchor comments within the original PowerPoint file, thereby transforming the static presentation into a self-contained, interactive learning artifact.

2.5 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is the process of converting visual text data from images or documents into machine-encoded text using pattern recognition techniques. This digitization enables systems to extract, index, and process textual information contained within visual media, transforming static pixels into searchable data [58].

Recent advancements in *OCR* leverage deep learning architectures, with performance often depending on the specific balance between accuracy, speed, and computational resource requirements. Three prominent open-source models are commonly evaluated in the literature.

EasyOCR utilizes a hybrid architecture combining *Convolutional Neural Networks (CNNs)* for feature extraction and *Long Short-Term Memory (LSTM)* networks for sequence modeling. It is designed to be lightweight and accessible, making it a popular choice for rapid prototyping, though it typically requires GPU acceleration to achieve optimal inference speeds [58].

In the domain of deep learning-based *OCR*, *PaddleOCR* is often noted for its robustness in handling complex scenarios. It uses a multi-stage architecture that excels at detecting text with irregular orientations, compression artifacts, or diverse stylistic compositions. However, this comprehensive approach involves a more complex pipeline and higher computational overhead compared to traditional engines, which can be a limiting factor for real-time or resource-constrained applications [59].

Conversely, *Tesseract*, maintained by *Google*, remains the industry standard for document analysis. Unlike models optimized for "natural" scene text, *Tesseract* is highly optimized for structured text layouts found in documents and presentations. While it historically relied on heuristic patterns, recent versions have integrated *LSTM*-based neural networks to improve accuracy. It is particularly valued for its computational efficiency and ease of integration on CPU-based environments, provided the input images are sufficiently pre-processed to ensure clarity [60].

For the application of slide analysis, the choice of *OCR* engine involves weighing the robustness of deep learning models against the efficiency of established document standards. While

PaddleOCR offers powerful handling of complex visual artifacts, *Tesseract* provides a mature, low-latency solution that is often more than sufficient for the structured, horizontal text typically found in educational presentations. Consequently, both engines present viable pathways depending on the system's specific prioritization of raw accuracy versus processing speed.

2.6 Video Processing and Slide Segmentation

Slide segmentation is a specialized subset of video shot boundary detection, where the objective is to identify temporal points where significant visual shifts occur between frames [61]. Fundamental methods for fragmenting a video rely on analyzing differences in low-level visual features, such as pixel intensity, color distribution, and edge stability. These approaches are broadly categorized into pixel-based and structure-based methods [62].

Pixel-based metrics, such as *Mean Square Error (MSE)* and *Color Histogram* comparisons, serve as widely used baselines. They detect transitions by aggregating pixel-level differences or comparing color distributions [63]. While these algorithms are computationally efficient, they are highly sensitive to noise and geometric alterations. Minor changes, such as lighting variations or compression artifacts, can trigger false positives, making them unreliable for robust segmentation [64, 65].

In contrast, structure-based approaches focus on the spatial arrangement of edges, shapes, and textures, which better preserves perceptual information even after video compression [66]. The *Structural Similarity Index Measure (SSIM)* assesses image quality based on human visual perception rather than absolute pixel differences [67].

Edge-based methods further refine this by tracking the stability of edge maps such as by using *Canny* edge detection to distinguish meaningful scene changes from minor fluctuations [68]. Given that this system processes compressed presentation videos where geometric integrity is preserved but compression artifacts may exist, *SSIM* provides a more robust metric for visual change detection than simple pixel differencing.

While global visual metrics are effective for general video, the specific domain of educational lecture videos presents unique challenges that limit the effectiveness of purely visual approaches. Deep learning models have attempted to address these complexities by learning semantic transition features. Sindel et al. [69] employed *CNNs* to detect slide transitions, yet the approach yielded relatively low F_1 scores. This highlights a key limitation of supervised learning in this domain, models trained on specific lecture styles often struggle to generalize to unseen scenarios or variable slide layouts.

Similarly, Yuan and Zhang [125] proposed detecting changes via *color clustering in small regions (CCSR)*. While efficient, this method remains sensitive to localized color shifts caused by animations or pointer movements.

Furthermore, "talking head" presenters often occlude parts of the slide, and their movements, along with mouse pointers or transition animations, can trigger false positives in both *SSIM* and *Histogram-based* detectors.

Conversely, static slides accompanied by long audio explanations may result in false negatives where no transition is detected despite a topic change. Early research by Che et al. [70] attempted to solve this using global *OCR*, but achieved low segmentation accuracy ($< 50\%$) due to the difficulty of extracting text from noisy full-frame video.To overcome these limitations, this works utilize a hybrid strategy. We employ a domain-specific *Region of Interest (ROI)* based *OCR* to track explicit slide numbers as a semantic ground truth, while utilizing *SSIM* as a robust visual fallback

for scenarios where slide numbers are absent or unreadable.

2.7 Multimodal Large Language Models

The integration of visual, auditory, and textual information to construct a unified context lies at the core of multimodal understanding [71]. In the educational domain, the importance of this integration has grown considerably, driven by the shift toward online learning, the proliferation of digital course materials, and the emergence of multimodal *Large Language Models (LLMs)* [72].

Recent research has demonstrated that leveraging these diverse modalities significantly enhances the interpretation of instructional content. Singh et al. [73] proposed utilizing multimodal signals to refine the segmentation of lecture recordings, thereby improving the review experience for students. Similarly, Lee et al. [74] introduced *PolyViLT*, a model designed for cross-modal retrieval between textual and visual elements, which achieved superior performance over earlier uni-modal approaches.

A key challenge in this field is managing the retrieval of relevant context for LLM generation. Wright et al. [75] applied multimodal *RAG* to large-scale repositories of digital textbooks and classroom data, confirming that retrieving specific context significantly improves the accuracy of LLM-generated responses.

However, this approach introduces trade-offs regarding complexity and quality. Li et al. [76] reported that when the available context is sufficiently small to fit within the model's context window, the RAG mechanism may introduce unnecessary architectural complexity and can even degrade output quality due to retrieval errors.

Consequently, in scenarios where the input data, such as a single lecture transcript and its associated slides, fits entirely within the model's capacity, full-context or long-text prompting proves to be a more effective and reliable strategy than *RAG* [77].

2.8 Summary

This chapter reviewed the foundational technologies required for automating lecture annotation, establishing a specific technological stack that balances accuracy with computational efficiency. For extracting semantic content from speech, *Whisper* was identified as the robust standard for *ASR*, while *BART* and *RaKUn2* were selected for abstractive summarization and keyword extraction due to their superior generalization capabilities.

In the visual domain, the literature supports a hybrid segmentation approach combining *SSIM* and *OCR* (PaddleOCR/Tesseract) to overcome the limitations of pixel-based metrics in compressed video. Furthermore, recent findings in multimodal analysis suggest that for lecture-specific tasks, full-context prompting offers a more reliable alternative to complex *RAG* architectures.

However, a critical gap remains, that is the existing systems function primarily as extraction pipelines that produce detached reports. This thesis addresses that disconnect by integrating these components to generate annotations directly within the original PowerPoint slides, thereby transforming static presentations into self-contained, interactive learning artifacts.

Chapter 3

Design and Implementation of Audio-to-Text Conversion Software Using Whisper Models

3.1 System Overview

This system is built to explore the potential of utilizing the *Whisper* models to create an audio transcription through a web application. To achieve it, the inputs of audio and the *Whisper* models from OpenAI are integrated into a web application as seen in Figure 3.1.

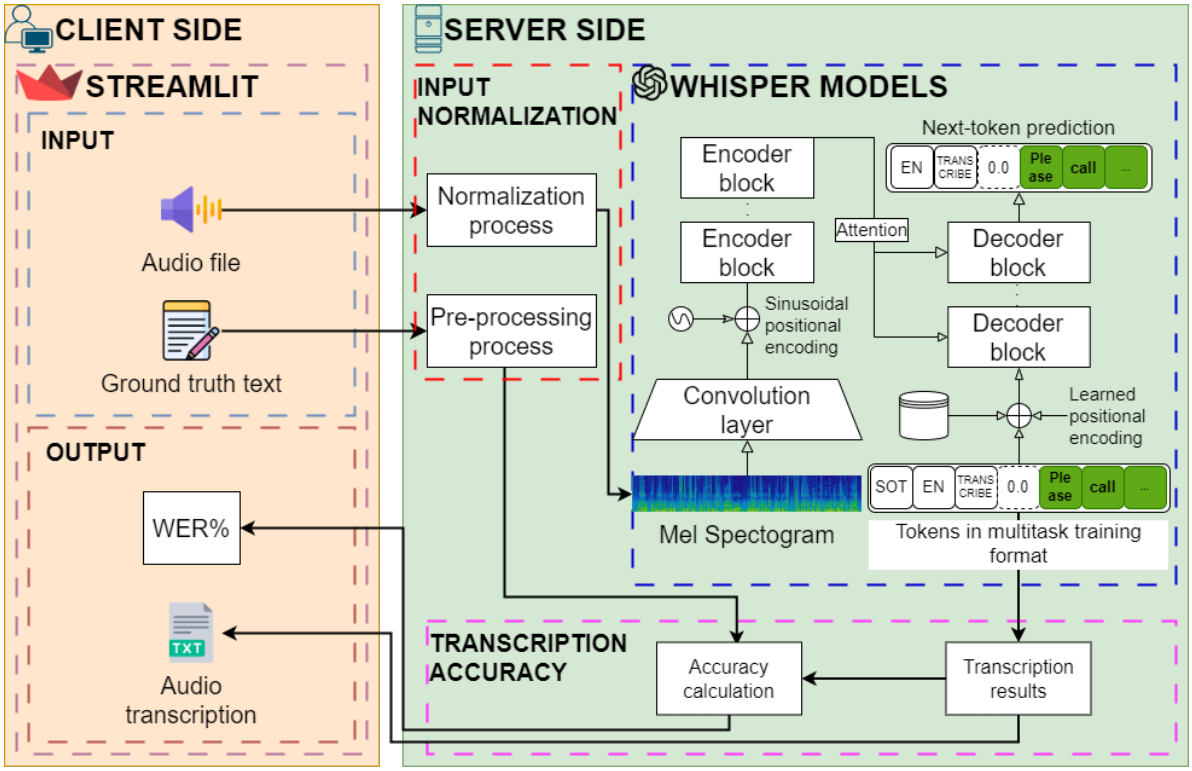


Figure 3.1: System Design of Audio-to-text Conversion Software.

The system is divided into three parts, input, server side, and output. The system takes audio and text as the input. The files are transferred into the systems using a file upload method. Com-

mon audio file formats such as *.mp3*, *.wav*, and *.flac* are acceptable. The server side of the web application functions as the hosting environment where the system resides.

Inside the server side, there are three parts input normalization, *Whisper* models, and transcription accuracy. Input normalization process for each uploaded file to fit the model's requirement. *Whisper* models take the normalized audio file and transcript the text. Transcription accuracy takes the transcripted text and compares it with the pre-processed input text by calculating the WER%. The web application produces transcribed text output and a WER% score for audio transcription.

3.1.1 Streamlit

Streamlit is a *Python* package that offers a streamlined approach to front-end web application development [78]. Its user-friendly API and automated rendering capabilities facilitate the rapid creation of interactive user interfaces without requiring expertise in HTML, CSS, or JavaScript.

With seamless integration with popular data science libraries, *Streamlit* empowers researchers and developers to effortlessly incorporate data visualizations, perform complex calculations, and visualize machine learning models. The framework's inherent reactivity enables real-time updates, ensuring a dynamic user experience. Furthermore, Streamlit provides robust functionality for handling diverse forms of user input, such as text input and file uploads.

3.1.2 Input

As a web application, the input is on the client side. The input receives audio files in *.mp3, .wav, .flac,* or other acceptable *FFMPEG* audio formats. The input is also able to receive text input. The text serves as the ground truth of the audio spoken. This text is used to calculate the accuracy of the audio transcription later on the output.

3.1.3 Server Side

As a web application, all processing and computational concentrations are on the server side. Server-side is divided into three parts: input normalization, Whisper models, and transcription accuracy.

3.1.3.1 Input Normalization

The input normalization component handles the uploaded files from the client side and processes them to fit the requirements of the Whisper models and transcription accuracy [79]. Many audio files are recorded on different device settings, including sampling rates. To ensure compatibility with the Whisper models, a normalization process is applied to the audio before being processed into the models. The Whisper models expect input in the form of 16,000 Hz audio samples and mono-channel audio format.

For sampling audio with a rate of 48,000 Hz, a down-sampling process is applied to the audio input. In the case of audio samples with a rate of 8,000 Hz, a re-sampling process is needed to increase the sample rate to 16,000 Hz. This process involves interpolating the existing samples to create new samples. For audio in dual-channel format, where there are two separate audio channels, both channels need to be merged into a single mono-channel audio. This ensures that the input format matches the transcription process requirements for the Whisper models.

In order to align the format of text files containing spoken text with the results of audio transcription, additional pre-processing steps are required. This step consists of lowercasing, symbol removal, and punctuation removal. Lowercasing converts all words to lowercase, ensuring that variations in capitalization are disregarded during analysis and comparison. Symbols removal means symbols such as @, #, $, %, and others are removed from the text. Punctuation removal means punctuation marks such as periods, commas, question marks, and exclamation marks are eliminated from the text. While these marks serve grammatical purposes, they are not essential for analyzing the content of the spoken text. By applying these pre-processing steps, the spoken text is transformed into a consistent format that aligns with audio transcription results.

3.1.3.2 Whisper Models

The audio transcription using *Whisper* models is stored inside the server side. *Whisper* is built on the classic transformer architecture, employing encoder and decoder blocks with attention mechanisms. It processes audio recordings in 30 s chunks, encoding the audio using the encoder section and saving word positions. The decoder predicts tokens, representing individual words, leveraging the encoded information and previously predicted words to generate the next word.

Whisper models are trained on over 600,000 hours of multilingual and multitask supervised data from the web. *Whisper* is a large and general audio model. It achieves robustness by learning from diverse sources of labeled audio data. Some of the labeled audios were transcribed using machine learning models instead of human transcribers. While the data imperfections may reduce precision, they contribute to the model's robustness compared to purely human-curated datasets

3.1.3.3 Transcription Accuracy

The common audio-to-text metric measurement uses WER%. WER% is the metric used to measure the accuracy of *ASR* systems. It calculates the percentage of incorrect words in the transcribed output compared to the reference or ground truth. The WER% is calculated by dividing the total number of word errors (insertions, deletions, substitutions) by the total number of words in the reference, and multiplying the result by 100 as shown in Equation 3.1 [80]. A lower WER% indicates higher accuracy, 0% WER% indicates a perfect match between the transcribed output and the reference.

$$WER\% = \frac{S_w + D_w + I_w}{N_w} \times 100 \tag{3.1}$$

Where:

S_w is the number of substitute words

D_w is the number of deleted words

I_w is the number of newly inserted words

N_w is the number of ground truth words

3.1.4 Output

Users can view the transcribed text on the web interface, allowing easy access and review of the content. The WER% score, computed by comparing the transcribed text with the ground truth

text, quantifies the accuracy of the transcription. This output enables users to quickly understand the audio content and evaluate the transcription's quality and reliability.

3.2 Experimental Results

In this section, the developed user interface of the web application is presented the experimental results conducted on the system are described.

3.2.1 User Interface

Figure 3.2 illustrates the *User Interface* (UI) from the developed web application. Once all the content is loaded, the user is presented with two upload file areas that serve as gateways for the user to upload the audio and ground truth text files. By uploading the ground truth file, the user can view additional information, including the transcription accuracy, the number of correctly transcribed words, the number of substituted words, the number of deleted words, and the number of newly inserted words. However, if the user doesn't upload the ground truth text file, users will only see the transcribed text from the audio.

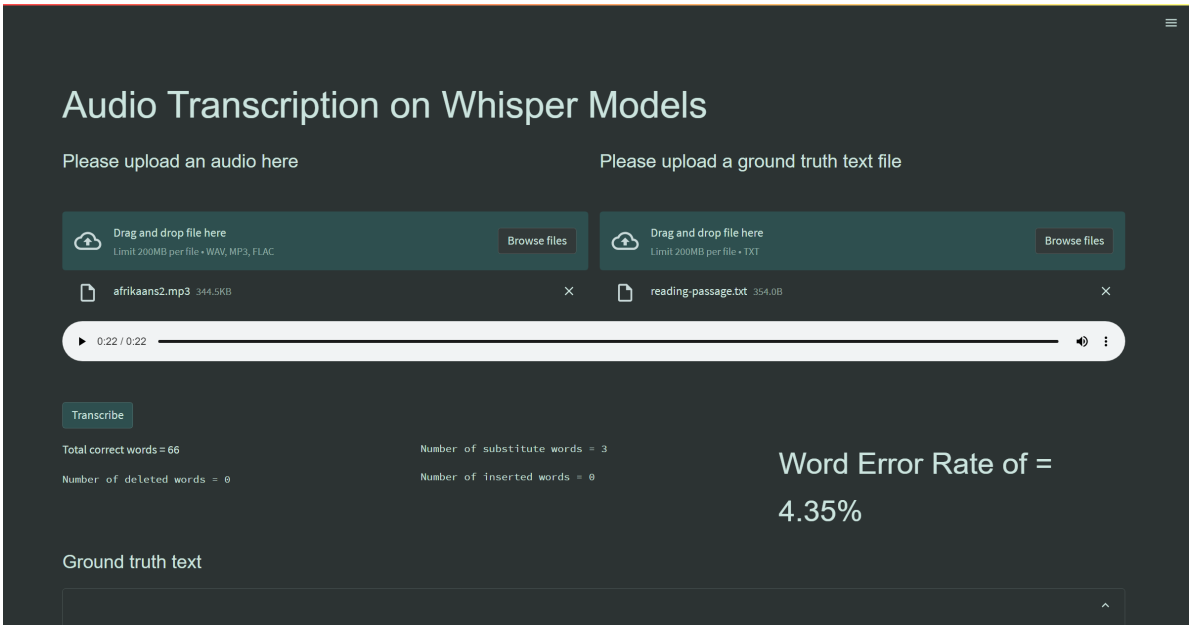


Figure 3.2: Audio-to-text Web Application User Interface.

3.2.2 Performance

Performance such as loading speed is the factor users measure when visiting a web application for the first time. To test the loading speed of the web application when all content is displayed, an extension called Lighthouse is utilized [81]. Lighthouse is the performance measurement tool developed by *Google* for developers. It enables the measurement of loading times for developed web applications on various popular web browsers. Figure 3.3 illustrates the loading time of the web application's content.

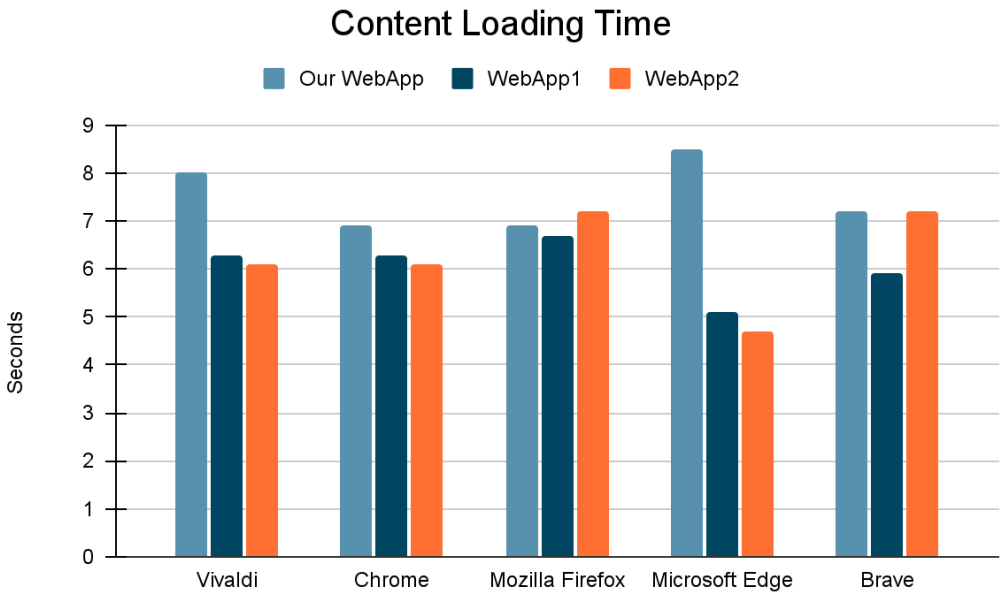
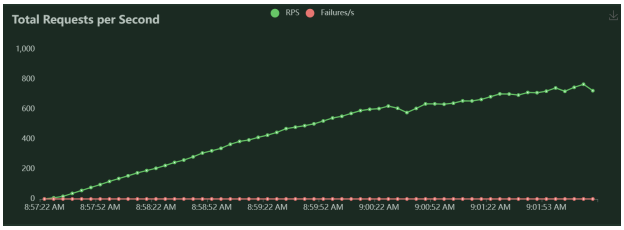


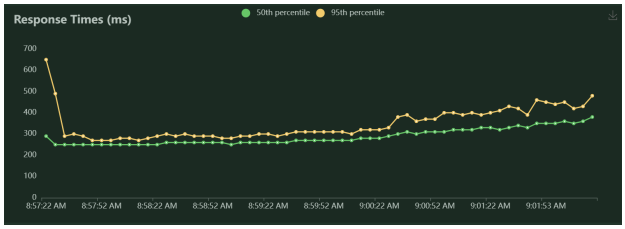
Figure 3.3: Duration of Content Loading Time on Each Tested Browser.

From Figure 3.3, it observed that each browser takes less than 10 s to fully load the content. In comparison with other well-designed and developed web applications that utilize both front-end and back-end engineers, the web application developed using Streamlit has achieved comparable results.

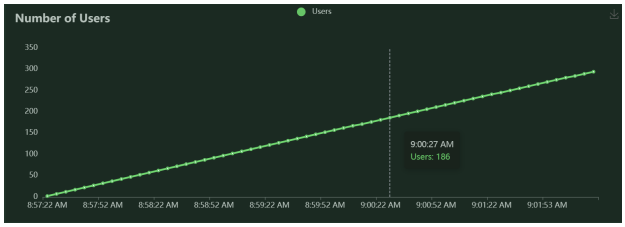
Furthermore, to simulate a stress load test on the web application while handling multiple users, the load testing tool called Locust is utilized [82]. Locust enables users to simulate hundreds of users concurrently, providing the means to measure the application's performance under heavy loads. The results of the Locust stress test are illustrated in Figure 3.4.



(a) Request per Second.



(b) Response Time (ms).



(c) Total User.

Figure 3.4: Locust Reading for 300 Users.

As observed from Figure 3.4, the average response time observed was 309 ms, with an average

of 471.5 requests per second. However, we noticed that beyond 186 users, the response time started to increase significantly, indicating that the web application's limit is around 180 users. This suggests the presence of a bottleneck that affects the performance of the application under heavy load.

3.2.3 Accuracy vs Speed

To evaluate the performance of the *Whisper* model and assess its potential use in web applications, measurements were made on transcription accuracy and transcription speed. Whisper models are classified into four categories based on their performance and parameters, indicated by their names. However, as the parameters increase, the duration of transcript audio also increases. Therefore, all four English audio transcription models are tested in the same environment on three different audio categories characterized by their duration before determining the best model to embed into the web application. The testing environment was done on the server side, the specification is described in Table 3.1.

Table 3.1: Server Specification.

Component	**Specification**
CPU	Intel(R) Xeon(R) Gold 5218
Core	4 @ 2.3 GHz
RAM	8 GB
GPU	Nvidia Quadro RTX 6000
VRAM	24 GB

The first audio category, *shortAudio*, consists of audio with a length of 23 s with a total of 69 words. The second audio category, *mediumAudio*, consists of audio with a length of 182 s with a total of 332 words. The third audio category, *longAudio*, consists of audio with a length of 374 s with a total of 965 words. This process aims to find a middle ground that achieves a desirable WER% while maintaining an acceptable transcription speed. Evaluation of the *Whisper* model on transcription accuracy and transcription speed is illustrated in Figure 3.5.

Based on the information provided in Figure 3.5, the *small.en* model demonstrates a desirable balance between transcription accuracy and transcription speeds. It achieves comparable results to the *medium.en* model across all three audio categories while significantly reducing the execution time. This makes the *small.en* model a rational choice for applications where an acceptable transcription accuracy is required without compromising on transcription duration. Alternatively, the *medium.en* model can be considered for applications that prioritize higher transcription accuracy at the expense of longer execution times. The results emphasize the state-of-the-art performance of Whisper models in the field of *Natural Language Processing*, specifically in the audio-to-text task.

3.3 Summary

This paper describes the integration of *Whisper* models for the audio-to-text task within a web application. While many audio-to-text models utilize unsupervised learning with the advantages of unlabelled datasets, Whisper models employ weakly labeled data in the dataset. This approach

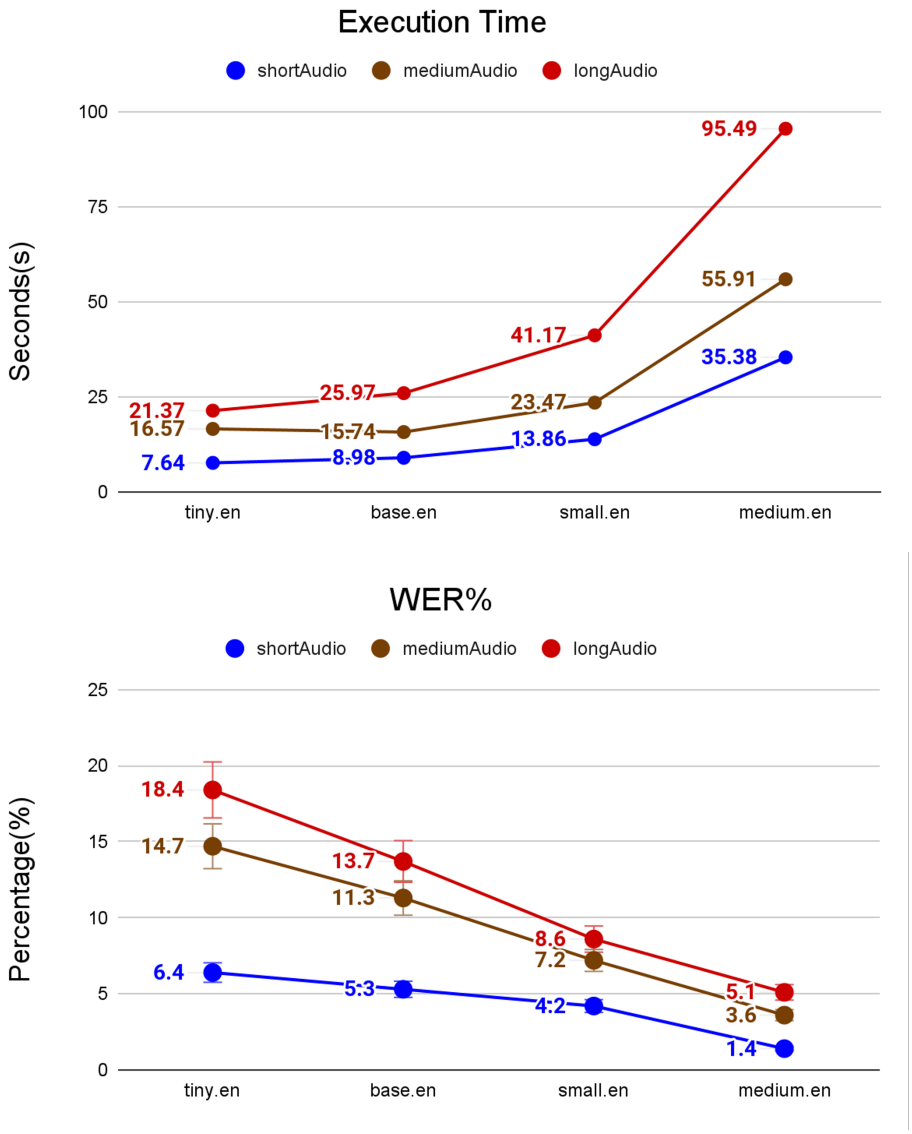


Figure 3.5: Execution Time and Word Error Rate of Four Whisper Models.

has enabled the Whisper models to achieve comparable results to those of professional human interpreters.

The development process of the web application was facilitated by the use of the Streamlit package, which allows the creation of *UI* layouts using code. The developed web application achieved a loading time of less than 10 s, which is comparable to web applications designed by both front-end and back-end engineers. The load simulation showed an average response time of 309 ms, with an average of 471.5 requests per second, and a maximum of 180 users. The *Whisper* models were tested on three audio categories based on audio duration before being used in the web application. It was found that the *small.en* model gives a good balance between transcription accuracy and transcription speed. The *small.en* model achieved the WER% of below 10% for all audio categories with half of the execution time of the *medium.en* model.

In future works, we will focus on improving the *Whisper* transcription accuracy on low-resource language or domain-specific fields. As stated by *Whisper*'s authors, *Whisper* is considered to be a large model that suffers from these particular problems.

Chapter 4

Development of a Meeting Minutes Generation Tool Using Open-Source Models

4.1 Review of Previous Study

Prior to the development of the fully integrated framework presented in this chapter, a preliminary investigation was conducted to identify the most effective open-source models for processing raw meeting transcripts. While the initial research established the utility of the *Whisper* model for transcription, the resulting text was often voluminous and unstructured, necessitating secondary processing layers for summarization and keyword extraction. Consequently, the primary objective of this preliminary study was to evaluate state-of-the-art transformer models and extraction algorithms to determine which architectures offered the best performance across varying meeting durations.

The investigation into summarization models revealed that performance is highly dependent on the correlation between the model's training data and the duration of the input audio. By categorizing inputs into short, medium, and long segments, the results demonstrated that models fine-tuned on the *SAMSum* dataset, specifically `philschmid/bart-large-cnn-samsum`, achieved superior *ROUGE* scores for shorter, informal dialogue segments. Conversely, for medium and longer recordings, models fine-tuned on the *DialogSum* dataset, such as `knkarthick/MEETING_SUMMARY`, significantly outperformed others. Notably, the latter achieved the highest *ROUGE-1* and *ROUGE-2* scores for long audio, suggesting that models trained on extensive dialogue corpora are essential for preserving context in lengthy meeting minutes. These findings established the heuristic used in this chapter: dynamically selecting the summarization model based on the transcript length.

Parallel to summarization, the study evaluated five keyword extraction algorithms, including statistical, graph-based, and embedding-based approaches, using Cosine Similarity metrics against human-annotated references. The graph-based algorithm *RaKUn2* demonstrated the most robust performance, achieving the highest similarity scores for both short and long audio categories (58.42% and 48.35%, respectively), while the *TextRank* algorithm proved most effective for intermediate lengths. The conclusions drawn from this preliminary study provided the architectural blueprint for the system detailed in this chapter. By validating that specific open-source models could rival proprietary performance when correctly matched to the input length, the study confirmed the feasibility of a fully open-source workflow and identified the specific "best-in-class" components for integration.

4.2 Proposal of Meeting Minutes Generation System

This section describe the methodologies workflow of the system.

4.2.1 System Overview

The proposed system aims to provide users with flexibility in its usage of open-source models and accessibility. For this purpose, the proposed system is implemented on top of multiple functions and deployed as a web application, as shown in the Figure 4.1.

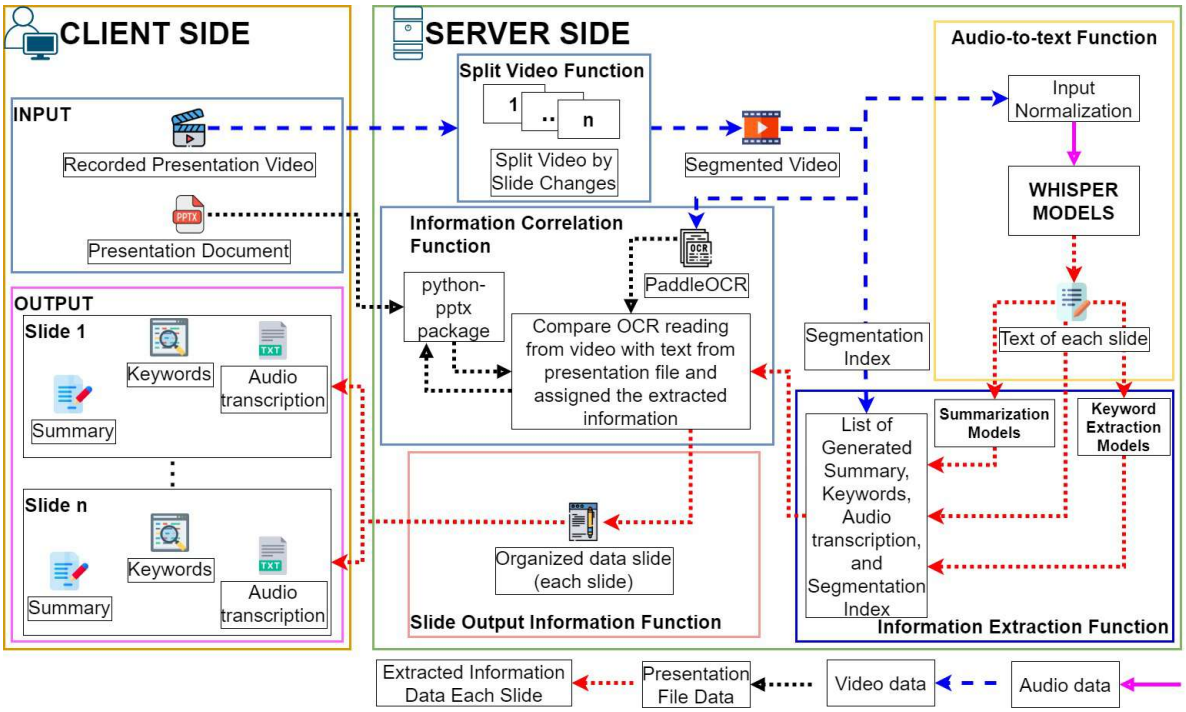


Figure 4.1: Overview of meeting minutes generation system.

This system architecture consists of the client-side and the server-side. The *client-side* comprises of the input function and the output function. The input function accepts a pair of the recorded online presentation and the document used during the presentation. The output function visualizes the results of the transcribed audio, generates summaries, extracts keywords, and matches slides through the *User Interface (UI)*.

The *server-side* is responsible for processing the video to generate meeting minutes. Since each slide in the video may contain different information, the system first segments the video by slides. This process uses the *split video function*, which includes the *scene change detection algorithm* and the *segmentation mechanism* [83]. The *scene change detection algorithm* compares visual information between adjacent frames and splits the video at the frame where visual information is changed.

Then, the audio from each segmented video is transcribed into a text using the *audio-to-text function* [84, 85]. This process converts the speaker's voice into a text format. From the text, the *information extraction function* generates the summary and extracts the keywords, helping users quickly grasp the key points of the speaker's message [39, 86].

At this stage, the system has generated the transcription, the summary, and the keywords for each video segment. However, these data still need to be linked to the corresponding slide. To

achieve this, the *information correlation function* is applied. This function uses *OCR*, the *python-pptx* library, and *regular expression (regex)* to match the extracted information with the appropriate slide [87]. Finally, the system organizes the correct information through the *slide output information function*, which is displayed to the user via the *client-side* interface.

4.2.2 Input Function

The presentation video contains both visual and audio data with a single presenter using screen sharing. The video format must be compatible with *ffmpeg*, and the resolution should be 1280×720. The presentation document provides the text content of the presentation, used as ground truth for validations. Currently, the document must be in *.pptx* format, with no images, shapes, or animations, and a single background color to minimize the complexity. Figure 4.2 shows an example of an acceptable document layout.

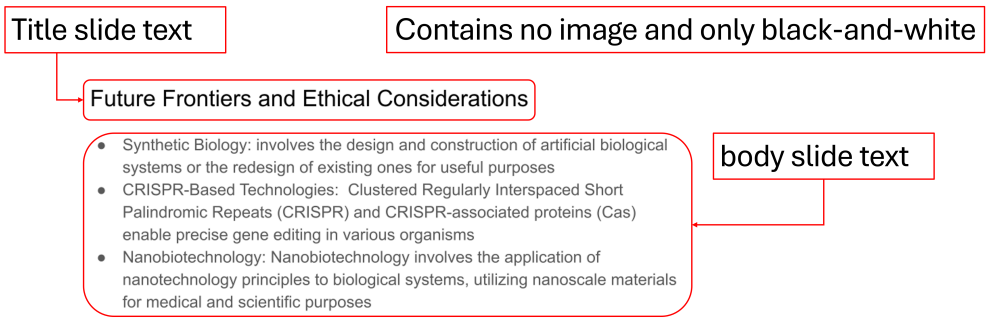


Figure 4.2: Acceptable document format

4.2.3 Split-Video Function

As explained above, since the presentation video may contain multiple slides, there is a possibility of information overlap between them. To prevent this, a scene change detection algorithm and segmentation mechanism are applied. Inspired by the work of Bulut F. et al. [88], we adopt the *split video function* to separate the input video into multiple segments using a *scene change detection* algorithm.

This algorithm detects differences between adjacent frames to determine segmentation points. Then, the segmentation mechanism splits the video with these segmentation points. Finally, the function produces segmented videos and segmentation indices. The segmentation index denotes the order of the segments within the input video and serves as the identifier for each segment. These segmented videos are then processed individually in the *audio-to-text function* to transcribe the audio. The segmentation index is attached to each produced extracted information in the *information extraction function*.

4.2.4 Audio-to-Text Function

This function employed the *audio-to-text* algorithm to convert audio signals into text. Before transcribing the audio from every segmented video, the normalization is performed. Resampling by

ffmpeg normalizes the video's sampling rate to match with the standard sampling rate of 16 kHz. Current *audio-to-text* models use the transformer architectures due to the efficiency of handling distant connections in audio data.

Then, pre-processing to convert the raw audio into a *Mel-spectrogram* is performed. *Mel-spectrogram* is a 2D representation of the audio's frequency content over time. Some models combine the convolutional feature encoder to extract patterns and reduce the data's complexity. The core component of the transformer architecture is the context model by the self-attention mechanism. It captures important connections across the entire audio sequence, allowing the model to process relations from short or long words sequence. The sequence order is defined by the positional encoding. Lastly, the postprocessing ensures the transcription is coherent, adding punctuation and making grammatical adjustments where needed. The process allows the transformer-based model to process entire sequences in parallel, improving transcription accuracy by focusing on different parts of the input at the same time.

4.2.5 Information Extraction Function

The *information extraction function* produces summaries, extracts keywords from the text, and gives the segmentation indices to them.

4.2.5.1 Abstractive Summarization

The *abstractive summarization* generates summaries while retaining original semantic contents of the source text [89]. In this study, *Bidirectional and Auto-Regressive Transformers Language Models (BART LMs)* is employed to achieve accurate results in generating coherent summaries [90]. This selection come from the results in our previous works to measure the performance of abstractive summarization models [39].

The *BART LMs* model is downloaded through the *HuggingFace* implementation [91]. The *BART LMs* model adopts the *bidirectional* encoder and the *auto-regressive* decoder. The *bidirectional* encoder allows the model to understand the contextual representations between words from before and after the current words in the transcribed text. It encodes these data into numerical vectors and heightened significant and flattened uninfluential data using an attention mechanism. The *auto-regressive* decoder enables the model to produce words by predicting the most potential words after the current words. This iterative process is performed continuously following the depth of the architecture. Finally, the model transforms these numerical vectors back into the text.

4.2.5.2 Keywords Extraction

The *keyword extraction* process automatically ranks and extracts the most relevant words from a source text. It determines the ranks based on word correlations, frequencies, and semantic meaning [92]. The *Rank-based Keyword Extraction via Unsupervised Learning (RaKUn2)* model is downloaded through *GitHub* repositories [93]. In this study, *RaKUn2* is employed due to its ability to deliver fast and accurate results across different keyword datasets [**?**]. This selection aligns with the findings of our previous work, where various keyword extraction methods were evaluated and the effectiveness of *RaKUn2* in diverse scenarios was confirmed [39].

The keyword extraction divides each sentence from the transcribed text into words and characters called *tokens*. This process is called *tokenization*. The relationship between tokens is determined by the frequency of their co-occurrence. It was calculated based on the frequency of

appearance of each token in a sentence. Then, it creates a graph where the tokens are the nodes, and the edges are the co-occurrence frequencies. It organizes the tokens' ranks based on the graph's number of edges. Finally, it performs the post-processing steps to remove the duplicated keyword candidates.

4.2.6 Information Correlation Function

The *information correlation function* compares and validates the text data from the recorded presentation video and the extracted information with the original text from the presentation document. This function workflow is visualized in Figure 4.3.

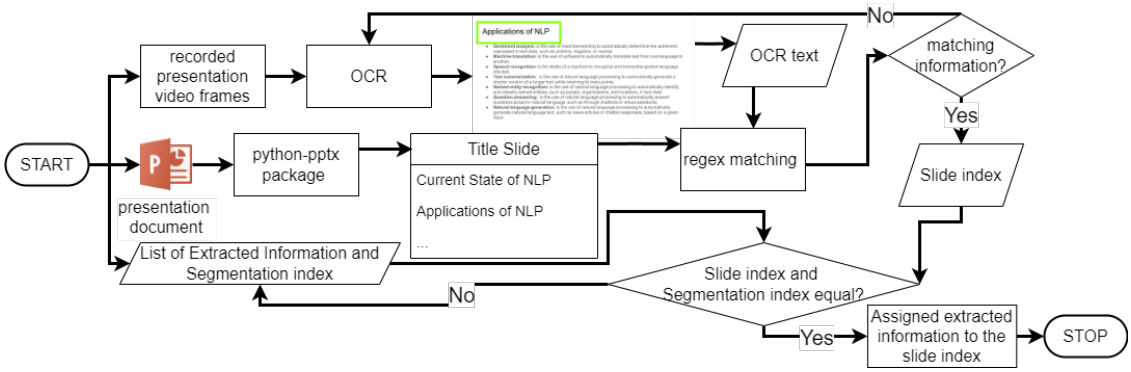


Figure 4.3: Workflow for *information correlation function*.

First, it recognizes the text from each frame in the video using *OCR*. Since the video was resized, only the text in the title area is recognized and extracted. For the comparison, the original text is gathered from the title text of each slide in the presentation document using the *python-pptx* package.

Then, it compares the *OCR* text with the original text using the *regex* string matching to determine which slide index the *OCR* text corresponds to. The slide index refers to the order of slides that appear in the presentation document. This process determines the exact identical strings that appear in both *OCR* text and the original text.

Finally, after the slide index is known, the extracted information is assigned to the similar segmentation index. This process ensures that the information extracted from the recorded presentation is accurately correlated with the corresponding slides, thus improving the overall accuracy and usefulness of the extracted information and avoid confusion for the readers.

4.2.7 Slide Output Information Function

Following the validation of data pairs by the *information correlation function*, the data needs to be organized for easy selections and managements. The data includes the extracted information, the *WER* score for audio-to-text conversion [94], the *CER* score for *OCR* accuracy [95], and the image for each slide. This function arranges the data in *JSON* format to ensure efficient handling and presentation. Figure 4.4 shows an example of the organized data for each slide, ready for display through the *UI* on the client-side.

In Figure 4.4, the *slide_index* indicates the order of slides. The *slide_image* represents the visual information shown in the recorded video. The *slide_ocr_text* shows the results of *OCR*. The *slide_presentation_text* shows the original text in the slide. The *slide_cer* display the validation

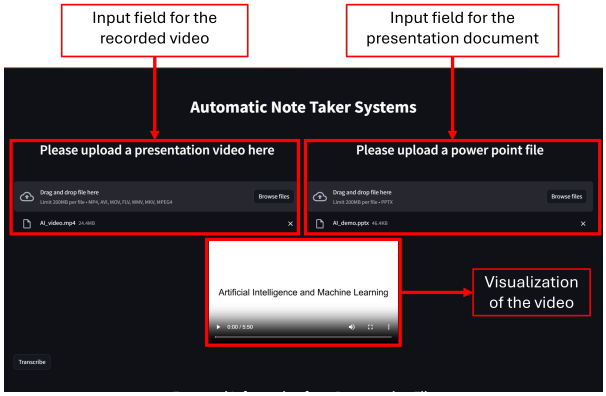
```
{
  "organized_data_slide_1": {
    "slide_index": "1",
    "slide_image": "slide_1.png",
    "slide_ocr_text": "machine learning explained",
    "slide_presentation_text": "machine learning explained",
    "slide_cer": "0.0",
    "slide_summary": "machine learning is a branch of artificial
      intelligence that...",
    "slide_keywords": "['machine learning', 'involves teaching',
      'artificial intelligence']",
    "slide_convert_text": "...The process of machine learning involves
      teaching computers to learn from data and improve their
      performance",
    "slide_wer": "4.7"
  },
  "organized_data_slide_2": "..."
}
```


Figure 4.4: Sample of organized data for each slide.

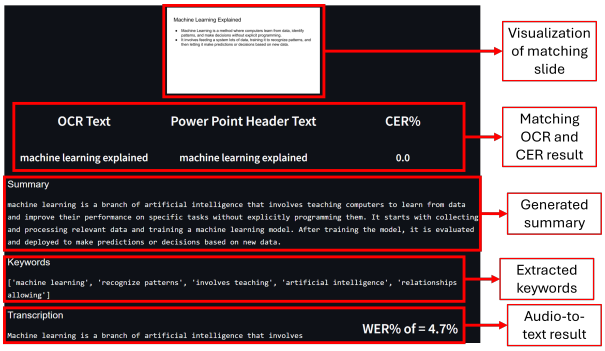
results of information correlation processes. The *slide_summary* and *slide_keywords* provide the brief explanation of the information. The *slide_convert_text* displays the transcripted text from *audio-to-text* process. The *slide_wer* indicates the accuracy of *audio-to-text*.

4.2.8 Output

The *output* displays the structured data in a web application using the *Streamlit* Python packag [78]. *Streamlit* integrates with the react *user interface* framework to develop web applications. Figure 4.5a shows the UI where users upload the recorded presentation video and the associated documents. Two input fields are provided to upload the required data. Once the video is loaded, it is displayed. Figure 4.5b shows the UI after the uploaded data were processed and the results were displayed based on the data stored in the *JSON* format. It displays the corresponding slide image, along the extracted information such as the *WER* and *CER* scores, the text extracted via *OCR*, and the title text extracted using the python-pptx library. Additionally, the transcriptions, summaries, and keywords generated from the transcription are displayed.



(a) Input field for recorded video and presentation document.



(b) Extracted results for each slide.

Figure 4.5: System User Interface showing the input and output stages.

4.3 Selection of Open-Source Models

This section outlines the process for selecting appropriate open-source models for the system.

4.3.1 Selection of Scene-Change Detection Algorithms

Since the algorithm needs to determine the transition points in video recordings based on scene changes, an accurate and fast technique is essential. However, while speed is important, accuracy must not be compromised, as it would impact the quality and disrupt subsequent processes. To meet these requirements, several algorithms from the *OpenCV* and *Scikit-Image* libraries were compared. The results are presented in Table 4.1.

Table 4.1: Comparison of image comparison algorithms.

Metric	***SSIM*** [96]	**Histogram Comparison** [97]			***MSE*** [98]
		Bhattacharyya	**Chi-Square**	**Correlation**	
Precision	100%	35.29%	26.32%	38.46%	11.36%
Recall	100%	60.00%	100.00%	50.00%	100.00%
F1 Score	100%	44.44%	41.67%	43.48%	20.41%
Duration (s)	180.85	296.91	309.79	297.13	173.16

Table 4.1 demonstrates the superiority of *SSIM* over other image comparison algorithms. *SSIM* achieves the perfect precision, recall, and *F1* scores. In contrast, *Histogram Comparison* and *MSE* struggle, especially with brightness variations and compression artifacts, leading to misclassified frames. Though *SSIM* requires longer processing time than *MSE*, its structural approach results in the higher accuracy, since it considers luminance, contrast, and image structure. This precision justifies the extra computation time, and makes *SSIM* the preferred choice for the scene change detection.

4.3.2 Selection of Audio-to-Text Models

The audio from each segmented video is extracted and converted into a text. Selecting an accurate model is essential to ensure for subsequent processes to accurately reflect the information conveyed in the audio. Additionally, the model efficiency is essential to reduce computational time. To identify the most suitable model, experiments were conducted to measure the *WER* and the computational duration (in seconds) of models from *HuggingFaceHub*. Table 4.2 shows the results.

Table 4.2: Comparison of *WER* and duration among audio-to-text models.

Component	***Wav2Vec*** [99]	***HuBERT*** [100]	***MMS*** [101]	***Whisper*** [102]
WER	51.23%	26.92%	19.38%	4.41%
Duration (s)	6.484	16.044	104.832	4.912

Table 4.2 compares the accuracy of *WER* and the computational duration. The *Whisper* model achieved the lowest *WER* and the shortest duration. While *Wav2Vec* had a relatively short duration,

its *WER* was the highest. Therefore, the *Whisper* model is the most suitable and thus, is selected for the system's audio-to-text conversion.

4.3.3 Selection of Summarization Models

Each transcribed text from the segmented video is summarized to help readers quickly grasp the main points of the presentation. The summary is generated using an abstractive approach, leveraging *Language Models (LMs)* to produce the coherent and concise output. Experiments were conducted to select the most suitable model from *HuggingFaceHub*. Table 4.3 shows the results.

Table 4.3: Comparison of *ROUGE-N* and duration among summarization models.

Metric	***T5*** [103]	***PEGASUS*** [104]	***BART*** [105]
ROUGE-1	30.66%	30.95%	39.34%
ROUGE-2	15.05%	16.30%	22.43%
ROUGE-L	25.56%	26.16%	35.35%
Duration (s)	4.448	1.374	0.962

Table 4.3 suggests that the *BART* model outperforms *PEGASUS* and *T5*, achieving the highest *ROUGE-1*, *ROUGE-2*, and *ROUGE-L* scores, as well as the fastest computational time. *BART*'s bidirectional encoder and auto-regressive decoder enables it to effectively process information from both the beginning and the end of sentences, allowing it to generate more coherent summaries. Despite its relatively complex architecture, *BART*'s high ability of summarization contributes to its computational efficiency. From these results, *BART* is the most suitable model for generating summaries in the system.

4.3.4 Selection of Keyword Extraction Models

For each transcribed segment, keywords are extracted to emphasize the key topics discussed. Various models were tested for keyword extractions, and a detailed comparison was made to identify the best-performing model. Table 4.4 shows the results.

Table 4.4: Comparison of Cosine similarity among keyword extraction models.

Metric	**KeyBERT** [106]	**RAKE** [107]	**YAKE** [108]	**TextRank** [109]	**RaKUn2** [110]
Cosine Similarity	28.46%	27.83%	34.01%	33.97%	47.70%

Table 4.4 highlights the performance differences across various keyword extraction models. *RaKUn2* achieved the highest *Cosine* similarity score at 47.70%, suggesting it is relatively effective at identifying relevant keywords within transcribed segments. Based on the results, *RaKUn2* emerged as the most suitable model for the keyword extraction and is employed in the system.

However, this measurement only considers the lexical similarity. Other factors such as the keyword diversity, duplications, and representativeness, are not captured by this metric alone. Therefore, these parameters should be explored further to gain insights into a more accurate keyword extraction model.

4.3.5 Selection of OCR Programs

The extracted information and slides image are aligned based on the extracted *OCR* text. Therefore, an accurate *OCR* program must be selected. To justify the selection, multiple *OCR* programs are compared and choose the lowest *CER* as the *OCR* for the system.

Table 4.5: Comparison of OCR algorithms

Size	**PaddleOCR** [111]		**Tesseract** [112]		**EasyOCR** [58]	
	Duration (s)	***CER***	**Duration (s)**	***CER***	**Duration (s)**	***CER***
1280×720	0.3442	0.00%	1.2262	0.00%	4.1956	0.00%
960×540	0.1951	0.00%	0.9713	0.00%	2.9337	0.00%
854×480	0.2672	0.00%	0.8805	0.00%	2.7653	0.00%
640×360	0.1892	0.00%	0.6989	0.00%	2.9206	0.00%
426×240	**0.1278**	**0.00%**	**0.1531**	**0.00%**	**3.6544**	**0.00%**
320×180	0.0991	5.22%	0.1338	4.26%	0.9936	8.88%
256×144	0.0773	7.02%	0.1204	100.00%	0.2197	55.56%
160×90	0.0594	15.40%	0.1010	100.00%	0.3462	100.00%

Table 4.5 shows that *PaddleOCR* consistently delivers the fastest processing time and maintains the high accuracy at any image resolution, showing the best overall performance. *Tesseract* performs well at higher resolutions but suffers from a significant accuracy drop at lower ones, reaching 100% *CER* below 320×180. *EasyOCR* is the slowest and least accurate, especially at lower resolutions, where error rates rise sharply. Thus, *PaddleOCR* is the most reliable and is selected as the best fit for the system.

4.4 Evaluation

In this section, the implementation of the proposed system is evaluated.

4.4.1 Experiment Preparation

Ten recorded presentation videos on different topics and with varying slide lengths are prepared as experimental materials to test the proposed system's ability to handle diverse contents. The videos were recorded with speakers from five non-native English accents including Vietnam (V), Congo (C), Singapore (S), Malaysia (M), and Indonesia (I). The recordings were created using the *zoom* screen share function and recorded through its meeting recording function. The output was a video file in *.mp4* format, with a resolution of 1280×720. Table 4.6 provides the details of each recorded presentation video.

Table 4.6: Recorded presentation videos for evaluations.

Topic	**Artificial Intelligence**	**Biotechnology**	**Blockchain**	**Cybersecurity**	**Renewable Energy**	**Environmental**	**Healthcare**	**Quantum Computing**	**Data Science**	**Space Exploration**
Duration (m:s)	5:50	5:29	6:44	6:37	6:53	6:05	5:28	4:32	6:13	8:10
Bitrate (kbps)	577	632	557	574	497	664	562	846	517	691

The video recordings have varied duration ranging from 4 to 8 minutes and average bitrate of $611 kpbs$ as shown in the Table 4.6. The proposed system is evaluated through statistical measurements and a user questionnaire. Each model's and algorithm's output are compared to reference data. This includes evaluating the *F1*-score for *scene change detection*, *WER* for *audio-to-text*, *ROUGE-N* and *ROUGE-L* for *abstractive summarization*, *cosine similarity* for *keyword extraction*, and *CER* for *OCR*. Then, the user questionnaire assessed the system's practical performance and the user experience. Each participant's session lasted about $45 min$.. Participants watched ten presentation videos covering varied topics and accents to introduce the speech pattern diversity. After viewing, users reviewed the system-generated meeting minutes and then, completed the questionnaire evaluating the system's usability, interface intuitiveness, meeting minute quality, audio transcription accuracy, and scene segmentation effectiveness.

4.4.2 Limitations and Biases of Recorded Videos

The recorded videos used in this study were designed to test the system across varied topics, presentation styles, and speaker accents, though some limitations should be noted. Speakers from Vietnam, Congo, Singapore, Malaysia, and Indonesia introduce accent variations. Yet, this sample set does not fully capture global English diversity. The topics, focusing on technical fields like AI and biotechnology, present structured, content-rich scenarios that may not depict more casual presentations. Audio quality was managed by recording in quiet rooms using standard laptop microphones, minimizing noise but possibly influencing transcription accuracy. Speech patterns were kept natural, though accents may still introduce phonetic challenges for the system. Presentation slides were standardized in the *.pptx* format without images or complex formatting, ensuring consistency but limiting the representation of real-world design diversity. These constraints provide a controlled testing environment but may not fully reflect broader real-world variability.

4.4.3 Performance of Split Video Function

This section evaluates the practical impact of *SSIM*'s algorithm. It categorizes video frames based on slide transitions as either "change" or "same" to accurately segment the video. To optimize processing time, video resolution was reduced to 426×240, and the algorithm was applied to one frame per second instead of every frame. This approach significantly lowered the execution time, enabling *SSIM* to segment videos within one minute while maintaining a perfect *F1* score as shown in Table 4.7. These results indicate that each detected frame change accurately corresponds the actual scene transition in the video.

Table 4.7: *SSIM* algorithm results for per-second approach.

Topic	Seconds	F1 Score	Execution Time (s)
Artificial Intelligence	350	100%	38.57
Biotechnology	329	100%	50.47
Blockchain	404	100%	47.17
Cybersecurity	397	100%	42.16
Renewable Energy	413	100%	42.37
Environmental	365	100%	39.62
Healthcare	328	100%	35.01
Quantum Computing	272	100%	27.56
Data Science	373	100%	39.86
Space Exploration	490	100%	46.96

4.4.4 Performance of Audio-to-Text Function

The *audio-to-text function* utilizes the *Whisper* model to convert *audio-to-text* and evaluates performance using *WER*. Audio is normalized to a sampling rate of $16kHz$. Despite the various speeches from non-English-speaking countries, including Vietnam, Congo, Singapore, Malaysia, and Indonesia, with varied duration, the *Whisper* model demonstrates averaged low *WER*, as shown in Table 4.8.

Table 4.8: Averaged *WER* on each slide from different English accents.

Accents	Slide 1	Slide 2	Slide 3	Slide 4	Slide 5
Vietnam	1.30%	1.58%	2.17%	0.82%	1.31%
Congo	1.32%	1.75%	2.45%	1.04%	1.60%
Singapore	0.89%	1.74%	2.37%	0.98%	1.43%
Malaysia	1.35%	1.57%	2.30%	0.98%	1.64%
Indonesia	1.36%	1.76%	2.41%	1.79%	1.40%

Table 4.8 shows the results. It demonstrates that the *Whisper* model consistently achieved low *WER* values across different accents from all the slides. It emphasizes *Whisper*'s robustness in handling varied accented speakers. Although some slides that contain the main information (i.e., slide2 and slide3) show slightly higher *WER* values. This behaviour may be attributed to the technical terms used in the presentation. However, the model continues to deliver accurate transcriptions on the other slides that are comparable to professional human transcribers [113].

4.4.5 Performance of Information Extraction Function

The *information extraction function* uses *BART LM* for *abstractive summarization* and *RaKUn2* for *keyword extraction*. *BART LM* is integrated into the proposed system due to its strong text generation capabilities, as evidenced by prior selections and reported in several studies [114, 115]. Similarly, *RaKUn2* was chosen based on its representative keywords extraction, supported by previous selection criteria and confirmed by [116].

Table 4.9: *ROUGE-1*, *ROUGE-2*, and *ROUGE-L* for each slide and topic.

Metric	Topic	Artificial Intelligence	Biotechnology	Blockchain	Cybersecurity	Renewable Energy	Environmental	Healthcare	Quantum Computing	Data Science	Space Exploration
ROUGE-1	Slide 1	27.4%	49.9%	33.9%	44.4%	49.9%	49.9%	59.5%	46.1%	0%	35.2%
	Slide 2	36.7%	45.7%	4.1%	60.8%	42.4%	69.1%	60.2%	46.8%	47.4%	33.5%
	Slide 3	68.4%	30.6%	45.3%	43.7%	52.9%	29.8%	37.9%	38%	50.6%	31.9%
	Slide 4	43.2%	48.1%	51.7%	55.3%	32.9%	41%	72.7%	46.8%	43.9%	55.7%
	Slide 5	41.5%	-	46.3%	38.2%	19%	39.6%	62.7%	45.3%	43.9%	49%
ROUGE-2	Slide 1	14%	19.6%	6.3%	9.8%	21.8%	27.1%	40.7%	20.3%	0%	22.9%
	Slide 2	5.9%	9.5%	21.7%	33.3%	16.2%	44.7%	33.6%	22.2%	19.1%	12.8%
	Slide 3	37.5%	8.6%	6.8%	18.8%	23.2%	9.3%	14.7%	17.1%	28.8%	15.5%
	Slide 4	13.1%	20.8%	17.9%	20.2%	11.7%	8.1%	44.4%	15.9%	20.6%	37.8%
	Slide 5	18.9%	-	16.2%	8.8%	5.8%	10.7%	20.3%	21.9%	10.2%	29.8%
ROUGE-L	Slide 1	23.5%	38.4%	22.6%	37%	45.8%	49.9%	59.5%	46.2%	0%	35.2%
	Slide 2	22.9%	25.7%	39.5%	55.7%	34.3%	64.2%	55.9%	36.3%	27.4%	19.1%
	Slide 3	50.6%	27%	34.6%	37.4%	44.1%	20.8%	31%	30.9%	45.8%	23.9%
	Slide 4	23.4%	40.5%	34.4%	33.8%	21.1%	24.6%	62.3%	36.3%	26.3%	39.3%
	Slide 5	23.7%	-	37.6%	27.6%	9.5%	31.6%	46.5%	34.6%	31.7%	43.1%

Table 4.9 shows that *BART LM* models perform variably across slides and topics, with *ROUGE-1*, *ROUGE-2*, and *ROUGE-L* scores peaking on slides 2, 3, and 4 across most topics, reflecting the typical presentation structure. *ROUGE* scores assess summary quality by comparing generated summaries to reference ones. Specifically, *ROUGE-1* measures uni-gram (single-word) coverage, *ROUGE-2* evaluates fluency through bi-gram (two-word) overlap, and *ROUGE-L* checks the longest common sub-sequence (continuous adjacent words) to assess structural coherence. High scores across technical and general topics indicate that the system produces coherent, reliable summaries across varied domains. This consistency suggests that *BART LM* is suitable for various summarization tasks.

The keyword extraction measures the similarity between the extracted and reference keywords using *cosine similarity* [117]. Table 4.10 illustrates the *cosine similarity* of the extracted keywords for each slide from every topic.

Table 4.10: Keywords *cosine similarity* from each slide and each topic.

Topic	**Artificial Intelligence**	**Biotechnology**	**Blockchain**	**Cybersecurity**	**Renewable Energy**	**Environmental**	**Healthcare**	**Quantum Computing**	**Data Science**	**Space Exploration**	**Average (per slide)**
Slide 1	0.714	0.503	0.669	0.462	0.487	0.721	0.874	0.801	0.394	0.566	**0.619**
Slide 2	0.504	0.597	0.453	0.423	0.481	0.356	0.447	0.496	0.429	0.499	**0.469**
Slide 3	0.478	0.577	0.516	0.484	0.549	0.499	0.522	0.592	0.509	0.462	**0.519**
Slide 4	0.571	0.559	0.526	0.577	0.649	0.539	0.547	0.416	0.725	0.362	**0.547**
Slide 5	0.306	-	0.316	0.340	0.360	0.399	0.416	0.547	0.269	0.441	**0.377**
Average (per topic)	**0.515**	**0.559**	**0.496**	**0.457**	**0.505**	**0.503**	**0.561**	**0.570**	**0.465**	**0.466**	**0.509**

Table 4.10 shows that the *RaKUn2* algorithm achieved the average *cosine similarity* score of 0.509 from all the slides and topics, capturing over half of the reference keywords accurately. While effective for general keyword identification, the cosine similarity relies mainly on lexical overlap, potentially missing deeper semantic nuances, especially in topics requiring complex contextual understanding [118].

4.4.6 Performance of Information Correlation Function

The *information correlation function* aligns the extracted text, summaries, and keywords with their corresponding presentation slides. Supported by the selection in Table 4.5, *PaddleOCR* is used to detect the title text from segmented video frames. It is then matched to the presentation document through *regex* matching. Since this matching mechanism relies on the string-level similarity, the perfect *OCR* text extraction as shown in Table 4.11 ensures error-free matching. However, as the presentations are controlled in this study, these results may not directly generalize to real-world scenarios where design variations are common.

Table 4.11: *CER* results from each slide and topic from resized video using *PaddleOCR*.

Topic	**Artificial Intelligence**	**Biotechnology**	**Blockchain**	**Cybersecurity**	**Renewable Energy**	**Environmental**	**Healthcare**	**Quantum Computing**	**Data Science**	**Space Exploration**
Slide 1	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Slide 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Slide 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Slide 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Slide 5	0%	-	0%	0%	0%	0%	0%	0%	0%	0%

4.4.7 Performance Comparison with Other Meeting Minutes Systems

This experiment evaluates the performance of our proposed system by comparing it with other existing systems. Due to the proprietary nature of these systems, data were manually gathered from free or trial versions. Thus, the execution time, meeting transcription, summary generation, keyword extraction, interface, and customization capabilities were compared, to demonstrate the effectiveness of integrating open-source models relative to professionally developed systems. Key metrics in this comparison included *WER* for transcription accuracy, *ROUGE-N* scores on the summary quality, and the cosine similarity for keyword extraction relevance. Table 4.12 shows comparison results.

Table 4.12: Performance comparison with other existing systems.

Application	**Time**	**Transcription (*WER*)**	**Summary**			**Keywords (Cosine-Similarity)**	**Runs on**	**Custom**
			ROUGE-1	**ROUGE-2**	**ROUGE-L**			
Meeting Booster [119]	1:18	2.24%	-	-	-	-	Browser	No
Fellow [120]	2:18	2.03%	40.47%	23.11%	31.53%	<u>0.511</u>	Browser, Desktop	No
Beenotes [121]	5:48	2.91%	38.09%	21.07%	29.47%	0.496	Desktop	No
Piglyph [122]	3:02	2.72%	**42.03%**	**25.08%**	**33.12%**	**0.521**	Desktop	No
Tactiq [123]	3:54	2.81%	39.14%	22.17%	30.61%	0.427	Browser	No
Our proposal	**1:34**	**1.88%**	<u>41.06%</u>	<u>23.76%</u>	<u>32.12%</u>	0.509	Browser	Yes

In Table 4.12, the bolded values represent the highest scores, while the underlined values denote the second-highest scores. These results indicate that the proposed system achieved the lowest *WER* of 1.88%, outperforming other systems in the transcription accuracy. While *MeetingBooster* had the fastest execution time, it only provided transcription. In contrast, our system handles transcriptions, summaries, and keywords, achieving the second-best time (1:34). For summarization, *Piglyph* achieved the highest *ROUGE-N* metrics, with our system closely following with second-highest scores, 41.06% for *ROUGE-1*, 23.76% for *ROUGE-2*, and 32.12% for *ROUGE-L*. *Piglyph* led with a *cosine similarity* of 0.521 in keyword extraction, while our system achieved 0.509 keyword relevance. The browser-based design offers easy access, and the system's use of customizable open-source models enables model updates, ensuring adaptability to the latest advancements.

Although our system does not consistently achieve the highest scores, its performance closely aligns with professionally developed systems. The system currently performs best with simple presentation formats and further testing is needed to confirm its effectiveness with more complex layouts, diverse speaker styles, and varied slide designs.

4.4.8 Questionnaire Results on Usability and Effectiveness

The usability and effectiveness of the proposed system were evaluated using a questionnaire based on the *Performance, Information, Economics, Control and Security, Efficiency, and Service*

(PIECES) framework [124]. The questionnaire was administered to 31 respondents. Each respondent first reviewed a provided presentation document, listened to the presentation recording, and executed the proposed system. Feedback was subsequently collected through the questionnaire, with the specific questions listed in Table 4.13.

Table 4.13: Questions in questionnaire on usability and effectiveness.

Question ID	Questionnaire Questions
Q1	The system was easy to use in extracting textual information from presentation videos.
Q2	The system's interface was intuitive and easy to navigate while correlating information between presentation videos and PowerPoint files.
Q3	I am satisfied with the accuracy of the system in converting audio to text and correlating textual information from both presentation videos and PowerPoint files.
Q4	I am satisfied with the quality of the generated summary and the extracted keywords.
Q5	I would recommend this system to others for similar tasks requiring information correlation between presentation videos and PowerPoint files.

Each question represents a key aspect of the framework. Questions Q1, Q2, and Q5 focus on usability aspects, such as ease of use, intuitive interface, and likelihood of recommending the system. Specifically, Q1 evaluates *Service* by measuring how quickly users can extract information. Q2 examines *Control and Security*, assessing how intuitively users can manage the system. Q5 checks *Economics* and *Efficiency*, gauging if users would recommend the system for similar tasks. Q3 and Q4 focus on effectiveness of the system. Q3 assesses *Information* by evaluating the accuracy of the audio-to-text conversion. Q4 addresses *Performance*, reflecting the quality of summaries and keyword extraction. Table 4.14 shows their responds to the questionnaire.

Table 4.14: Answers to questions in questionnaire.

Question ID	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Q1	3	0	6	14	8
Q2	0	1	9	19	2
Q3	0	0	11	10	10
Q4	0	3	16	4	8
Q5	2	4	3	14	8

As shown in Table 4.14, the system received positive feedback on ease of use, interface intuitiveness, and recommendation likelihood. Most users found the system is easy to navigate. 22 respondents agreed that it was simple to extract information from videos, and 21 appreciated its intuitive design. These findings align with the performance of the *Whisper* model, which achieved a low WER. However, mixed responses on the summary and keyword quality, as indicated by the responses to Q4, suggest that while the *BART* model and *RaKUn2* performed well, there is a room for improvements. The system accurately aligned extracted information with presentation

slides, as indicated by the responses to Q2, utilizing *PaddleOCR* and *regex*. Overall, the system is user-friendly, accurate, and effective, with potential areas for enhancements.

4.5 Discussion

Online meetings and online presentations have become essential with the rise of remote works. They often lead participants to over-fatigue due to information overloads and cramped meetings. To address this, the proposed system extracts, summarizes, and correlates the key information from recorded meetings using fully open-source AI models. The system enables users to review essential contents without replaying entire sessions and to update the system whenever a new advancement of an AI model occurs. Following a modular and task-specific approach, each audio, text, and visual data is processed individually to maximize computational efficiency and accuracy.

The *WER* by *Whisper* is closely similar to the human transcription accuracy [30]. Although the *ROUGE-N* scores, *cosine similarity*, and user feedback obtained in this study indicate slight limitations in generating cohesive summaries and representative keywords, in practical settings, their usefulness often depend on how effectively they capture the critical information discussed in the meeting.

Another limitation of the proposed system is that it was tested with controlled recorded online meeting presentations. These presentations excluded images, figures, and animations, and used a single background color with the standard font size and style. The speech pattern was kept uniform, with no variation in speed or intonations.

Considering these limitations, future investigations will focus on varied presentation formats and speech patterns. More complex scenarios will represent real-world challenges. Additional techniques or model fine-tuning can be necessary to ensure the system can generate usable information from more complex inputs.

4.6 Summary

With the rise of remote works, online meetings and presentations have become essential, often leading to participant fatigues in information overload and back-to-back sessions. The proposed system addresses this issue by extracting, summarizing, and correlating key information from recorded meetings. As fully open-source AI models, the system employs *Whisper* for *audio-to-text* transcription, *BART* for *summarization*, *RaKUn2* for *keyword extraction*, *SSIM* for scene detection, and *PaddleOCR* for extracting and *regex* for correlating visual-textual data.

Evaluation on ten recorded presentations demonstrates the system's accuracy, which emphasizes the potential to streamline note-taking for meetings, conferences, and seminars. The practical impact could be beneficial for organizations and educational settings that rely on efficient meeting reviews. However, accommodating diverse and dynamic presentation designs and varied speech patterns remains an area for future investigation.

Chapter 5

Development of an LLM-Based Slide Annotation System for Online Presentation Review

5.1 System Overview

This study applies multimodal analysis to support the automatic generation of slide annotations. The proposed system is implemented as a web-based application consisting of a client side for user interaction and a server side for the core processing. An overview of the proposed *Slide Annotation System* is shown in Figure 5.1.

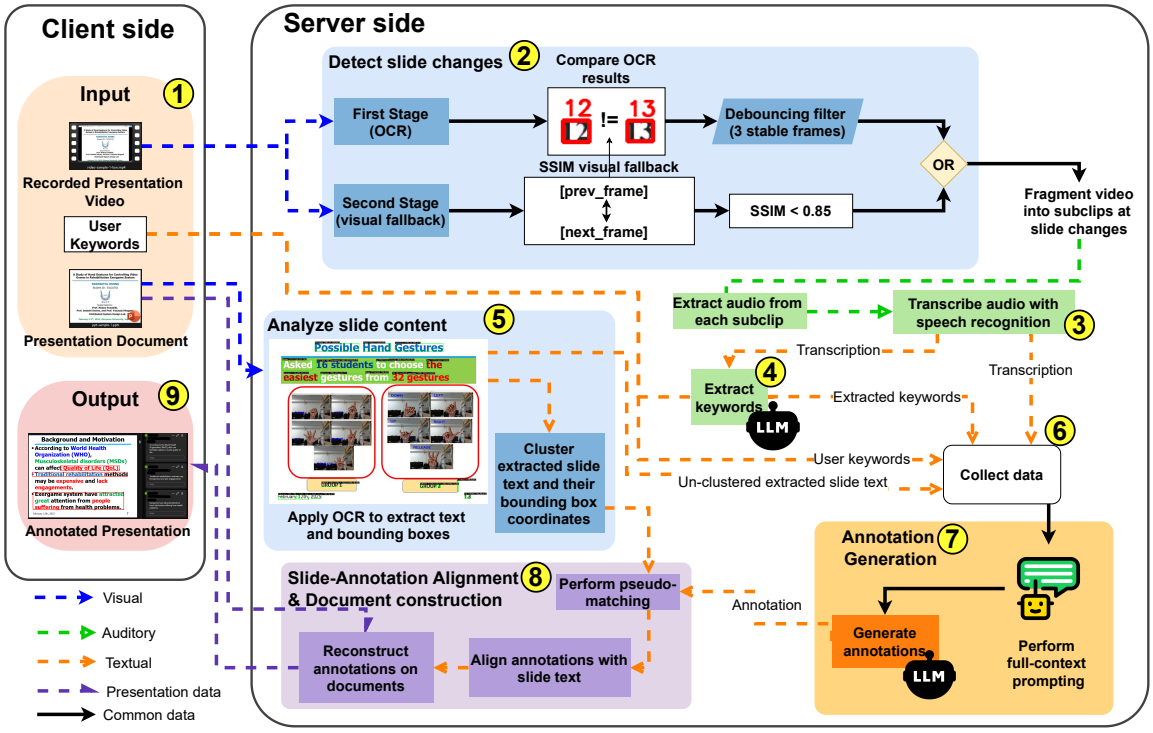


Figure 5.1: System overview of the Slide Annotation System with a numbered workflow. Best viewed in color and at full size for clarity.

On the client side, the system provides a user interface that allows users to upload presentation video and documents, specify keywords of interest, and retrieve the final annotated output. On the server side, the system first segments the video by detecting slide transitions using the Two-Stage detection, where if the *OCR* fails to detect slide change the *visual fallback* mechanism using *SSIM* operates. For each segmented clip, the corresponding audio is transcribed using speech recognition. Keywords are extracted from the transcription according to the user-defined keywords of interest. Meanwhile, the presentation slides are processed using *OCR* to obtain slide text and bounding box information. These multimodal data are then fused and used as input for an *LLM*-based annotation generation process. The generated annotations are aligned with the extracted slide text through an approximate text-matching step. Finally, the aligned annotations are reconstructed within the presentation document and provided to the user.

5.2 Input

The proposed system relies on three primary inputs, the recorded presentation video, the presentation document, and user-defined keywords. The *User Interface (UI)* is implemented as a web-based application using Streamlit.

The uploaded presentation video serves as the primary source of visual and auditory information, with a resolution of 1920×1080 at 30 frames per second and an audio sampling rate of 48 kHz. The system supports common video formats such as *.AVI* and *.MP4*. After the video is uploaded, the user is prompted to define a *Region of Interest (ROI)* by drawing a rectangle around a consistent visual element, typically the slide number or a fixed footer, within a sampled reference frame. This *ROI* is displayed in the interface, as shown in Figure 5.2, to enable digit recognition by *OCR* and visual stability checks.

The uploaded presentation document serves as a basis for reconstructing the final annotated presentation and contributes additional textual information extracted through *Optical Character Recognition (OCR)*. *OCR* processing extracts both the text content and the corresponding bounding box coordinates from each slide. Additionally, user-defined keywords guide the annotation generation process by emphasizing terms of interest, helping to generate more focused and contextually relevant annotations.

5.3 Processing

This subsection describes the processing pipeline, outlining the main automated operations running in the background to minimize user involvement.

5.3.1 Slide Change Detection

In this paper, segmentation refers to the temporal partitioning of presentation videos into slide-level clips. While prior works often rely solely on detecting slide numbers via *OCR* [70, 125], such methods can be prone to failure when numbers are obscured, absent, or decorative. The complete proposed hybrid workflow is summarized in Algorithm 1.

First, *Tesseract OCR* extracts numeric indicators from the *ROI*. A temporal debouncing filter is applied to mitigate flickering noise; a transition is recorded only if the digit sequence changes and remains stable for a 3-frame window. This duration was optimized to eliminate noise without latency, as seen in Section 5.5.4.

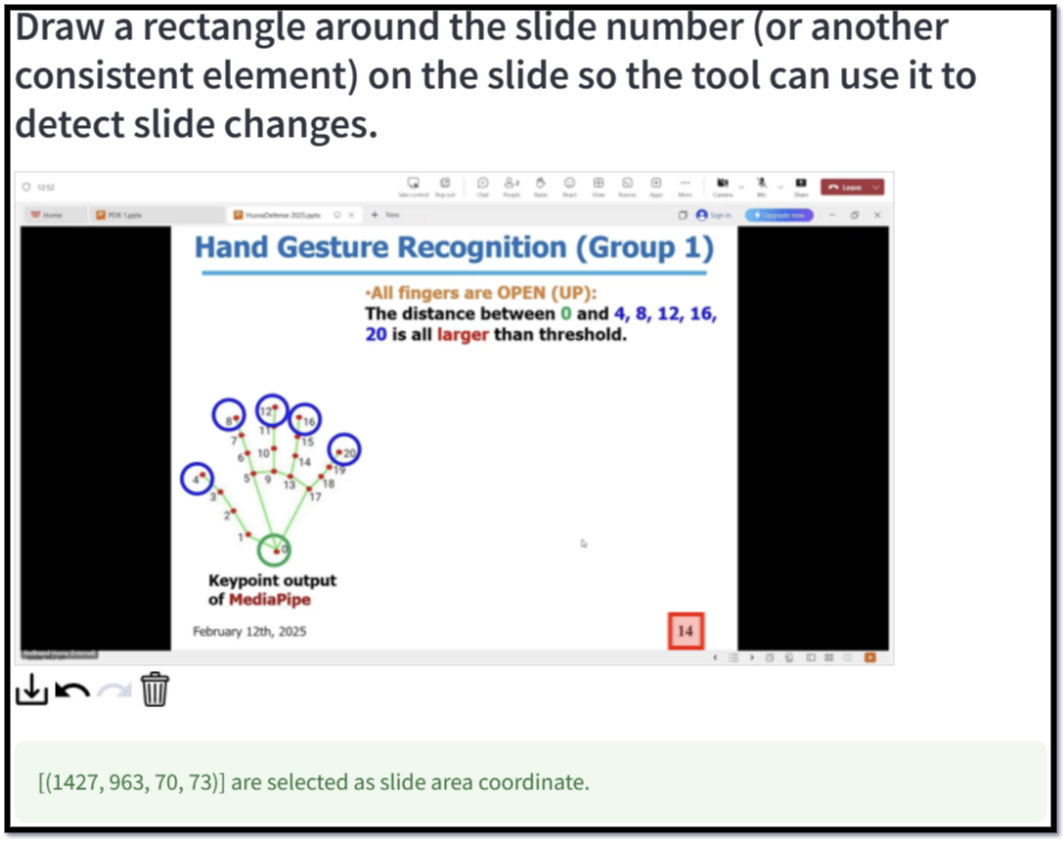


Figure 5.2: User-defined *ROI* drawn around the slide number (bottom-right), used as the reference area for hybrid slide-change detection.

Second, as a fallback for low-confidence *OCR*, the system computes the *Structural Similarity Index (SSIM)* between frames. A boundary is triggered if the score drops below $\tau = 0.85$, indicating significant visual change. This threshold was empirically tuned to maximize recall while minimizing false positives, as detailed in Section 5.5.4.

5.3.2 Speech Recognition

As most semantic content in presentations is conveyed through speech, this component transcribed each segmented clip's audio into text. Whisper, an open-source *Automatic Speech Recognition (ASR)* model from OpenAI known for its robustness and high accuracy under diverse conditions, is employed [126, 127]. The *small.en* variant is chosen for its balance of accuracy and computational efficiency for English speech [84]. The raw audio stream was processed directly without external *Voice Activity Detection (VAD)* or additional *DSP* noise suppression, relying on the model's architecture to robustly handle the ambient acoustic conditions typical of lecture recordings.

To reduce noise, raw transcripts are cleaned by removing filler words such as "um", "yeah", and "uh" without altering the meaning. Cleaned transcripts are paired with file names and stored in structured JSON format for data exchange. The workflow is summarized in Algorithm 2.

Algorithm 1 Hybrid slide segmentation (*OCR* with *visual fallback*).

Require: V (VideoClip), *ROI* (Coordinates), τ (SSIM Threshold: 0.85)
Ensure: Clips: list of records $\{file,\ segment,\ start,\ end\}$

$B \leftarrow [0]$; $prev\_frame \leftarrow GetFrame(V, 0)$
$prev\_ocr \leftarrow OCR(prev\_frame, ROI)$
for $t \leftarrow \Delta t$ **to** $V.duration$ **step** Δt **do** ▷ Sampling interval (3s)
  $curr\_frame \leftarrow GetFrame(V, t)$
  $curr\_ocr \leftarrow OCR(curr\_frame, ROI)$
  $is\_boundary \leftarrow$ **false**
  if $curr\_ocr$ is valid number **then** ▷ **Stage 1: *OCR* Check**
    if $curr\_ocr \neq prev\_ocr$ **and** $IsStable(curr\_ocr)$ **then** ▷ Debouncing
      $is\_boundary \leftarrow$ **true**
      $prev\_ocr \leftarrow curr\_ocr$ ▷ Update reference number
    end if
  else ▷ **Stage 2: *visual fallback***
    $sim\_score \leftarrow SSIM(curr\_frame, prev\_frame)$
    if $sim\_score < \tau$ **then**
      $is\_boundary \leftarrow$ **true**
    end if
  end if
  if $is\_boundary$ **then**
    $B.append(t)$ ▷ Record timestamp
    $prev\_frame \leftarrow curr\_frame$ ▷ Update reference frame
  end if
end for
$B.append(V.duration)$; WriteToFile(B, "`boundaries.txt`")
Clips $\leftarrow$ ExtractAudioSegments(V, B)
↵ Clips

Algorithm 2 Whisper-based transcription pipeline with filler removal and JSON storage.

Require: W: Whisper *ASR* model (preloaded); Clips: list of records $\{file\_path,\ segment,\ start\_time,\ end\_time\}$; F: predefined filler-word set
Ensure: `segments.json`: JSON file with cleaned transcripts and filenames

Segments $\leftarrow$ [ ]
for each $c \in$ Clips **do**
  $(file\_path, i, s, e) \leftarrow (c.file\_path,\ c.segment,\ c.start\_time,\ c.end\_time)$
  $raw \leftarrow WhisperTranscribe(W,\ file\_path)$
  $tokens \leftarrow Tokenize(raw)$
  $filtered \leftarrow [\, w \mid w \in tokens \wedge w \notin F \,]$ ▷ remove filler words
  $transcript \leftarrow Join(filtered)$
  Record $\leftarrow \{file\_path : file\_path,\ transcription : transcript,\ segment : i,\ start\_time : s,\ end\_time : e\}$
  Segments.append(Record)
end for
WriteJSON(Segments, "`segments.json`") ▷ export structured transcripts
↵ Segments

5.3.3 Keyword Extraction

This component extracts key terms from transcripts using both speech content and user-defined keywords to emphasize topics of interest. Extraction is performed with an *LLM* using a few-shot prompting strategy, where prompts include task instructions, output constraints, and examples to guide the model [128].

The prompt design assigns the *LLM* the role of a keyword extractor and applies a weighted scoring scheme based on relevance, frequency, specificity, and contextual alignment, with additional weight for user-defined keywords [129]. This approach improves semantic quality compared to traditional frequency-based methods [130].

Implementation uses gemma3:4b with Q4_K_M quantization via Ollama and LangChain, leveraging its 128k-token context window to process long transcripts without *Retrieval-Augmented Generation (RAG)*. Extracted keywords and relevance scores are stored in JSON for downstream annotation. The prompt structure is shown in Figure 5.3. The full prompt template and parameters used for keyword generation are provided in Appendix 8.1.

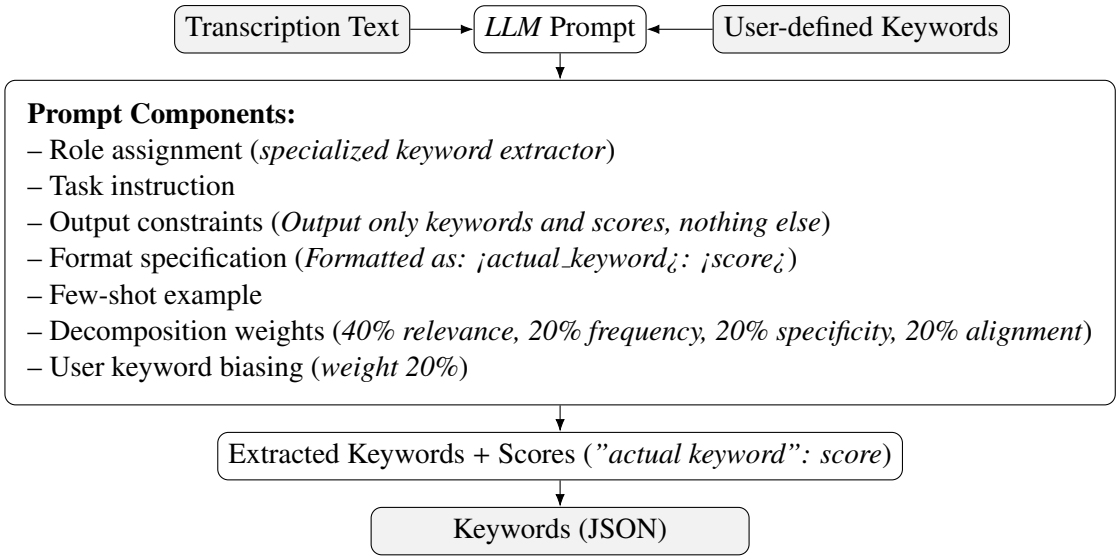


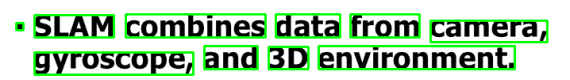
Figure 5.3: Few-shot prompt structure illustrating the inputs, role assignment, and weighted scoring mechanism for keyword extraction.

5.3.4 Slide Content Analysis

This component extracts text and bounding boxes from slides using *OCR*, which can detect text embedded in visual elements such as figures and charts [131]. For Latin-script documents, Tesseract processes text in a top-to-bottom, left-to-right order, often producing word-level bounding boxes as illustrated in Figure 5.4.

To reconstruct meaningful text regions, bounding boxes are grouped by spatial proximity using DBSCAN along the vertical axis. We configured the clustering algorithm with a vertical epsilon (ϵ_y) of 10 pixels, which is based on a standard 1080p frame height, to effectively bridge the gap between adjacent text lines, and a minimum points (min_pts) threshold of 1, ensuring that isolated text elements such as labels or page numbers are preserved. This simplified approach clusters words with similar y coordinates, merging fragmented tokens into coherent text blocks as shown in Figure 5.4c. Each block, along with its bounding box, is stored in JSON for annotation alignment.

(a) Original image.



(b) Raw word-level bounding boxes detected by the *OCR* engine.

(c) Coherent text blocks formed by grouping the raw bounding boxes using DBSCAN based on vertical proximity.

Figure 5.4: Illustration of *OCR* and clustering results.

Raw OCR tokens are also preserved for contextual completeness. The workflow is summarized in Algorithm 3.

Algorithm 3 Per-slide *OCR* token clustering with DBSCAN and JSON export.

Require: Slides: list of $\{slide\_id,\ tokens = \{text, x, y, w, h\}\}$; parameters eps_y, min_pts
Ensure: `raw_tokens.json`, `clustered_blocks.json`

Raw ← [ ]; Clusters ← [ ]
for each $(sid, Tokens)$ in Slides **do**
  Raw.append($\{slide\_id : sid,\ tokens : Tokens\}$)
  $Y \leftarrow [\, y + h/2 \mid (x, y, w, h) \in \text{Tokens}\,]$
  labels ← DBSCAN(Y, $eps\_y = 10$, $min\_pts = 1$)
  Blocks ← [ ]
  for each cluster c in labels **do**
    $S \leftarrow$ tokens with label c, sorted by x
    Blocks.append($\{id : c,\ bbox : BBox(S),\ y\_anchor : MeanY(S),\ text : Concat(S)\}$)
  end for
  Sort(Blocks, by y_anchor then $bbox.x$)
  Clusters.append($\{slide\_id : sid,\ blocks : Blocks\}$)
end for
WriteJSON(Raw, "`raw_tokens.json`"); WriteJSON(Clusters, "`clustered.json`")
↵ Clusters

5.3.5 Annotation Generation

This component generates slide-level annotations using contextual information from multiple sources, following the approach of List and Lin [57]. It integrates multimodal outputs including transcriptions, extracted keywords, and raw *OCR* text into unified data for each video sub-clip. Transcriptions provide narrative content, keywords guide the *LLM* toward key topics, and *OCR* text offers slide-specific cues such as section titles.

Leveraging Tesseract's reading order, the system identifies slide titles, typically from top-left, and uses them for prompt selection. If a title matches common academic sections such as *Introduction*, *Background*, or *Results*, a specialized prompt is applied. Otherwise, a default system prompt is used. This design follows the principles in [128], enabling the same *LLM* to act as multiple

task-specific extractors without fine-tuning.

Annotation generation is powered by the gemma3:4b Q4_K_M quantization model, accessed via Ollama and LangChain. To ensure reproducibility, this study adhered to strict inference settings: the *temperature* was set to *0.2* to minimize hallucination and maintain factual consistency, and the *random seed* was fixed to *42* for deterministic outputs. The full system instructions and specialized prompt templates used for annotation generation are provided in Appendix 8.2. The conceptual prompt structure is illustrated in Figure 5.5.

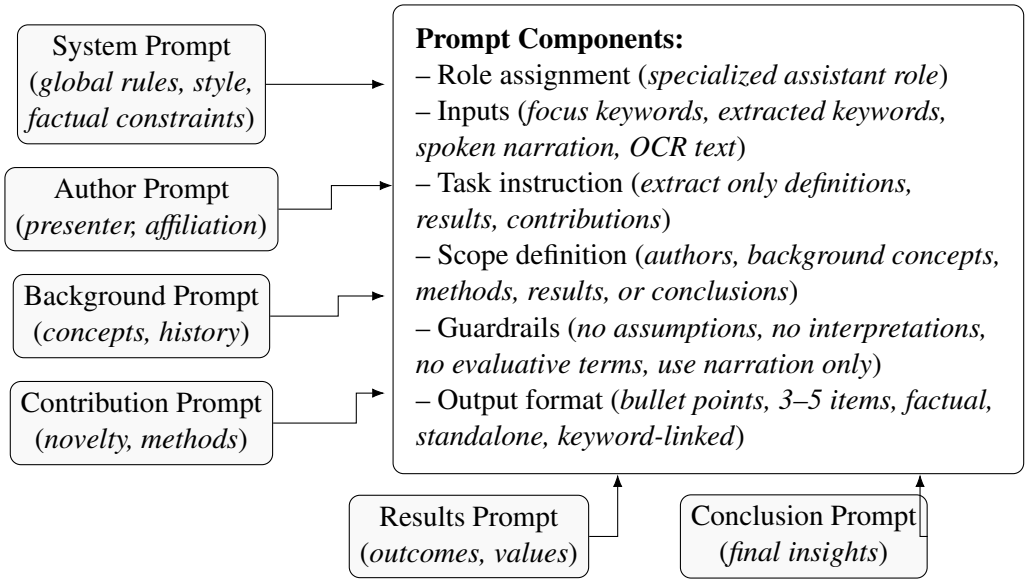


Figure 5.5: Hierarchical *LLM* Prompting Architecture for structured information extraction.

The prompt design includes a system prompt for global rules such as factual accuracy, conciseness, style and specialized prompts for specific content types including background, methods, results, conclusions. This enables context-aware annotation generation without fine-tuning. Each annotation is stored as a structured JSON record along with its context, such as transcription snippet, keywords, and *OCR* text. The end-to-end workflow, from context aggregation to prompt-based inference and JSON storage, is shown in Algorithm 4.

Algorithm 4 Annotation generation with context aggregation and prompt selection.

Require: D: JSON record with $\{file\_path, transcription, keywords, ocr\_tokens\}$
Ensure: `annotations.json`

function AGGREGATECONTEXT(F, T, K, O)
    $T' \leftarrow \text{Clean}(T)$ ▷ remove *ASR* glitches, filler remnants
    $K' \leftarrow \text{Deduplicate}(K)$
    $O' \leftarrow \text{Normalize}(O)$ ▷ retain early tokens for titles
    ↵ $\{file\_path : F,\ transcription : T',\ keywords : K',\ ocr : O'\}$
end function
procedure GENERATEANNOTATION(D)
    $(F, T, K, O) \leftarrow (D.file\_path, D.transcription, D.keywords, D.ocr\_tokens)$
    $C \leftarrow \text{AggregateContext}(F, T, K, O)$
    $title \leftarrow \text{FirstOr}(O, \varnothing)$
    if IsTitleMatch(*title*) **then**
        $p \leftarrow \text{SelectPrompt}(title)$ ▷ specialized prompt
    else
        $p \leftarrow \text{SystemPrompt}()$
    end if
    $y \leftarrow LLM\_\text{Inference}(p, C)$
    $\text{Record} \leftarrow \{file\_path : F,\ annotation : y,\ prompt : p,\ context : C\}$
    AppendJSON(Record, "`annotations.json`")
    ↵ Record
end procedure

5.3.6 Annotation Alignment and Slide Reconstruction

This component represents the final stage of the backend pipeline, aligning generated annotations with corresponding slide content and reconstructing them into an enriched presentation format. It integrates textual annotations produced by the *LLM* with spatial/visual information extracted from slides to ensure contextual accuracy and spatial consistency. First, annotations are mapped to relevant slide regions based on semantic similarity. Then, aligned annotations are embedded back into the presentation as comments anchored to corresponding visual elements. The resulting annotated presentation is provided as a downloadable file via the *UI*.

Annotation alignment. The process of aligning the generated annotation text to the precise region of the OCR slide text is a multi-stage retrieval pipeline. This alignment is crucial for visual grounding and is summarized conceptually in Figure 5.6. It operates by first selecting initial candidates using a composite scoring mechanism, followed by precise re-ranking via a deep cross-encoder model.

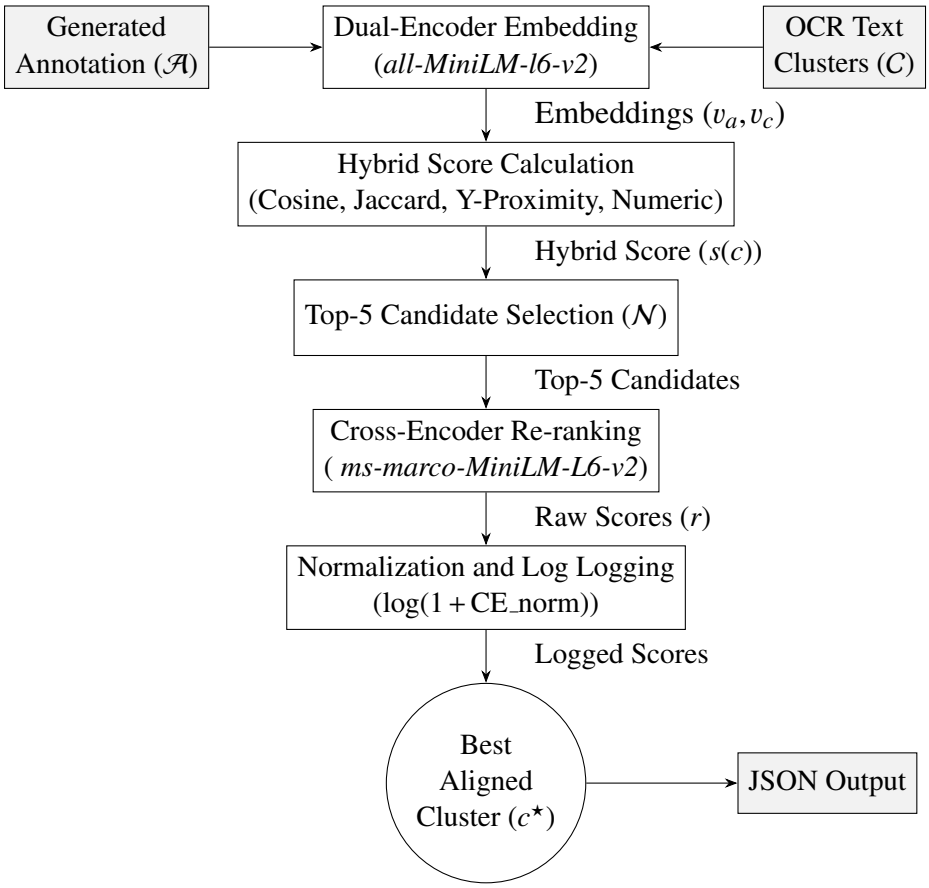


Figure 5.6: Flow Diagram of the Multi-Stage Annotation Alignment Process.

Hybrid alignment score is computed from four equally weighted metrics to map annotations to slide regions. The first component is cosine similarity between L2-normalized embeddings using *all-MiniLM-l6-v2*. The second is Jaccard token similarity, computed case-sensitively and restricted to alphanumeric tokens from the annotation and the *OCR* text. The third is Y-proximity, defined as the vertical center distance of the *OCR* text relative to the annotation. The fourth is numeric matching, such as overlap of numbers appearing in the annotation and the *OCR* text. The combined score is the mean of these four components. Then, the top-5 candidates are selected

and re-ranked using a cross-encoder *ms-marco-MiniLM-L6-v2* model by taking the argmax of the cross-encoder score. The cross-encoder raw scores are logged and normalized to the range from 0 to 1. The complete workflow, from text embedding to cross-encoder re-ranking and JSON storage, is summarized in Algorithm 5.

Algorithm 5 Hybrid Top-5 with cross-encoder re-ranking for annotation alignment.

Require: Annotations $\mathcal{A}$ with fields $\{file\_path, annotation, prompt, context\}$; *OCR* clusters C with $\{id, bbox, y\_anchor, text\}$; embedder E; cross-encoder CE
Ensure: JSON $\mathcal{J}$ with aligned pairs

for all $c \in C$ **do**
  $v_c \leftarrow \mathrm{norm}_2(\mathsf{E}(c.text))$
end for
for all $a \in \mathcal{A}$ **do**
  $v_a \leftarrow \mathrm{norm}_2(\mathsf{E}(a.annotation))$
  $C_{\text{slide}} \leftarrow \{c \in C : c.slide\_id = a.slide\_id\}$
  for all $c \in C_{\text{slide}}$ **do**
    $\cos \leftarrow v_a \cdot v_c$
    tok $\leftarrow$ JaccardTokens($a.annotation$, $c.text$)
    yprox $\leftarrow$ YProximity(a, c)
    num $\leftarrow$ NumMatch($a.annotation$, $c.text$)
    $s(c) \leftarrow (\cos + \text{tok} + \text{yprox} + \text{num})/4$
  end for
  $\mathcal{N} \leftarrow \mathrm{TopK}_c(s(c), 5)$
  CE_scores $\leftarrow$ []
  for all $c \in \mathcal{N}$ **do**
    $r \leftarrow \mathsf{CE}(a.annotation, c.text)$
    CE_scores.append(r)
  end for
  CE_norm $\leftarrow$ NormalizeTo01(CE_scores)
  CE_log $\leftarrow \log(1 + \text{CE\_norm})$
  $c^\star \leftarrow \arg\max_{c \in \mathcal{N}} \text{CE\_log}(c)$
  Record $\leftarrow$ {annotation:$a.annotation$, ocr_cluster:$c^\star.text$, bbox:$c^\star.bbox$, slide_id:$c^\star.slide\_id$}
  AppendJSON(Record, `"aligned.json"`)
end for

Slide reconstruction. Aligned JSON records are used to reintegrate annotations into the presentation, producing an enriched version of each slide. Bounding boxes from the *OCR* process are converted into PowerPoint coordinates using a calibration function that scales and translates positions to the target slide dimensions. For each matched annotation, a transparent rectangle shape with a visible border is drawn at the calibrated location, and the annotation text is attached as a comment anchored to that shape.

This design preserves the visibility of the original slide content while keeping each annotation contextually linked to its visual element. By anchoring comments to shapes rather than overlaying text directly on the slide, the system avoids occlusion and maintains clarity. To ensure spatial accuracy, bounding box coordinates are mapped from the video frame resolution to the presentation slide dimensions using a uniform scaling function ($\phi_x = \phi_y = 0.75$), as detailed in Algorithm 6. This factor represents the conversion from screen pixels (96 DPI) to PowerPoint points (72 DPI),

ensuring that the aspect ratio of the annotated regions is preserved without distortion during the import process.

Algorithm 6 Slide reconstruction with bounding-box anchored comments

Require: Aligned JSON $\mathcal{J}$; OCR canvas $(W_{\text{ocr}}, H_{\text{ocr}})$; PowerPoint path
Ensure: PowerPoint updated with shapes and anchored comments

OpenPresentation()
$cf_x \leftarrow 0.75$ ▷ 72 pt/in ÷ 96 px/in
$cf_y \leftarrow 0.75$
$\Phi(bbox = (x, y, w, h)) \leftarrow (x \cdot cf_x,\ y \cdot cf_y,\ w \cdot cf_x,\ h \cdot cf_y)$
for all $j \in \mathcal{J}$ **do**
    SelectSlide(j.`slide_id`)
    $(L, T, W, H) \leftarrow \Phi(j.\texttt{bbox})$
    $shape \leftarrow \textsc{DrawRectangle}(slide, L, T, W, H)$
    $anchor \leftarrow (L + W,\ T)$
    $comment \leftarrow \textsc{AddComment}(slide, anchor, j.\texttt{annotation})$
    $\textsc{Tag}(shape, \{\texttt{comment\_id} : comment.id\})$
    $\textsc{Tag}(comment, \{\texttt{shape\_id} : shape.id\})$
end for
SaveAndClose()

5.4 Output

The final output of the system is an annotated PowerPoint presentation file that integrates the generated annotations with the original slides. Each slide contains context-aware annotations provided as comments aligned with the corresponding content, enabling users to review or study the material with additional contextual information. After all processing steps have been completed, the annotated presentation is made available for download through the *UI*. Sample slides from the annotated output are shown in Figure 5.7.

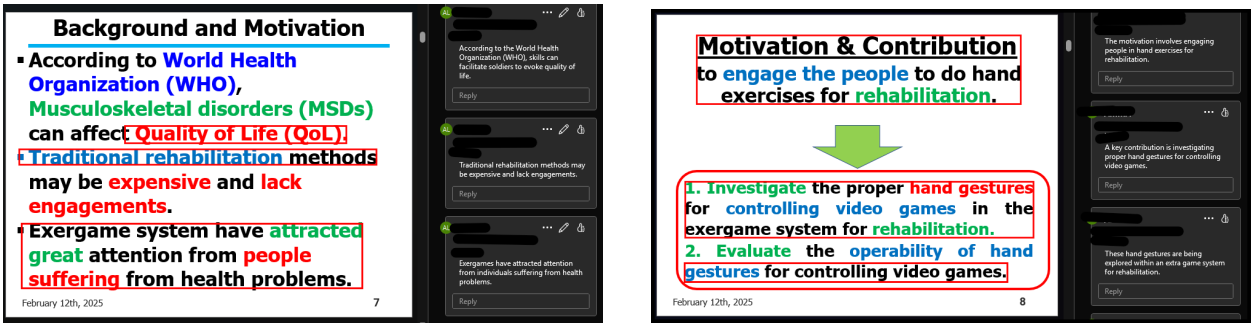


Figure 5.7: Sample annotated output demonstrating the integration of annotation inside the PowerPoint.

5.5 Evaluation

This section presents the experimental design and evaluation procedures conducted to assess the feasibility and performance of the proposed *Slide Annotation System*. It describes the participant selection criteria, data collection process, and analysis methods used in the study. Both quantitative and qualitative evaluations were carried out, consisting of participant-based testing to examine segmentation accuracy, annotation coherence, task performance, and overall usability, and presentation-owner testing to assess the validity and usefulness of the embedded annotations.

5.5.1 Experimental Design

The evaluation used a multi-stage measurement comprising three integrated components. This measurement began with a technical benchmark of algorithmic performance, followed by an expert-based validation involving presentation owners, and concluded with a participant-based usability study. Controlled conditions governed both the expert and participant components to ensure consistency across evaluations. The technical benchmark focused on assessing the segmentation robustness and alignment precision of the system. The expert validation examined the contextual accuracy and usefulness of the generated content, whereas the participant study evaluated task efficiency and overall system usability.

Algorithm Performance Evaluation. Distinct from the user study, a controlled benchmark evaluated the system's core algorithms. First, the study assessed Segmentation Robustness by testing the algorithm under the *Standard Condition* (original videos with visible slide numbers) serving as the primary benchmark, and the *Obscured Condition* (digitally masked regions) acting as a stress test to force reliance on the *visual fallback* mechanism.

Second, an ablation study evaluated Alignment Accuracy on a stratified subset of slides. This component tested the system's ability to correctly identify the specific visual anchor (such as a text block or figure) for each annotation against a human-verified ground truth, validating the spatial precision of the hybrid scoring module.

Presentation-Owner-Based Validation. This phase engaged five presentation owners whose recorded presentations served as the main dataset. Each presenter reviewed the automatically generated annotations corresponding to their own slides and evaluated whether the annotations were contextually accurate, factually correct, and practically useful. A structured evaluation form gathered their feedback through five-point Likert scale questions and optional written comments. This phase verified that the generated annotations were valid and meaningful before the study proceeded to the user testing phase.

Participant-Based Usability Study. The second phase recruited 37 participants, a sample size sufficient to identify usability issues [132]. The cohort comprised 15 undergraduate and graduate students majoring in Computer Science and 22 students from Multimedia Broadcasting, ensuring a diverse representation of users who frequently engage with technical presentation materials.

Each participant browsed the *Slide Annotation System* through the client-side web interface. A 5-minute training session familiarized participants with the full pipeline. During this training phase, the protocol directed participants to upload a sample presentation video, specify keywords, and observe the annotation generation process.

Following the training, participants advanced to the 10-minute measured task session. To ensure experimental consistency, the study assigned the same set of pre-processed annotated presentations to all participants rather than allowing personal file uploads. These videos represented the *Standard Condition* where slide numbers were visible. This controlled approach allowed for

direct comparison of task completion times and success rates across the cohort. Strict time limits encouraged efficient review behaviour.

During each session, task success rates and interaction logs were automatically recorded. Segmentation accuracy was validated against manually annotated slide boundaries or the *Ground Truth*. Annotation coherence was evaluated separately using an *LLM*-as-Judge approach, and user experience was assessed through the *System Usability Scale (SUS)* and a post-experiment feedback questionnaire. The summarized experimental procedure is shown in Figure 5.8.

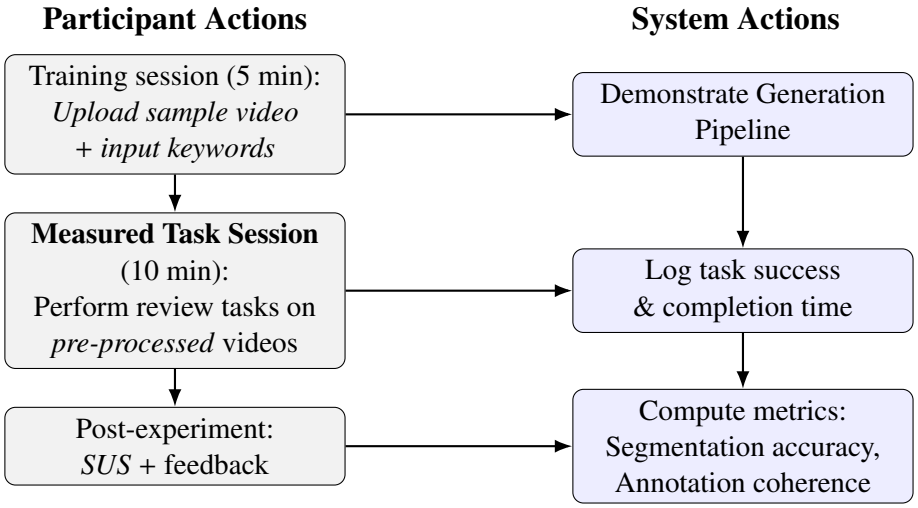


Figure 5.8: Experimental procedure distinguishing the training phase (upload demonstration) from the controlled task phase.

Following prior studies on slide summarization and multimodal interaction [43, 133, 134], four representative tasks were adopted to emulate typical reviewer behaviours. These tasks cover key aspects of the review process, including information retrieval, summarization, cross-checking, and slide browsing. Table 5.1 summarizes these tasks along with their corresponding evaluation purposes and example activities.

Table 5.1: Mapping of evaluation tasks and example activities.

Task	**Purpose**	**Example Activity**
Information retrieval	Evaluate how efficiently participants locate key content.	Locate the slide where the main contribution is introduced.
Summarization	Evaluate how well annotations convey content quality and coherence.	Summarize the main contributions of the work in 2–3 sentences.
Cross-checking	Evaluate consistency between narration and slide text.	Check whether the accuracy value mentioned in the narration matches the value on the slide.
Slide browsing	Evaluate ease of browsing and contextual orientation.	Identify the slide that introduces background concepts and report its title.

5.5.2 Evaluation Criteria

To comprehensively assess the system, this study adopted a multi-dimensional framework covering three core areas. Algorithmic performance was evaluated through two metrics. Segmentation accuracy measured using Precision, Recall, and F_1 Score under both standard (visible slide numbers) and obscured (masked *ROI*) conditions. And Alignment Accuracy assessed via an ablation study measuring the correct anchoring of annotations to visual content against a human-verified ground truth. Content quality was assessed via a dual approach, an automated *LLM*-as-Judge metric scoring factual consistency, coverage, specificity, and clarity (validated against human consensus using Weighted Cohen's κ and Spearman's ρ), and a qualitative expert review by presentation owners rating contextual validity and usefulness. Finally, User Experience was quantified through task performance metrics (success rate and completion time) and the standardized *System Usability Scale (SUS)*.

5.5.3 Experiment Materials

The experimental materials comprised pre-recorded academic presentation videos and their corresponding presentation documents. A workspace equipped with a GPU supported the multimodal processing and *LLM* inference throughout the evaluation.

5.5.3.1 Presentation Videos and Documents

The dataset includes five pre-recorded academic presentation videos and their corresponding documents, ranging from 8 to 27 minutes with an average of 30.4 slides per presentation. Sourced from natural online settings to ensure ecological validity, the recordings feature non-native speakers (average B2 proficiency) with Indonesian accents. These materials introduce specific technical challenges. Visually, the slides contain figures and mathematical formulas that strain standard *OCR* engines. Acoustically, the accented delivery and residual noise complicate transcription. Following segmentation, the sub-clips average 37.8 seconds in duration, ranging from 3 to 149 seconds.

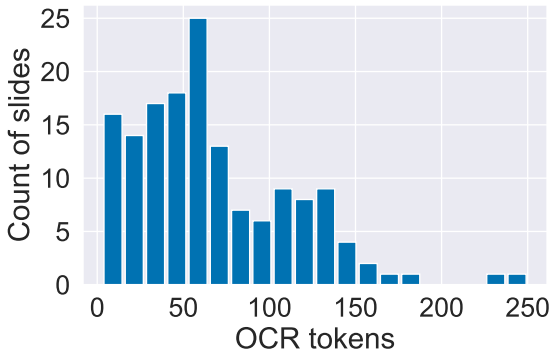
To support the robustness ablation study, digital masking of the slide number *ROI* in all five videos generated a derivative called the *Obscured Dataset*. This dataset simulates presentation styles lacking visible numbering, forcing the system to rely on the *visual fallback* mechanism. Figure 5.9 summarizes the detailed statistics for the videos and slides.

5.5.3.2 Hardware and Runtime Setup

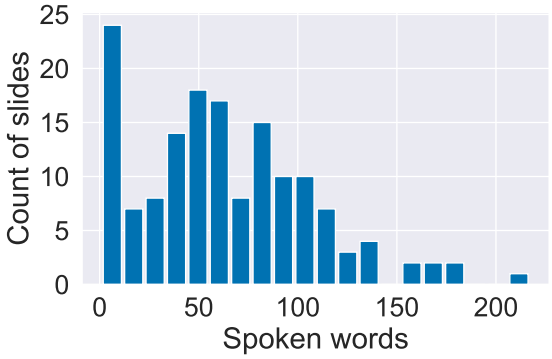
The system was executed on the following configurations. It was equipped with an NVIDIA RTX 3070 Ti GPU with 8GB of VRAM, an AMD Ryzen 9 CPU, and 32 GB of RAM, running Windows 11. The client interface was implemented in Streamlit v1.40, while the backend modules were developed using Python v3.10.16 with CUDA support.

5.5.3.3 Data Collection Procedure

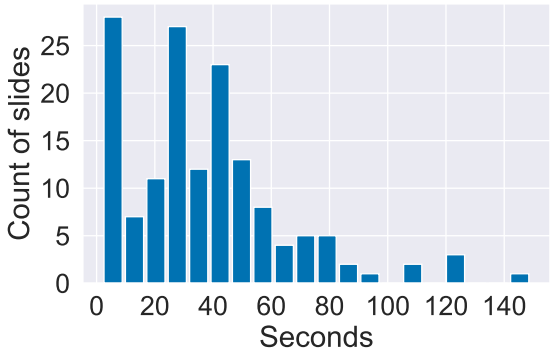
The system backend automated the data collection by logging user interaction events, including task initiation and completion times. To establish reliable baselines for the algorithmic evaluation, the study employed two manual annotation efforts. Two independent raters manually annotated



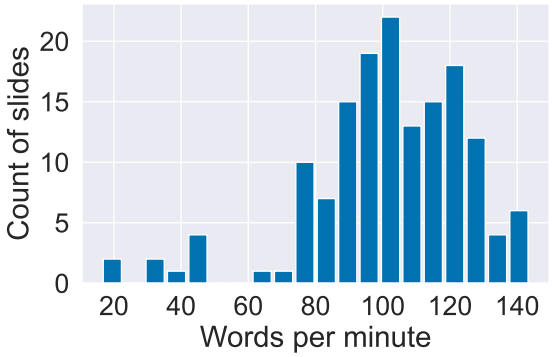
(a) Distribution of the number of *OCR*-extracted text tokens per slide.



(b) Distribution of spoken words per slide derived from the audio transcript.



(c) Duration of each slide segment (in seconds) after video segmentation.



(d) Speaking rate measured as words per minute for each slide segment.

Figure 5.9: Summary statistics of the recorded videos and corresponding presentation documents.

the slide transition timestamps for all five videos using the VLC media player's next frame feature. These timestamps served as the universal ground truth for both the Standard and Obscured condition evaluations. To validate the *LLM*-as-Judge, the same two raters independently scored a random subset of 30 generated annotations using the 5-point coherence rubric. Consensus discussions resolved any discrepancies between raters to ensure the high quality of the reference data [135].

Presenter Validation and User Feedback provided the subjective evaluation data. Prior to user testing, five original presentation owners reviewed their corresponding enriched presentations. Structured evaluation forms assessed the contextual validity and practical usefulness of the generated annotations via a 5-point Likert scale. Regarding User Feedback, upon completing the task session, participants responded to the *System Usability Scale (SUS)* questionnaire and provided open-ended feedback regarding their experience. Structured JSON files stored all outputs, including segmentation logs, annotations, and survey responses, for subsequent analysis.

5.5.4 Parameter Selection Analysis

Before evaluating the proposed method against state-of-the-art baselines, this section justifies the choice of two critical hyperparameters, the debouncing window size and the SSIM threshold (τ). The study calibrated these parameters on a pilot dataset to maximize system robustness.

5.5.4.1 Impact of Debouncing Window (Stage 1)

To handle transient *OCR* errors such as flickering cursors, the study conducted a controlled stress test using $n = 1000$ frames with simulated cursor noise. As detailed in Table 5.2, a 3-frame window (approx. 100 ms) proves sufficient to eliminate all stochastic noise. The analysis confirms a *False Positive Rate (FPR)* of 0.0% with a tight 95% confidence interval of [0.0, 0.4], indicating high reliability. Increasing the window to 5 frames provides no additional accuracy benefit but increases system latency without statistical justification.

Table 5.2: Debouncing Window Selection with 95% Confidence Intervals ($n = 1000$).

Window	**FP Count**	**FPR (95% *CI*)**	**Trade-off**
1 frame	47	4.7% [3.5, 6.2]	Too sensitive, triggers on noise
2 frames	4	0.4% [0.2, 1.0]	Still some false alarms
3 frames	**0**	**0.0% [0.0, 0.4]**	**Optimal Balance (100ms)**
4 frames	0	0.0% [0.0, 0.4]	Slower (133ms) with no benefit
5 frames	0	0.0% [0.0, 0.4]	Much slower, may miss transitions

5.5.4.2 Impact of SSIM Threshold (Stage 2)

Following the primary detection, the visual fallback mechanism was calibrated on a dataset containing $n = 250$ annotated slide transitions and $n = 800$ static sequences. The analysis examines the trade-off between *True Positive Rate (TPR)* and *False Positive Rate (FPR)* by sweeping τ from 0.75 to 0.95. As shown in Table 5.3, lower thresholds at 0.75 result in excessive false alarms with *FPR* value of 3.1%, *CI* [2.0, 4.4]). Conversely, overly strict thresholds at 0.95 significantly degrade the F_1 score to 0.92. Consequently, the system employs $\tau = 0.85$ as the optimal operating point. This configuration achieves a near-perfect F1 score of 1.00 (95% *CI* [0.99, 1.00]) and maintains a statistically robust 0.0% *FPR*.

Table 5.3: SSIM Threshold Selection with 95% Confidence Intervals.

τ	**TPR (95% *CI*)**	**FPR (95% *CI*)**	**F1 Score (95% *CI*)**	**Trade-off**
0.75	100.0% [98.5, 100]	3.1% [2.0, 4.4]	0.95 [0.94, 0.96]	Too loose
0.80	99.6% [97.8, 99.9]	0.4% [0.1, 1.1]	0.99 [0.98, 1.00]	Some FP
0.85	**99.6% [97.8, 99.9]**	**0.0% [0.0, 0.5]**	**1.00 [0.99, 1.00]**	**Optimal**
0.90	98.0% [95.4, 99.1]	0.0% [0.0, 0.5]	0.99 [0.98, 1.00]	Misses some
0.95	84.8% [79.8, 88.7]	0.0% [0.0, 0.5]	0.92 [0.89, 0.94]	Misses many

Note: Confidence intervals for TPR/FPR calculated via Wilson score. CIs for F1 score calculated via non-parametric bootstrapping ($B = 10,000$).

5.5.5 Slide Segmentation Accuracy

To evaluate the segmentation accuracy, the system-generated slide boundaries were compared against manually annotated *ground-truth (GT)* data across five presentation videos. Each boundary represents a timestamp (in seconds) marking the transition between two consecutive slides within

the presentation video. The segmentation performance was measured using precision, recall, and F_1 *score*.

Several baseline methods were included for comparison. *Uniform Segmentation* divides the video into equal-length sub-clips regardless of content, *Text Tiling* [136, 137] applies the topic-based text segmentation to the presentation transcripts. For visual baselines, this study compared it against *Structural Similarity Index (SSIM)-based frame differencing* and *Histogram-based detection*. Additionally, *PySceneDetect* v0.6.7 [138], as a popular open-source scene detection library, was also evaluated. To determine the optimal sampling rate for the proposed hybrid two-stage detector, this study evaluated its performance under two configurations, a high-precision 1-second interval and the efficiency-focused 3-second interval.

Table 5.4 presents the aggregated quantitative results. While the *uniform segmentation*, *Text-Tiling*, and *Histogram* baselines proved ineffective ($F_1 < 0.18$), the visual *SSIM* baseline achieved a high accuracy (0.892 ± 0.047) but incurred a substantial processing overhead. Conversely, PySceneDetect offered the fastest execution (58s) but suffered from significant under-segmentation ($F_1 = 0.743$). The proposed 1-second configuration achieved the highest performance ($F_1 = 0.955 \pm 0.023$), yet required the longest processing time. Consequently, the 3-second configuration emerges as the optimal trade-off. It leverages the *OCR* precision to maintain a macro-averaged F_1 score (0.879 ± 0.024) comparable to *SSIM*, while significantly reducing computation time to 1:49.

Table 5.4: Segmentation accuracy comparison across all 5 videos (Mean ± SD).

Method	**Precision**	**Recall**	**F_1 score**	**Avg. Time (mm:ss)**
Uniform Segmentation	0.182 ± 0.002	0.167 ± 0.002	0.174 ± 0.008	00:07 ± 00:01
Text Tiling	0.062 ± 0.001	0.030 ± 0.001	0.041 ± 0.001	00:04 ± 00:01
SSIM	0.913 ± 0.016	0.875 ± 0.017	0.892 ± 0.047	03:52 ± 01:50
Histogram	0.079 ± 0.002	0.807 ± 0.002	0.144 ± 0.003	04:50 ± 02:15
PySceneDetect v0.6.7	0.806 ± 0.009	0.691 ± 0.009	0.743 ± 0.013	**00:58 ± 00:28**
Proposed (1 s)	**0.971 ± 0.009**	**0.938 ± 0.008**	**0.955 ± 0.023**	06:34 ± 03:15
Proposed (3 s)	0.900 ± 0.011	0.860 ± 0.011	0.879 ± 0.024	01:49 ± 00:54
Ground Truth (GT)	1.000	1.000	1.000	–

Further analysis reveals that the performance gap between the 1-second and 3-second configurations stems primarily from timestamp granularity rather than missed detections. Both configurations identified identical structural changes. However, the wider sampling interval introduces a slight temporal offset. For example, the transition marking at 15s produced by the 3-second configuration versus the ground truth of 13.5s. Consequently, the lower F_1 score reflects alignment precision rather than segmentation failure.

To provide a qualitative assessment, Figure 5.10 visualizes and compares the segmentation outputs across different methods. As illustrated, the *Proposed (3 s)* configuration generates segment boundaries that align closely with the ground truth timestamps. In contrast, while *PySceneDetect* offers rapid processing, it misses several key transitions, resulting in merged slides. Balancing segmentation accuracy with runtime efficiency, the 3-second configuration serves as the default parameter for the final implementation of the *Slide Annotation System*.

To analyze the stability of the primary *OCR* component, this study examined 2,229 sampled frames across the five presentation videos. The analysis revealed that raw *OCR* readouts were empty or unstable in 4.2% of processed frames, primarily attributed to title slides, visual transition effects, or cursor occlusions. While the temporal debouncing filter effectively mitigates these

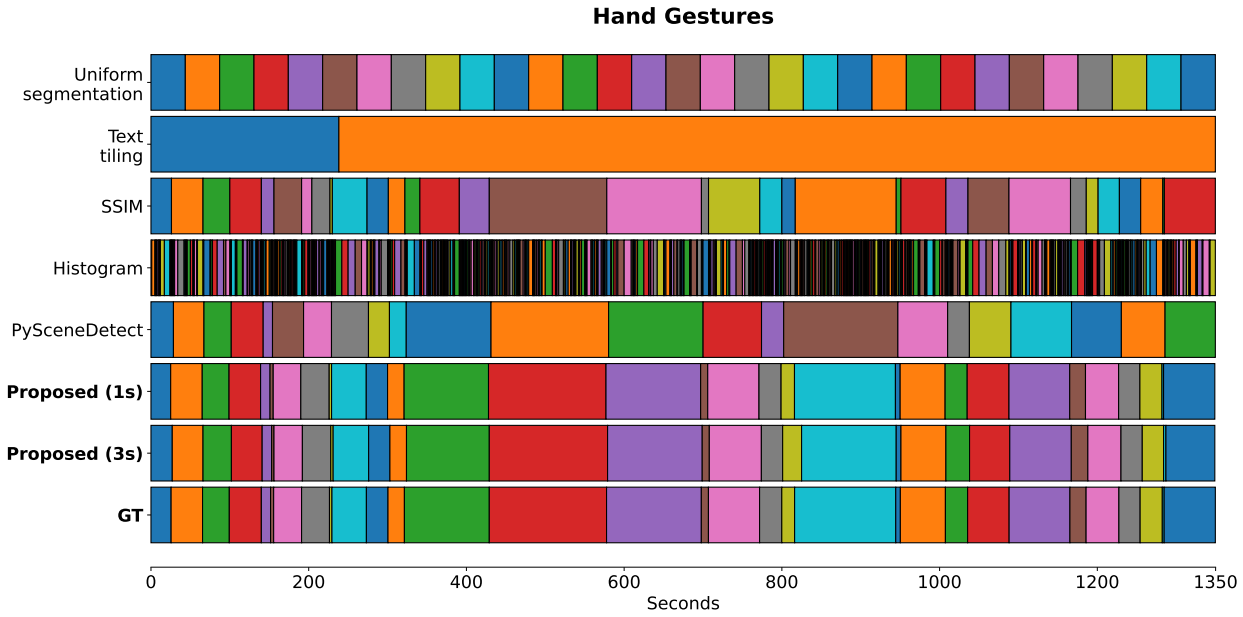


Figure 5.10: Visual comparison of slide segmentation outputs on a sample presentation video.

transient artifacts, it remains insufficient for scenarios where slide numbers are entirely absent.

To evaluate system resilience under such conditions, the *Obscured Dataset*, where slide numbers were digitally masked facilitated an ablation study. As Table 5.5 shows, the *OCR*-only baseline yielded no valid detections ($F_1 = 0.0$) under this condition, as it relies heavily on explicit digit recognition. In contrast, the proposed hybrid system successfully triggered the *visual fallback* mechanism, achieving a robust mean F_1 score of 0.885 ($SD = 0.011$, 95% *CI* [0.871, 0.899]). Notably, this performance rivals the *SSIM* baseline reported in Table 5.4 (0.892), yet operates within the more computationally efficient hybrid architecture. This statistical consistency demonstrates that the two-stage design enables the system to remain functional and reliable, even in scenarios where slide numbers are absent or occluded.

Table 5.5: Ablation study evaluating system robustness on the *Obscured Dataset*. Comparison of the *OCR*-only baseline vs. the proposed Hybrid Two-Stage detector. Mean ± SD.

Method	**Precision**	**Recall**	**F_1 score**	**Status**
***OCR*-Only Baseline**	0.000	0.000	0.000	*Failed*
Hybrid Two-Stage (Proposed)	**0.865 ± 0.010**	**0.907 ± 0.011**	**0.885 ± 0.011**	*Functional*

5.5.6 LLM-as-Judge for Annotation Coherency

Annotation coherence refers to the logical consistency and contextual alignment of the generated annotations in relation to both the slide text and the spoken narration. To enable objective and scalable assessment of this quality, this study adopted an *AI*-assisted evaluation approach based on the *LLM*-as-Judge framework [139]. Specifically, *DeepSeek v3* was employed to evaluate annotations generated by gemma3:4b Q4_K_M quantization, selected for its demonstrated reasoning capability [140]. This subsection presents the configuration and validation of the *LLM*-as-Judge framework used to assess annotation coherence.

To ensure consistency and reproducibility, the judge *LLM* was assigned the role of evaluating the coherence and faithfulness of generated slide annotations. It processed three aligned inputs consisting of the slide text extracted through *OCR*, the spoken transcription from *ASR*, and the generated annotation. Coherence was rated on a structured rubric covering four criteria using a five-point Likert scale, with 1 indicating incoherent and 5 indicating fully coherent. The criteria included factual consistency, coverage of key ideas, specificity, and linguistic clarity. The verbatim definitions for each score level are provided in Appendix 8.3. The results were stored in a standardized JSON format containing the fields *score*, *rationale*, and *error_tags*. An overview of the *LLM*-as-Judge configuration is illustrated in Figure 5.11.

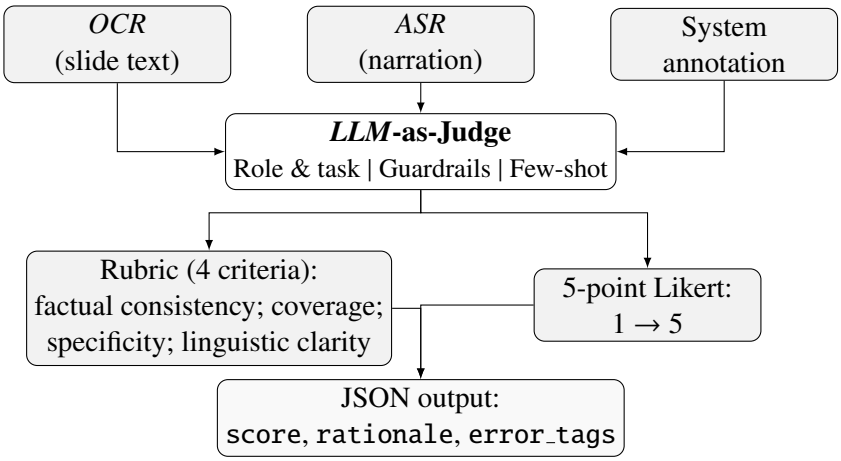


Figure 5.11: Structured setup for the *LLM*-as-Judge evaluation of generated annotations.

A comparison between the *LLM*-based evaluation and human judgments was conducted to verify the reliability of this approach. A representative subset of 30 annotation samples, spanning the full 1–5 rating scale, was evaluated by two human judges using the same rubric. First, to validate the human ground truth, the inter-rater reliability between the two human judges were calculated. They achieved a Weighted Cohen's κ of 0.655 (95% *CI* [0.39, 0.82]) and a Spearman's ρ of 0.745 (95% *CI* [0.49, 0.91]), indicating substantial agreement. Human scores were then averaged to obtain a consensus rating [141]. To measure the agreement between the *LLM* and this consensus using Cohen's κ, the averaged human scores were rounded to the nearest integer to produce the necessary discrete categories.

As shown in Table 5.6, the *LLM*'s scores demonstrated strong alignment with the human consensus. The system achieved a Weighted κ of 0.705 (95% *CI* [0.48, 0.85]), slightly exceeding the human-human agreement, and a strong rank correlation ($\rho = 0.836$, 95% *CI* [0.67, 0.93]). Although the *LLM* exhibited a slightly more conservative scoring tendency (average score = 3.09) compared to human ratings (average score = 3.25), it maintained consistent alignment with human quality rankings. Given the limited sample size ($n = 30$) and the non-parametric nature of Likert scale data, bootstrap resampling with $B = 1000$ iterations were employed to estimate the 95% *Confidence Intervals (CI)*. This method was chosen to ensure the stability of the inter-rater reliability metrics without relying on assumptions of normality. These results validate the reliability of the *LLM*-as-Judge framework for large-scale annotation coherence evaluation.

Table 5.6: Annotation coherence reliability analysis on 30 samples. Metrics include categorical agreement (Weighted Cohen's κ) and rank correlation (Spearman's ρ) with 95% *CI*.

Comparison Pair	**Avg. Score Diff.**	**Cohen's κ (95% *CI*)**	**Spearman's ρ (95% *CI*)**
Human 1 vs. Human 2	0.12	0.655 [0.39, 0.82]	0.745 [0.49, 0.91]
***LLM* vs. Human Consensus**	0.16	**0.705** [0.48, 0.85]	**0.836** [0.67, 0.93]

Note. Human consensus scores were rounded to the nearest integer for κ calculation. CIs computed via bootstrap resampling (n = 1000).

5.5.7 Annotation Alignment Accuracy

To evaluate alignment accuracy, an ablation study on a diverse subset of 30 slides assessed the system's performance. For each generated annotation, presentation owners identified the most specific visual content in the slide to serve as the ground truth. The analysis compared three configurations, the Cosine-only baseline, the Proposed Hybrid Score, and the Full Pipeline with Cross-Encoder re-ranking.

As Table 5.7 summarizes, the Cosine-only baseline encountered difficulties with dense academic content, achieving an accuracy of only 53.3% (95% *CI* [36.1%, 69.8%]) due to semantic ambiguity between visually similar elements. The integration of spatial and numeric cues in the Hybrid Score improved accuracy to 76.7% (95% *CI* [59.1%, 88.2%]), confirming the critical role of non-semantic signals in filtering irrelevant candidates. Finally, the Full Pipeline achieved 90.0% accuracy (95% *CI* [74.4%, 96.5%]), demonstrating that cross-encoder re-ranking robustly resolves remaining subtle ambiguities.

Table 5.7: Ablation study of Annotation Alignment Accuracy (N = 30 slides, stratified across dense text, figures, and formulas). Accuracy is reported with 95% *CI*.

Configuration	**Accuracy (95% *CI*)**	**Key Observation**
Cosine-Only Baseline	53.3% [36.1, 69.8]	Low confidence on dense academic slides.
Hybrid Score (w/o Re-rank)	76.7% [59.1, 88.2]	Recovered 7 failures via numeric/spatial cues.
Full Pipeline (Proposed)	**90.0% [74.4, 96.5]**	Resolved final semantic ambiguities.

5.5.8 System Latency and Deployability

To assess the feasibility of real-world deployment, this study measured the end-to-end pipeline latency averaged across the five test videos. As detailed in Table 5.8, the full pipeline required an average processing time of 7 minutes and 13 seconds (SD = 98 s). The computational load was distributed primarily across the video processing and *LLM* inference stages. Notably, the use of gemma3:4b Q4_K_M quantization for both keyword extraction and annotation generation contributed equally to the inference latency. Despite these multiple distinct processing passes, the system maintained a real-time factor of 0.35× and an average throughput of 18.0 seconds per slide. This confirms that the pipeline operates efficiently within the latency constraints required for offline lecture archiving.

Table 5.8: End-to-end pipeline runtime breakdown averaged across $N = 5$ videos. Data is reported as Mean ± SD.

Pipeline Stage	**Duration (Mean ± SD)**
1. Video Segmentation	1 min 49 s ± 25 s
2. *Automatic Speech Recognition (ASR)*	1 min 35 s ± 25 s
3. Keyword Extraction (LLM)	1 min 15 s ± 18 s
4. Slide Content Extraction (OCR)	5 s ± 2 s
5. Annotation Generation (LLM)	1 min 18 s ± 20 s
6. Annotation Alignment	1 min 07 s ± 15 s
7. Slide Reconstruction	4 s ± 1 s
Total End-to-End Runtime	**7 min 13 s ± 1 min 38 s**

5.5.9 Validity and Usefulness of Embedded Annotations

The validity of embedded annotations refers to the degree to which the generated annotations accurately and faithfully reflect the intended meaning of the original presentation content. Usefulness reflects the practical value of these annotations in supporting comprehension of the content. Since the original presenters possess the most comprehensive understanding of their materials, they were assigned as expert evaluators to assess both the validity and usefulness of the embedded annotations.

The assessment employed a ten-item questionnaire using a five-point Likert scale, ranging from 1 for Strongly Disagree to 5 as Strongly Agree. The questionnaire was divided into two categories, validity that covers factual accuracy, relevance, alignment with the slide content and usefulness, which covers clarity, time efficiency, and potential for reuse during review sessions. Each presenter reviewed the annotated version of their own presentation and rated each item accordingly.

The results yielded a high mean validity score of 4.48 ($SD = 0.18$, 95% CI [4.26, 4.70]) and a mean usefulness score of 4.36 ($SD = 0.36$, 95% CI [3.92, 4.80]). The overall average across all items was 4.42 ($SD = 0.26$). Notably, 94% of the questionnaire items were rated 4 or higher, indicating that the presenters perceived the annotations as both factually accurate and contextually appropriate. Minor qualitative issues were noted, including occasional abbreviation mismatches such as "FPLAS" recognized as "f+" and suggestions to allow multiple annotations within a single paragraph to better capture complex explanations.

These findings confirm that the generated annotations effectively preserve the intended meaning of the original materials while providing additional value for later review. They also ensure that the annotated slides used in the subsequent usability testing accurately reflected real-world system performance. A summary of the validity and usefulness ratings across all presenters is shown in Figure 5.12.

5.5.10 Task Performance for Success Rate & Completion Time

The evaluation assessed task performance to determine the system's effectiveness and efficiency in supporting review activities across two participant groups with distinct educational backgrounds. The first group comprised 15 students majoring in Computer Science (CS), while the second included 22 students from Multimedia Broadcasting (MB).

Prior studies on slide summarization, content retrieval, and multimodal interaction in recorded

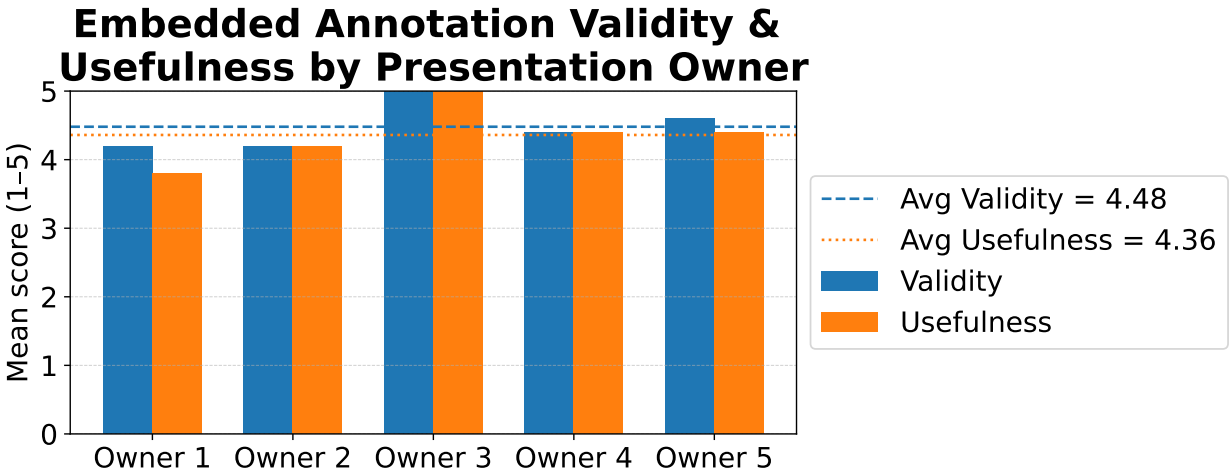


Figure 5.12: Summary of presenter ratings on the validity and usefulness of embedded annotations.

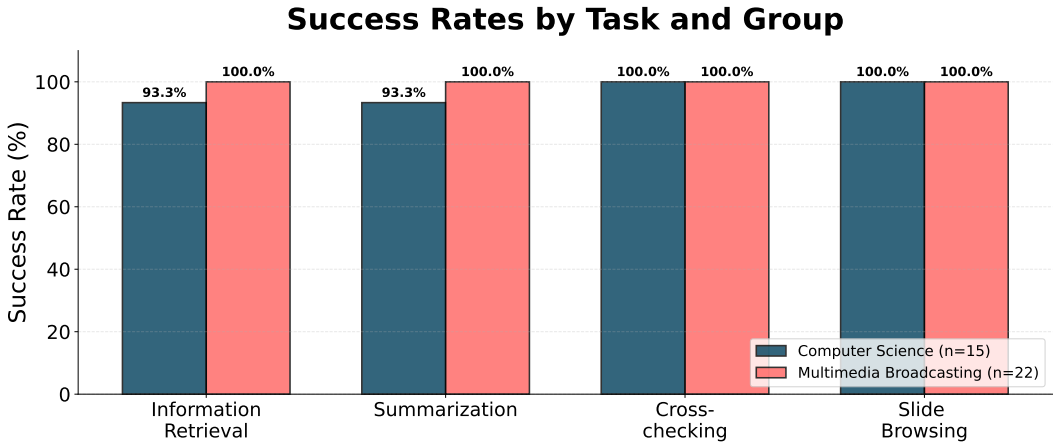
presentations guided the selection of the four task categories [43, 133, 134]. The *success rate*, defined as the proportion of participants who successfully completed each task within the allotted 10 min, quantified effectiveness. Simultaneously, the *completion time*, calculated as the average duration (in seconds) among successful participants, measured efficiency. A 10-min time limit simulated a realistic review scenario. The system automatically logged both metrics, and Figure 5.13 summarizes the results. Prior to analysis, task completion time distributions were assessed for normality using the Shapiro-Wilk test ($p > 0.05$) and for homogeneity of variance using Levene's test ($p > 0.05$). Visual inspection of the boxplots on Figure 5.13b further confirmed the absence of extreme outliers. Given that the assumptions for parametric testing were met, this study report independent samples t-tests ($df = 35$) accompanied by Cohen's d to estimate the magnitude of performance differences.

The proportion of participants who completed tasks within the 10-minute limit defined effectiveness. The system yielded a high aggregate success rate of 94.6% (35/37), with a 95% *CI* of [82.3%, 98.5%]. As Figure 5.13a demonstrates, both groups attained high success rates (93%–100%) across all four task categories, confirming the system's effectiveness in supporting the review process regardless of educational background. However, the analysis of completion times revealed significant task-specific performance patterns aligned with domain expertise.

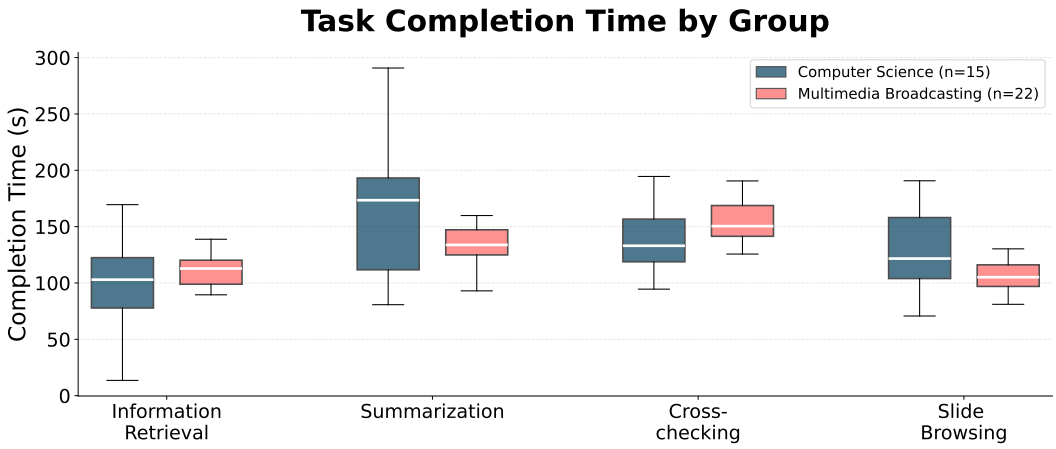
The average duration in seconds for successful trials quantified efficiency, serving to identify performance disparities between participant groups. Table 5.9 summarizes detailed statistical comparisons, including Means, Standard Deviations (SD), and significance levels derived from independent t-tests. The results highlight significant domain-specific advantages, with Multimedia Broadcasting (MB) students demonstrating superior speed in browsing tasks, while Computer Science (CS) students excelled in data verification.

5.5.11 System Usability and Post-Experimentation Feedback

Usability was assessed through the standardized *System Usability Scale (SUS)* [142], administered immediately after participants completed the tasks. The *SUS* is a ten-item questionnaire that produces a usability score on a 0–100 scale, where higher values indicate better usability [143]. In addition to the *SUS* questionnaire, participants were asked to provide open-ended feedback on strengths, limitations, and suggestions for improvement. Both the *SUS* questionnaire and the open-



(a) Success rates (%) by task category and participant group (CS vs. MB).



(b) Boxplots of completion times (seconds) for successful task completion, grouped by participant group and task category.

Figure 5.13: Task performance comparison between Computer Science (CS) and Multimedia Broadcasting (MB) student groups across four review task categories.

Table 5.9: Comparison of Task Completion Times (Efficiency). Statistical differences were assessed via independent t-tests. Effect sizes (Cohen's d) indicate practical significance.

Task Category	**CS Group (s)**	**MB Group (s)**	t**-stat**	**Sig.** (p)	**Cohen's** d
Slide Browsing	130.7 ± 21.4	**106.6 ± 18.2**	3.68	< **0.001**	**1.23** (Large)
Summarization	160.7 ± 25.3	**136.0 ± 22.1**	3.10	**0.004**	**1.05** (Large)
Cross-Checking	**139.0 ± 19.8**	155.5 ± 24.2	−2.15	**0.038**	**0.73** (Med)
Info. Retrieval	111.6 ± 28.5	101.0 ± 30.1	1.05	0.320	0.36 (Small)

ended feedback were collected using a digital form. The questions for the open-ended feedback are illustrated in Table 5.10, which documents the qualitative dimensions of participant experience.

The system achieved a mean *SUS* score of 80.5 ($SD = 6.7$), with a 95% *CI* of [78.3, 82.7]. Reliability analysis yielded a Cronbach's α of 0.83, indicating high internal consistency in the participant responses. The distribution of individual scores is illustrated in Figure 5.14. Qualitative feedback revealed recurring strengths, particularly the ease of browsing, interface clarity, and

Table 5.10: Open-ended feedback prompts after *SUS* questionnaire.

Category	**Question**
Strengths	What aspects of the system did you find most useful or effective?
Limitations	What aspects of the system did you find confusing, frustrating, or difficult to use?
Suggestions	If you could improve one thing about the system, what would it be?
Overall impression	How would you describe your overall experience with the system in one or two sentences?

the time efficiency gained through automated annotations. As one participant explained, *"Slide navigation was very smooth and helped me focus on the important parts."*

However, participants also identified areas for refinement, most notably annotation precision and system responsiveness. Typical remarks included *"Some annotations were slightly misaligned, especially around numerical content,"* and *"There was a minor delay when generating PowerPoint files, which made the process feel a bit slow."* Several users further proposed practical enhancements, such as improved text alignment, automatic cleaning of extracted text, and lighter export options to streamline the overall workflow.

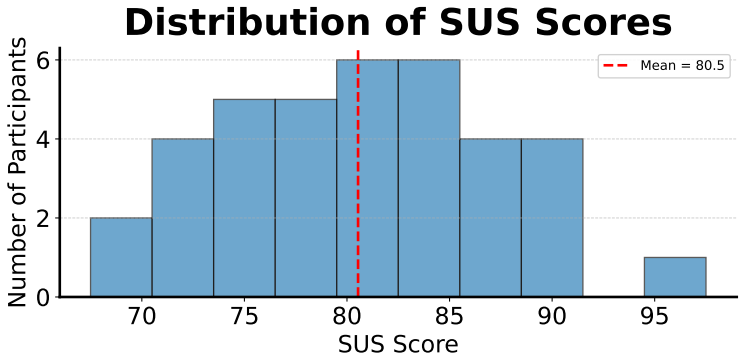


Figure 5.14: Distribution of *System Usability Scale (SUS)* scores reported by participants, illustrating overall usability ratings and individual variability.

These results indicate that the system was generally well-received, with some variation in user perception reflecting users' differing levels of familiarity with digital systems and expectations of annotation accuracy. Since participants were primarily computer science and information systems students, a degree of technology-oriented bias cannot be ruled out, and the findings may not fully generalize to broader user populations. The combination of strong usability scores and constructive feedback supports the system's overall feasibility for presentation review.

5.6 Discussion

The evaluation results demonstrate that the proposed *Slide Annotation System* effectively integrates multimodal inputs, including hybrid slide segmentation, speech-to-text transcription, and *LLM*-driven annotation generation, to produce coherent and contextually accurate slide-level annotations. Across all evaluation criteria, the system exhibited high technical accuracy, semantic

reliability, and strong user acceptance, indicating its practical feasibility for presentation review and the enrichment of educational content.

Technical and User Performance. The segmentation function achieved an effective balance between accuracy and computational efficiency. The proposed hybrid two-stage detector achieved a macro-average F_1 score of 0.879 ($SD = 0.024$, 95% CI [0.849, 0.909]) on the standard dataset. Notably, the system maintained functional performance with an F_1 score of 0.885 ($SD = 0.011$, 95% *CI* [0.871, 0.899]) even when visual slide markers were digitally obscured, overcoming the limitations of pure *OCR* approaches. Regarding annotation quality, the *LLM*-as-Judge evaluation showed substantial agreement with human ratings ($\kappa = 0.705$, $\rho = 0.836$), confirming the reliability of the automated assessment [139]. Complementarily, expert validation by presentation owners yielded high validity scores of 4.48/5 ($SD = 0.18$, 95% *CI* [4.26, 4.70]), confirming that annotations were contextually aligned with the presenters' intent. In terms of usability, the average *SUS score* of 80.5 ($SD = 6.7$, 95% *CI* [78.3, 82.7]) indicates excellent usability, with participants demonstrating high task success rates. Tasks involving information browsing were completed significantly faster by multimedia students, while summarization tasks reflected cognitive load patterns consistent with multimedia learning theory [144].

Comparative Positioning. To contextualize the contribution of the proposed system, Table 5.11 presents a qualitative comparison against standard commercial meeting assistants and educational video indexing platforms. Existing commercial tools such as Otter.ai and Zoom AI are primarily audio-centric. While they provide effective transcription, they lack visual alignment, meaning they cannot relate information to specific slide regions. Conversely, video indexing platforms such as Panopto utilize *OCR* for keyword retrieval but typically lack the generative capabilities to synthesize explanatory notes. The proposed system fills this gap by integrating generative *LLM* capabilities with hybrid spatial-temporal analysis, enabling precise annotations that are contextually aligned with the corresponding slide.

Table 5.11: Qualitative feature comparison between the proposed system and existing video analysis paradigms.

Feature Capability	**Meeting Assistants (Otter.ai, Zoom)**	**Video Indexing (Panopto)**	**Proposed System**
Input Modality	Audio Only	Audio + Visual	Audio + Visual
Transcription (ASR)	✓	✓	✓
Generative Summarization	✓(Global)	×	✓(Slide-Level)
Slide Text Extraction (OCR)	×	✓	✓
Visual Alignment	×	×	✓

Note. **Visual Alignment** refers to the ability to utilize slide changes to segment and associate spoken content with specific presentation slides.

Pedagogical Implications and Recommendations. Beyond technical utility, the system supports a shift from passive watching to active learning. At the course level, this study recommend using the system to break long lectures into short, focused clips. These should be assigned for pre-class viewing, allowing class time to be used for problem-solving rather than listening. To help students manage their study time, these clips can be paired with simple to-do checklists and quick surveys where students can flag confusing topics. At the staff level, adoption depends on making the process easy for instructors. Institutions should provide simple recording kits such as studio-in-a-box and standardized slide templates that are easy for the *AI* to read, along with a shared library

of good examples to guide slide design. Finally, at the institutional level, clear policies are needed regarding who owns the video data and how long it is stored. We also advise using the system's data, such as monitoring where students stop watching to identify boring or difficult sections and improve the course content each semester.

Scalability and Computational Optimization. For long-term adoption and deployment on larger video corpora, optimizing runtime performance is critical. The current implementation relies on GPU acceleration for the *LLM* and *ASR* modules, which may present a cost barrier for large-scale processing. To address this, future iterations can employ model quantization to reduce memory footprint without compromising generation quality. Furthermore, handling a large-scale repository of videos would require transitioning to an indexed architecture. Integrating vector databases would allow for efficient storage and retrieval of slide embeddings across thousands of presentations, ensuring the system remains responsive as the dataset grows.

Limitations and Future Work. Despite these encouraging results, several limitations remain. The participants primarily consisted of technically proficient students, which may have biased usability perceptions. While the hybrid detector significantly improves robustness, the *visual fallback* mechanism (Stage 2) can occasionally be sensitive to animation movement or in-slide embedded video playback, which may trigger false positive transitions. Additionally, the alignment ablation study revealed a 10% error rate in spatial anchoring, even with cross-encoder re-ranking. This algorithmic limitation aligns with participant feedback regarding "occasional misalignments" in dense slide regions

To address this, future iterations will explore audio-based segmentation cues and apply *Digital Signal Processing (DSP)* to further refine detection in highly dynamic scenarios. User feedback also highlighted challenges in aligning numerical content. Future iterations will implement a *Human-in-the-Loop (HITL)* mechanism allowing users to manually adjust annotation placement. Finally, as the current evaluation focused primarily on textual coherence, future work will incorporate *Vision-Language Models* such as *GPT-4V* and objective metrics like *ROUGE* to explicitly assess visual consistency and spatial alignment accuracy.

Nevertheless, the findings collectively highlight that multimodal fusion with *LLM*-based generation can transform presentation recordings into semantically enriched materials, offering practical potential for intelligent educational review systems.

5.7 Summary

This study presented a multimodal *slide annotation system* that automatically generates coherent and contextually aligned annotations from recorded presentation videos by integrating hybrid slide segmentation, speech-to-text transcription, and *LLM*-based annotation generation. Experimental results demonstrated high segmentation accuracy (macro-average $F_1 = 0.879$, $SD = 0.024$, 95% $CI[0.849, 0.909]$), precise annotation alignment (90.0% accuracy), and reliable annotation coherence validated by both human and automated evaluation. User evaluations further revealed high satisfaction, with an average *SUS score* of 80.5 ($SD = 6.7$, 95% *CI* [78.3, 82.7]), confirming the system's feasibility for enhancing the review process by transforming recorded presentations into interactive and semantically enriched PowerPoint materials. While improvements in text normalization and runtime efficiency remain, this work establishes a foundation for scalable, semantically coherent documentation of presentation content. Future research will focus on integrating audio-based segmentation cues to handle dynamic visual scenarios and extending the system's capabilities to support multilingual annotation generation.

Chapter 6

Discussion and Comparative Analysis of Proposed Frameworks

6.1 Evolution of Architecture: From Modular to End-to-End Multimodal

The development of the software tools presented in this thesis reflects a deliberate progression from specific acoustic interfaces to fully integrated multimodal systems. This evolution was driven by the increasing need to not just transcribe content, but to semantically ground it within its visual context.

The research began by establishing a robust acoustic foundation through the design and implementation of the *Audio-to-Text Conversion Software*. The initial phase focused on utilizing the *Whisper* model to create a unified interface for transcription. This software was designed to resolve the preparatory dependency found in raw model usage, creating a stable server-side architecture capable of handling input normalization and high-fidelity transcription. Once this acoustic layer was validated, the research focus shifted toward handling multi-stream data.

Building upon this foundation, the second phase expanded the architecture into the *Meeting Minutes Generation Tool*. This system integrated various open-source models, including summarization, keyword extraction, and *Optical Character Recognition (OCR)* to process video and textual information. However, the approach was modular, meaning audio and visual streams were processed separately. To prevent data misalignment, an information correlation mechanism using *OCR* and regular expressions was required to link text to slides. While effective, this modularity highlighted a critical limitation: the lack of intrinsic contextual awareness between the modalities.

The final phase addresses this limitation through the development of the fully integrated *Slide Annotation Generation Tool*. Unlike the previous modular approach that generated detached summaries, this system employs a hybrid two-stage detector and a multimodal *LLM* to perform simultaneous visual, auditory, and textual analysis. This architectural shift allows the system to identify slide boundaries and generate concise, context-aware annotations that are directly aligned to their specific visual locations. This integration ensures that the generated content is not just textually accurate, but spatially and semantically coherent.

6.2 Performance Trade-offs

Throughout this trajectory, each architectural approach necessitated specific trade-offs between computational efficiency, scalability, and granular accuracy. In the *Audio-to-Text Conversion Software*, the priority was concurrency versus latency. The *small.en* model demonstrated a desirable balance, achieving transcription accuracy comparable to larger models while significantly reducing execution time. This allowed the system to handle up to 180 concurrent users with an average response time of 309 ms. However, stress testing revealed that beyond 186 users, response times increased significantly, identifying a clear bottleneck under heavy load that dictated the need for optimized inference in later stages.

In the *Meeting Minutes Generation Tool*, the trade-off shifted to scope and precision. While the *audio-to-text* conversion remained strong, the addition of summarization and keyword extraction introduced noise. Because the text data from meetings is lengthy, summarization was necessary. However, *ROUGE-N* scores indicated that standard *NLP* models struggled to generate cohesive summaries without visual context, resulting in functional but occasionally disjointed outputs.

Consequently, the *Slide Annotation Generation Tool* accepted a higher computational cost to achieve maximum semantic precision. The hybrid two-stage detector achieved a mean F_1 score of 0.879, maintaining functional performance even when visual slide markers were digitally obscured. The trade-off for this high accuracy (90.0% alignment) is the reliance on GPU acceleration for the *LLM* and *ASR* modules, which, unlike the lightweight architecture of the *Audio-to-Text Conversion Software*, presents a scalable barrier for large-scale processing.

6.3 Comparison with SOTA

The proposed framework contributes distinct advancements compared to both academic baselines and commercial products, particularly in the domain of visual grounding. Regarding acoustic performance, the system parallels state-of-the-art benchmarks. The *Whisper* model employed in the *Audio-to-Text Conversion Software* demonstrated performance closely matching human professional transcribers. Similarly, the *Meeting Minutes Generation Tool* yielded higher transcription accuracy than existing proprietary systems, confirming that open-source stacks can rival commercial engines in raw text generation.

However, the key differentiator lies in multimodal integration. Existing commercial tools like *Otter.ai* and *Zoom AI* are primarily audio-centric. While they excel at transcription, they lack visual alignment, meaning they cannot relate spoken information to specific slide regions. Users receive a wall of text but no visual context. Conversely, video indexing platforms such as Panopto utilize *OCR* for keyword retrieval but typically lack the generative capabilities to synthesize explanatory notes.

The proposed Slide Annotation Generation Tool bridges this gap. By integrating generative *LLM* capabilities with hybrid spatial-temporal analysis, it enables annotations that are not only descriptive but are contextually aligned (90.0% accuracy) with the corresponding slide. This capability to ”place” information spatially is absent in current standard tools.

6.4 Summary

This dissertation presents a comprehensive trajectory for automated presentation analysis, moving from accessibility to deep semantic integration. The *Audio-to-Text Conversion Software* estab-

lished a scalable web application that democratized access to *Whisper* models, capable of handling 471.5 requests per second. The *Meeting Minutes Generation Tool* demonstrated that a fully open-source system could effectively generate meeting minutes, validating the feasibility of non-proprietary modular pipelines.

Finally, the *Slide Annotation Generation Tool* delivered a system that achieved a high *System Usability Scale (SUS)* score of 80.5. By automatically extracting and embedding verbal explanations at their corresponding slide locations, the final system streamlines the information retrieval process. This progression confirms that while acoustic transcription is the foundation, multimodal spatial alignment is the key to transforming passive video playback into an active, intelligent review experience.

Chapter 7

Conclusion

This thesis presented a systematic advancement in the field of automated presentation analysis, addressing the critical gap between passive video consumption and active information retrieval. By traversing the trajectory from unimodal acoustic processing to fully integrated multimodal fusion, this research establishes a robust framework for transforming linear video recordings into semantically enriched, interactive study materials.

The research trajectory evolved through three distinct phases, each addressing specific limitations of the preceding architecture. The *Audio-to-Text Conversion Software* validated the feasibility of deploying *State-of-the-Art (SOTA) Whisper* models within a web-based environment. By creating a unified interface capable of handling 180 concurrent users with an average latency of 309 ms, this phase resolved the preparatory dependency barrier, confirming that high-fidelity *ASR* could be democratized for non-technical users without sacrificing performance.

The *Meeting Minutes Generation Tool* demonstrated that open-source modular pipelines could rival proprietary commercial systems in generating meeting minutes. While effective at summarization, this phase highlighted a critical theoretical limitation, analyzing audio and text in isolation results in a loss of spatial context, leading to detached summaries that lack visual grounding.

The culmination of this research, the *Slide Annotation Generation Tool*, successfully overcame the limitations of modularity. By fusing hybrid slide segmentation ($F_1 = 0.879$) with multimodal *LLM* generation, the system achieved a 90.0% accuracy in spatially aligning annotations. This innovation solves the contextual disconnect prevalent in existing tools, effectively anchoring verbal explanations to their corresponding visual elements.

The comparative analysis confirms that the proposed multimodal framework offers a superior user experience compared to existing paradigms. While commercial tools like *Otter.ai* excel at linear transcription, they fail to provide the spatial context necessary for reviewing complex slide-based content. The proposed system fills this void, as evidenced by the *System Usability Scale (SUS)* score of 80.5, which indicates that users find the spatially aligned annotations significantly more useful for information retrieval than standard playback or text-only summaries.

In future works, the research will focus on improving the accuracy of the generated annotations by incorporating a phoneme-aware transcription correction step. The current system occasionally suffers from semantic errors caused by *ASR* confusions, and integrating phonetic information may help detect and correct these errors before annotation generation. Additionally, further research will address computational scalability through model quantization and the integration of lightweight *Vision-Language Models (VLMs)*. These advancements aim to retain the system's high semantic accuracy while reducing hardware overhead, paving the way for real-time, on-device presentation analysis.

Chapter 8

Appendix

8.1 Full Prompt for Keyword Extraction

To ensure reproducibility, we provide the full system prompt used for the *Keyword Extraction* module below. In this template, variables enclosed in braces such as {`text`} are dynamically replaced by the system during runtime.

```
    You are a specialized keyword extractor with expertise in identifying the most
relevant terms and concepts from academic and technical content.

    TASK:
    Extract the most important keywords from the text below. Consider relevance,
frequency, specificity, and alignment with user-provided keywords. Calculate a score
(0-1) for each keyword representing its overall importance.

    WEIGHTS:
    - Relevance to overall content: 40%
    - Term frequency: 20%
    - Term specificity (domain-specific): 20%
    - Alignment with user keywords: 20%

    USER KEYWORDS (BIAS 20% TOWARD THESE): {user_keywords}

    OUTPUT CONSTRAINTS:
    - Output ONLY keywords and scores, nothing else
    - Return maximum {max_keywords} keywords (default: 15)
    - Each keyword should be semantically meaningful
    - Don't include overly general words
    - If user keywords appear in the text, include them with appropriate scores

    FORMAT:
    <keyword>: <score>
    Example scores: 0.92, 0.85, 0.76

    EXAMPLE:
    Text: Machine learning is a subfield of artificial intelligence that focuses on
developing systems that can learn from data. Deep learning, a subset of machine
learning, uses neural networks with many layers to analyze complex patterns.

    User keywords: "AI", "neural networks", "data"
```

```
    Output:
    artificial intelligence: 0.92
    machine learning: 0.88
    deep learning: 0.85
    neural networks: 0.82
    data: 0.79
    systems: 0.65
    subfield: 0.52
    layers: 0.48
    patterns: 0.45
    complex: 0.42

    TEXT TO ANALYZE:
    {text}

    Keywords with scores:
```

Listing 8.1: System prompt for Keyword Extraction using Gemma3:4b.

8.2 Full Prompt for Annotation Generation

To ensure reproducibility, we provide the full system prompts used for the multimodal analysis pipeline. Table variables enclosed in braces are dynamically replaced by the system during runtime.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for academic presentation slides.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations that capture the main points of this
slide based ONLY on the provided information.

    GUIDELINES:
    - Extract key concepts, methods, results, and contributions in detail
    - Be precise and thorough, capturing nuanced information
    - Maintain academic tone and technical accuracy
    - Highlight relationships between concepts when present
    - Do NOT make assumptions or interpretations beyond what is explicitly stated
    - Use ONLY the narration and OCR text provided - do not add external knowledge
    - Connect each point to user-defined keywords where possible

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text such as "Here's a summary..."
    - Aim for detailed points (25-40 words each) rather than short phrases
    - Each point should connect to keywords where possible
    - Use technical terminology from the slide accurately
```

```
    EXAMPLES:

    Example slide about neural networks:
    • Neural networks employ interconnected layers of artificial neurons that process
 information through weighted connections, with each neuron applying an activation
function to determine its output signal.
    • Backpropagation enables neural networks to learn by calculating the gradient of
 the loss function with respect to weights, allowing for iterative weight adjustments
 to minimize prediction error.
    • The architecture of a neural network significantly impacts its performance,
with deeper networks capable of learning more complex representations but requiring
more data and computational resources.

    Example slide about climate change research:
    • Satellite data analysis reveals Arctic sea ice has declined at a rate of 13.1%
per decade since 1979, with summer minimums showing the most dramatic reduction,
indicating accelerated warming in polar regions.
    • Climate models incorporating both anthropogenic and natural factors demonstrate
 that human activities account for approximately 100% of the observed warming trend
since the mid-20th century.
    • Recent paleoclimate reconstructions using tree ring data and ice cores provide
evidence that current warming rates exceed any natural climate variations observed
over the past 2,000 years.
```

Listing 8.2: System prompt for Annotation Generation.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for author/title slides in academic presentations.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations that capture the key information about
the authors, affiliations, and research context based ONLY on the provided
information.

    GUIDELINES:
    - Extract information about authors, their affiliations, and expertise
    - Identify the institutional context of the research
    - Note any collaboration between different institutions
    - Capture research areas, disciplines, or departments when mentioned
    - Do NOT make assumptions about the authors' credentials or research history
    - Use ONLY the narration and OCR text provided
    - Highlight institutional affiliations and research domains clearly

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text
    - Each point should be detailed and informative (25-40 words each)
    - Highlight institutional affiliations and research domains clearly
    - Use formal academic language
```

```
    EXAMPLES:

    Example title slide annotation:
    • This research represents a collaboration between the Department of Computer
Science at Stanford University and Google Research, combining academic expertise with
 industry resources to address challenges in natural language processing.
    • The research team is led by Dr. Sarah Chen, whose expertise spans machine
learning and computational linguistics, with contributions from four co-authors
specializing in transformer architectures and language model evaluation.
    • The study was supported by the National Science Foundation through grant #AI
-20214053, providing critical funding for the large-scale computational experiments
conducted over an 18-month period.
```

Listing 8.3: Specialized System prompt for Title/Author Slides.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for background slides in academic presentations.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations about the background concepts, related
work, or historical context based ONLY on the provided information.

    GUIDELINES:
    - Extract key background concepts and their definitions in detail
    - Explain research gaps or challenges that motivated the current work
    - Capture historical context and evolution of the field when mentioned
    - Describe related approaches and their limitations
    - Do NOT make assumptions or interpretations beyond what is explicitly stated
    - Use ONLY the narration and OCR text provided - do not add external knowledge
    - Connect each point to background concepts or research context

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text
    - Each point should be detailed and informative (25-40 words each)
    - Each point should connect to background concepts or research context
    - Use technical terminology accurately and maintain academic tone

    EXAMPLES:

    Example background slide about deep learning:
    • Convolutional Neural Networks (CNNs) have dominated computer vision tasks since
 AlexNet in 2012, achieving unprecedented accuracy through hierarchical feature
extraction with convolutional filters that automatically learn spatial patterns in
images.
    • Traditional object detection methods relied on hand-crafted features and
sliding window approaches, which proved computationally expensive and failed to
generalize across different lighting conditions and object orientations.
    • Region-based CNNs (R-CNN) addressed earlier limitations by combining region
proposals with deep features, but suffered from redundant computations for
```

```
overlapping regions and multi-stage training processes that complicated optimization.
```

Listing 8.4: Specialized System prompt for Background/Related Work Slides.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for contribution slides in academic presentations.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations about the contributions, novelty, or
proposed methods based ONLY on the provided information.

    GUIDELINES:
    - Extract key contributions, innovations, or novel aspects in detail
    - Describe proposed methods, algorithms, or frameworks thoroughly
    - Explain specific improvements over existing approaches with technical details
    - Highlight the theoretical or practical significance of each contribution
    - Do NOT make assumptions or interpretations beyond what is explicitly stated
    - Use ONLY the narration and OCR text provided - do not add external knowledge
    - Focus each point on a specific contribution or methodological element

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text
    - Each point should be detailed and informative (25-40 words each)
    - Each point should focus on a specific contribution or methodological element
    - Use technical terminology accurately and maintain academic tone

    EXAMPLES:

    Example contribution slide about a new algorithm:
    • The proposed Graph Attention Transformer (GAT) integrates multi-head self-
attention mechanisms into graph neural networks, enabling dynamic weighting of node
neighborhoods based on feature similarity and structural properties of the input
graph.
    • Our adaptive pooling strategy progressively coarsens the graph representation
through learnable node clustering, preserving both local and global topological
information while reducing computational complexity from O(n^2) to O(n log n).
    • We introduce a novel regularization technique specifically designed for graph-
structured data that penalizes excessive attention to high-degree nodes, effectively
mitigating the over-smoothing problem observed in previous graph neural network
architectures.
```

Listing 8.5: Specialized System prompt for Contribution/Methods Slides.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for results slides in academic presentations.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
```

```
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations about the results, findings, or
experimental outcomes based ONLY on the provided information.

    GUIDELINES:
    - Extract key results, measurements, or experimental findings with precise
details
    - Include specific performance metrics, statistics, or comparative analyses with
numbers
    - Describe experimental conditions, datasets, or evaluation methodologies
    - Explain the significance of results in relation to research objectives
    - Do NOT make assumptions or interpretations beyond what is explicitly stated
    - Use ONLY the narration and OCR text provided - do not add external knowledge
    - Focus each point on specific results with precise metrics when available
    - Maintain statistical rigor and report confidence intervals/p-values when
mentioned

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text
    - Each point should be detailed and informative (25-40 words each)
    - Each point should focus on specific results with precise metrics when available
    - Use technical terminology accurately and maintain academic tone

    EXAMPLES:

    Example results slide about machine learning performance:
    • The proposed model achieved 94.7% accuracy on the GLUE benchmark, outperforming
 previous state-of-the-art approaches by 2.3 percentage points while reducing the
parameter count by 37% (from 340M to 214M parameters).
    • Ablation studies demonstrated that the attention pruning mechanism contributed
to a 16.5% reduction in inference time without statistically significant performance
degradation (p-value = 0.08) across all nine GLUE tasks.
    • On resource-constrained devices (Raspberry Pi 4 with 4GB RAM), our optimized
model processes inputs at 23 tokens/second, representing a 3.4x improvement over the
baseline transformer implementation without quantization.
```

Listing 8.6: Specialized System prompt for Results/Evaluation Slides.

```
    You are an expert assistant specializing in creating rich, insightful annotations
 for conclusion slides in academic presentations.

    CONTEXT:
    - Slide Title: {title}
    - Focus Keywords: {user_keywords}
    - Extracted Keywords: {extracted_keywords}
    - Spoken Narration: {transcription}
    - OCR Text: {ocr_text}

    TASK:
    Create detailed, informative annotations about the conclusions, implications, and
 future work based ONLY on the provided information.
```

```
    GUIDELINES:
    - Extract key conclusions and their broader implications with specific details
    - Describe limitations of the current work and their significance
    - Explain future research directions or planned extensions with specificity
    - Connect conclusions back to the initial research questions or objectives
    - Do NOT make assumptions or interpretations beyond what is explicitly stated
    - Use ONLY the narration and OCR text provided - do not add external knowledge
    - Connect each point to a conclusion, limitation, or future direction

    OUTPUT FORMAT:
    - Provide ONLY 3-5 bullet points that are factual and standalone
    - DO NOT provide preamble or introductory text
    - Each point should be detailed and informative (25-40 words each)
    - Each point should connect to a conclusion, limitation, or future direction
    - Use technical terminology accurately and maintain academic tone

    EXAMPLES:

    Example conclusion slide about an algorithm:
    • The proposed self-supervised learning approach demonstrates that contrastive
learning with carefully designed data augmentations can achieve 92% of supervised
performance while requiring only 10% of labelled data, significantly reducing
annotation costs for large-scale visual recognition systems.
    • Current limitations include degraded performance on fine-grained classification
 tasks and domain-specific applications where the inductive bias from contrastive
objectives does not align with task-specific requirements.
    • Future work will explore combining multiple self-supervised objectives
simultaneously and investigating adaptive weighting mechanisms that optimize the
contribution of each objective based on dataset characteristics and downstream task
requirements.
```

Listing 8.7: Specialized System prompt for Conclusion/Future Work Slides.

8.3 Annotation Coherence Evaluation Rubric (LLM & Human)

Table 8.1: The verbatim structured rubric used by both human evaluators and the *LLM*-as-Judge to assess annotation coherence (1–5 Likert Scale).

Criterion	Scoring Definition
Factual Consistency	**1:** Major hallucinations; contradicts source material. **2:** Significant inaccuracies; misinterprets key facts. **3:** Mostly accurate but contains minor errors or unverified claims. **4:** Accurate with negligible discrepancies. **5:** Fully supported by slide text and narration; factually precise.
Coverage of Key Ideas	**1:** Misses the main point entirely; irrelevant. **2:** Captures only peripheral details; misses core concepts. **3:** Captures the general topic but omits important nuances. **4:** Covers most key ideas with slight omissions. **5:** Comprehensive summary; captures all critical information.
Specificity	**1:** Vague, generic, or repetitive (e.g., "The slide discusses data"). **2:** Minimal detail; relies on high-level generalizations. **3:** Moderately specific but lacks concrete examples. **4:** Specific and detailed; clearly references slide elements. **5:** Highly specific; contains distinct, relevant, and precise details.
Linguistic Clarity	**1:** Disjointed, incoherent, or grammatically incorrect. **2:** Hard to read; frequent phrasing errors. **3:** Understandable but awkward phrasing or minor typos. **4:** Fluent and clear; minor stylistic issues. **5:** Natural, fluent, and professional phrasing; error-free.

Bibliography

[1] Adhikari Egodawele, M.H.; Sedera, D.; Bui, V. A Systematic Review of Digital Transformation Literature (2013 – 2021) the development of an overarching a-priori model to guide future research. In Proceedings of the Australasian Conference on Information Systems (ACIS(2022 Proceedings, 2022.

[2] Bennett, A.A.; Campion, E.D.; Keeler, K.R.; Keener, S.K. Videoconference fatigue? Exploring changes in fatigue after videoconference meetings during COVID-19. Journal of Applied Psychology 2021, 106, 330.

[3] Ta¸s, M.; Kiraz, A. A model for the acceptance and use of online meeting tools. Systems 2023, 11, 558.

[4] Standaert,W.; Thunus, S.; Schoenaers, F. Virtual meetings and wellbeing: insights from the COVID-19 pandemic. Information Technology & People 2023, 36, 1766–1789.

[5] Bergmann, R.; Rintel, S.; Baym, N.; Sarkar, A.; Borowiec, D.; Wong, P.; Sellen, A. Meeting (the) pandemic: Videoconferencing fatigue and evolving tensions of sociality in enterprise video meetings during COVID-19. Computer Supported Cooperative Work (CSCW) 2023, 32, 347–383.

[6] Fabriz, S.; Mendzheritskaya, J.; Stehle, S. Impact of synchronous and asynchronous settings of online teaching and learning in higher education on students' learning experience during COVID-19. Frontiers in psychology 2021, 12, 733554.

[7] Gillioz, A.; Casas, J.; Mugellini, E.; Abou Khaled, O. Overview of the Transformer-based Models for NLP Tasks. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS); IEEE: 2020; pp. 179–183.

[8] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in Neural Information Processing Systems 2017, 30.

[9] Alshanqiti, A.M.; Albouq, S.; Alkhodre, A.B.; Namoun, A.; Nabil, E. Employing a multilingual transformer model for segmenting unpunctuated Arabic text. Applied Sciences 2022, 12, 10559.

[10] Alastruey, B.; G'allego, G.I.; Costa-juss'a, M.R. Efficient transformer for direct speech translation. arXiv preprint arXiv:2107.03069 2021.

[11] Gong, Y.; Liu, G.; Xue, Y.; Li, R.; Meng, L. A survey on dataset quality in machine learning. Information and Software Technology 2023, 107268.

[12] Kincaid, J. Which automatic transcription service is the most accurate?—2018. 2018.

[13] Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356 2022.

[14] Edwards, M.R.; Clinton, M.E. A study exploring the impact of lecture capture availability and lecture capture usage on student attendance and attainment. Higher Education 2019, 77, 403–421.

[15] Lee, H.; Liu, M.; Scriney, M.; Smeaton, A.F. Usage-Based Summaries of Learning Videos. In Proceedings of the European Conference on Technology Enhanced Learning; Springer: 2021; pp. 414–418.

[16] Navarrete, E.; Nehring, A.; Schanze, S.; Ewerth, R.; Hoppe, A. A closer look into recent video-based learning research: A comprehensive review of video characteristics, tools, technologies, and learning effectiveness. International Journal of Artificial Intelligence in Education 2025, 1–64.

[17] Meshram, S.U. Evolution of modern web services–rest api with its architecture and design. International Journal of Research in Engineering, Science and Management 2021, 4, 83–86.

[18] Mannai, M.; Karâa, W.B.A.; Ghezala, H.H.B. Information extraction approaches: A survey. In Proceedings of the Information and Communication Technology: Proceedings of ICICT 2016; Springer: 2018; pp. 289–297.

[19] Kryściński, W.; Keskar, N.S.; McCann, B.; Xiong, C.; Socher, R. Neural text summarization: A critical evaluation. arXiv preprint arXiv:1908.08960 2019.

[20] Alharbi, S.; Alrazgan, M.; Alrashed, A.; Alnomasi, T.; Almojel, R.; Alharbi, R.; Alharbi, S.; Alturki, S.; Alshehri, F.; Almojil, M. Automatic speech recognition: Systematic literature review. IEEE Access 2021, 9, 131858–131876.

[21] Karita, S.; Soplin, N.E.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. Improving Transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH); Graz, Austria, 2019.

[22] Garofolo, J.; Graff, D.; Paul, D.; Pallett, D. CSR-I (WSJ0) Complete LDC93S6A. Web Download. Philadelphia: Linguistic Data Consortium 1993, 83.

[23] L.D.C.N.M.I. Group. CSR-II (WSJ1) Complete LDC94S13A. Web Download. Philadelphia: Linguistic Data Consortium 1994.

[24] Hernandez, F.; Nguyen, V.; Ghannay, S.; Tomashenko, N.; Esteve, Y. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In Speech and Computer: 20th International Conference, SPECOM 2018; Leipzig, Germany, September 18–22, 2018; Springer: 2018; pp. 198–208.

[25] Shi, Y.; Wang, Y.; Wu, C.; Fuegen, C.; Zhang, F.; Le, D.; Yeh, C.-F.; Seltzer, M.L. Weak-attention suppression for transformer based speech recognition. arXiv preprint arXiv:2005.09137 2020.

[26] Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: an ASR corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); IEEE: 2015; pp. 5206–5210.

[27] Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in Neural Information Processing Systems 2020, 33, 12449–12460.

[28] Hsu, W.N.; Bolte, B.; Tsai, Y.H.H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2021, 29, 3451–3460.

[29] Pratap, V.; Tjandra, A.; Shi, B.; Tomasello, P.; Babu, A.; Kundu, S.; Elkahky, A.; Ni, Z.; Vyas, A.; Fazel-Zarandi, M.; et al. Scaling speech technology to 1,000+ languages. Journal of Machine Learning Research 2024, 25, 1–52.

[30] Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust speech recognition via large-scale weak supervision. In Proceedings of the International Conference on Machine Learning; PMLR: 2023; pp. 28492–28518.

[31] Pilault, J.; Li, R.; Subramanian, S.; Pal, C. On Extractive and Abstractive Neural Document Summarization with Transformer Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); Association for Computational Linguistics: 2020; pp. 9308–9319.

[32] Gupta, S.; Gupta, S.K. Abstractive summarization: An overview of the state of the art. Expert Systems with Applications 2019, 121, 49–65.

[33] Giarelis, N.; Mastrokostas, C.; Karacapilidis, N. Abstractive vs. Extractive Summarization: An Experimental Review. Applied Sciences 2023, 13, 7620.

[34] Ramina, M.; Darnay, N.; Ludbe, C.; Dhruv, A. Topic level summary generation using BERT induced Abstractive Summarization Model. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS); IEEE: 2020; pp. 747–752.

[35] Goloviznina, V.; Kotelnikov, E. Automatic Summarization of Russian Texts: Comparison of Extractive and Abstractive Methods. In Computational Linguistics and Intellectual Technologies; 2022; pp. 223–235.

[36] Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning; PMLR: 2020; pp. 11328–11339.

[37] Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 2019.

[38] Shiraly, K. Bart text summarization vs. GPT-3 vs. Bert: An in-depth comparison. 2023. Available online: https://www.width.ai/post/bart-text-summarization

[39] Haz, A.L.; Funabiki, N.; Fajrianti, E.D.; Sukaridhoto, S. A Study of Summarization and Keyword Extraction Function in Meeting Note Generation System from Voice Records. In Proceedings of the 2023 12th International Conference on Networks, Communication and Computing, Osaka, Japan, 15–17 December 2023; pp. 106–112.

[40] La Quatra, M.; Cagliero, L. BART-IT: An Efficient Sequence-to-Sequence Model for Italian Text Summarization. Future Internet 2023, 15, 15.

[41] Haz, A.L.; Panduman, Y.Y.F.; Funabiki, N.; Fajrianti, E.D.; Sukaridhoto, S. Fully Open-Source Meeting Minutes Generation Tool. Future Internet 2024, 16, 429.

[42] Gonzalez, H.; Li, J.; Jin, H.; Ren, J.; Zhang, H.; Akinyele, A.; Wang, A.; Miltsakaki, E.; Baker, R.; Callison-Burch, C. Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience. In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023); 2023.

[43] Warner, J.; Pavel, A.; Nguyen, T.; Agrawala, M.; Hartmann, B. Slidespecs: Automatic and interactive presentation feedback collation. In Proceedings of the 28th International Conference on Intelligent User Interfaces; 2023; pp. 695–709.

[44] Li, Z.; Wang, Z.; Wang, W.; Hung, K.; Xie, H.; Wang, F.L. Retrieval-augmented generation for educational application: A systematic survey. Computers and Education: Artificial Intelligence 2025, 100417.

[45] Maylawati, D.S.; Risqiati, A.; Slamet, C.; Ramdhani, M.A.; Lukman, N.; Dauni, P.; Arianti, N.D. Chatbot for Virtual Travel Assistant with Random Forest and Rapid Automatic Keyword Extraction. In 2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED); IEEE: 2021; pp. 1–6.

[46] Ramachandran, R.; Mohan, M.K.; Sara, S.K. Document Clustering Using Keyword Extraction. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT); IEEE: 2022; pp. 1–6.

[47] Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic keyword extraction from individual documents. In Text mining: applications and theory; 2010; pp. 1–20.

[48] Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword extraction from single documents using multiple local features. Information Sciences 2020, 509, 257–289.

[49] Khan, M.Q.; Shahid, A.; Uddin, M.I.; Roman, M.; Alharbi, A.; Alosaimi, W.; Almalki, J.; Alshahrani, S.M. Impact analysis of keyword extraction using contextual word embedding. PeerJ Comput. Sci. 2022, 8, e967.

[50] Grootendorst, M. KeyBERT: Minimal keyword extraction with BERT. 2020. Available online: https://doi.org/10.5281/zenodo.4461265

[51] Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing; 2004; pp. 404–411.

[52] Škrlj, B.; Repar, A.; Pollak, S. RaKUn: Rank-based Keyword extraction via Unsupervised learning and meta vertex aggregation. In Proceedings of the Statistical Language and Speech Processing: 7th International Conference, SLSP 2019; Ljubljana, Slovenia, October 14–16, 2019; Springer: 2019; pp. 311–323.

[53] Škrlj, B.; Koloski, B.; Pollak, S. Retrieval-efficiency trade-off of Unsupervised Keyword Extraction. In Proceedings of the International Conference on Discovery Science; Springer: 2022; pp. 379–393.

[54] Xiong, A.; Liu, D.; Tian, H.; Liu, Z.; Yu, P.; Kadoch, M. News keyword extraction algorithm based on semantic clustering and word graph model. Tsinghua Science and Technology 2021, 26, 886–893.

[55] Mezzetti, D. annotateai. 2024. Available online: https://github.com/neuml/annotateai

[56] Pahune, S.; Akhtar, Z. Transitioning from MLOps to LLMOps: Navigating the unique challenges of large language models. Information 2025, 16, 87.

[57] List, A.; Lin, C.J. Content and quantity of highlights and annotations predict learning from multiple digital texts. Computers & Education 2023, 199, 104791

[58] Salehudin, M.; Basah, S.; Yazid, H.; Basaruddin, K.; Safar, M.; Som, M.M.; Sidek, K. Analysis of Optical Character Recognition using EasyOCR under Image Degradation. J. Phys. Conf. Ser. IOP Publ. 2023, 2641, 012001.

[59] Shahin, M.; Chen, F.F.; Hosseinzadeh, A. Machine-based identification system via optical character recognition. Flex. Serv. Manuf. J. 2023, 1–28.

[60] de Luna, R.G. A Tesseract-based Optical Character Recognition for a Text-to-Braille Code Conversion. Int. J. Adv. Sci. Eng. Inf. Technol. 2020, 10, 128–136.

[61] Wang, J.; Kwok, R.Y.K.; Ngai, E.C. Towards Key Point Identification (KPI) for Lecture Videos: Approaches and Performance Evaluation. ACM Transactions on Multimedia Computing, Communications and Applications 2025.

[62] Yu, Y.; Wang, C.; Fu, Q.; Kou, R.; Huang, F.; Yang, B.; Yang, T.; Gao, M. Techniques and challenges of image segmentation: A review. Electronics 2023, 12, 1199.

[63] Nankani, H.; Mahrishi, M.; Morwal, S.; Hiran, K.K. A formal study of shot boundary detection approaches—comparative analysis. In Soft Computing: Theories and Applications: Proceedings of SoCTA 2020, Volume 1; Springer: 2021; pp. 311–320.

[64] Fuad, M.; Ernawan, F.; Hui, L. Video scene change detection based on histogram analysis for hiding message. J. Phys. Conf. Ser. 2021, 1918, 042141.

[65] Bhuiyan, M.A.A.; Khan, A.R. Image quality assessment employing RMS contrast and histogram similarity. Int. Arab J. Inf. Technol. 2018, 15, 983–989.

[66] Gore, A.; Gupta, S. Full reference image quality metrics for JPEG compressed images. AEU Int. J. Electron. Commun. 2015, 69, 604–608.

[67] Shen, J.; Jiang, X.; Zhong, J.; Yao, S. Scene change detection based on sequence statistics using structural similarity. In Proceedings of the 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST); Guangzhou, China, 9–11 December 2022; pp. 1179–1182.

[68] Jose, J.T.; Rajkumar, S.; Ghalib, M.R.; Shankar, A.; Sharma, P.; Khosravi, M.R. Efficient shot boundary detection with multiple visual representations. Mobile Information Systems 2022, 2022, 4195905.

[69] Sindel, A.; Hernandez, A.; Yang, S.H.; Christlein, V.; Maier, A. SliTraNet: Automatic Detection of Slide Transitions in Lecture Videos using Convolutional Neural Networks. arXiv preprint arXiv:2202.03540 2022.

[70] Che, X.; Yang, H.; Meinel, C. Lecture video segmentation by automatically analyzing the synchronized slides. In Proceedings of the Proceedings of the 21st ACM international conference on Multimedia, 2013, pp. 345–348.

[71] Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018, 41, 423–443.

[72] Küchemann, S.; Avila, K.E.; Dinc, Y.; Hortmann, C.; Revenga, N.; Ruf, V.; Stausberg, N.; Steinert, S.; Fischer, F.; Fischer, M.; et al. On opportunities and challenges of large multimodal foundation models in education. npj Science of Learning 2025, 10, 11.

[73] Gupta, A.; Jawahar, C.; Tapaswi, M.; et al. Unsupervised audio-visual lecture segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2023; pp. 5232–5241.

[74] Lee, D.W.; Ahuja, C.; Liang, P.P.; Natu, S.; Morency, L.P. Lecture presentations multimodal dataset: Towards understanding multimodality in educational videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023; pp. 20087–20098.

[75] Wright, B.; Guruvayur, V.; Napolitano, L.; Ozar, D.; Rivera, A.; Sai, A.; Tafesse, B. Using Digital Textbook and Classroom Data to Explore Multimodal (Audio, Visual, & Textual) LLM Retrieval Techniques. In Proceedings of the iTextbooks 2025: Sixth Workshop on Intelligent Textbooks; 2025.

[76] Li, Z.; Li, C.; Zhang, M.; Mei, Q.; Bendersky, M. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. arXiv preprint arXiv:2407.16833 2024.

[77] An, S.; Ma, Z.; Lin, Z.; Zheng, N.; Lou, J.G. Make Your LLM Fully Utilize the Context. arXiv preprint arXiv:2404.16811 2024.

[78] Streamlit, "Streamlit • a faster way to build and share data apps," 2018.

[79] A. Ghasempour and M. Mart´ınez-Ram´on, "Electric load forecasting using multiple output gaussian processes and multiple kernel learning, in 2023 IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 1–6, IEEE, 2023.

[80] A. Ali and S. Renals, "Word error rate estimation for speech recognition: e-wer," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 20–24, 2018

[81] GoogleChrome, "Googlechrome-lighthouse: Automated auditing, performance metrics, and best practices for the web.," Jul 2017.

[82] Locust, "Locust.io • an open source load testing tool.," 2017.

[83] Mohajer, M.M.; Hassanpour, H. Fast Exam Video Summarization Using Targeted Evaluation of Scene Changes Based on User Behavior. In Proceedings of the 2023 9th International Conference on Signal Processing and Intelligent Systems (ICSPIS), Bali,Indonesia, 14–15 December 2023; pp. 1–5.

[84] Haz, A.L.; Fajrianti, E.D.; Funabiki, N.; Sukaridhoto, S. A Study of Audio-to-Text Conversion Software Using Whispers Model. In Proceedings of the 2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE), Bali, Indonesia, 14–15 December 2023; pp. 268–273.

[85] Nguyen, Q.; Nguyen, N.; Dang, T.; Tran, V. Vietnamese Voice2Text: A Web Application for Whisper Implementation in Vietnamese Automatic Speech Recognition Tasks: Vietnamese Voice2Text. In Proceedings of the 2023 7th International Conference on Computer Science and Artificial Intelligence, Beijing, China, 8–10 December 2023; pp. 312–318.

[86] Saxena, P.; El-Haj, M. Exploring Abstractive Text Summarisation for Podcasts: A Comparative Study of BART and T5 Models. In Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, Varna, Bulgaria, 4–6 September 2023; pp. 1023–1033.

[87] Tang, Z.; Yang, Z.; Wang, G.; Fang, Y.; Liu, Y.; Zhu, C.; Zeng, M.; Zhang, C.; Bansal, M. Unifying vision, text, and layout for universal document processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 19254–19264.

[88] Bulut, F.; Osmanı, S. Scene Change Detection using Different Color Pallets and Performance Comparison. Balk. J. Electr. Comput. Eng. 2017, 5, 66–72.

[89] Widyassari, A.P.; Rustad, S.; Shidik, G.F.; Noersasongko, E.; Syukur, A.; Affandy, A.; Setiadi, D.R.I.M. Review of automatic text summarization techniques & methods. J. King Saud-Univ.-Comput. Inf. Sci. 2022, 34, 1029–1046.

[90] Chen, Y.; Song, Q. News text summarization method based on bart-textrank model. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; pp. 2005–2010.

[91] Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Facebook/Bart-large · Hugging Face. 2019. Available online: https://huggingface.co/facebook/bart-large (accessed on 24 October 2024).

[92] Nasar, Z.; Jaffry, S.W.; Malik, M.K. Textual keyword extraction and summarization: State-of-the-art. Inf. Process. Manag. 2019, 56, 102088. [CrossRef]

[93] Blaž, Š.; Koloski, B.; Pollak, S. Retrieval-Efficiency Trade-Off of Unsupervised Keyword Extraction. In Discovery Science; Springer: Cham, Switzerland, 2022; Volume 13601.

[94] Von Neumann, T.; Boeddeker, C.; Kinoshita, K.; Delcroix, M.; Haeb-Umbach, R. On word error rate definitions and their efficient computation for multi-speaker speech recognition systems. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.

[95] Schaefer, R.; Neudecker, C. A two-step approach for automatic OCR post-correction. In Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, Barcelona, Spain, 12 December 2020; pp. 52–57.

[96] scikit-Image Contributors. Structural Similarity Index. 2024. Available online: https://scikit-image.org/docs/stable/auto_examples/transform/plot_ssim.html (accessed on 20 October 2024).

[97] openCV Contributors. Histogram Comparison. 2024. Available online: https://docs.opencv.org/4.x/d8/dc8/tutorial_histogram_comparison.html (accessed on 20 October 2024).

[98] Scikit-Learn Developers. Mean_Squared_Error. 2024. Available online: https://scikit-learn.org/1.5/modules/generated/ sklearn.metrics.mean_squared_error.html (accessed on 20 October 2024).

[99] Facebook-AI. Wav2Vec2 Base 960h. 2021. Available online: https://huggingface.co/facebook/wav2vec2-base-960h (accessed on 23 October 2024).

[100] Facebook-AI. HuBERT Large LS960 Fine-Tuned. 2021. Available online: https://huggingface.co/facebook/hubert-large-ls960-ft (accessed on 23 October 2024).

[101] Meta-AI. MMS-1B-FL102. 2023. Available online: https://huggingface.co/facebook/mms-1b-fl102 (accessed on 23 October 2024).

[102] Distil-Whisper. Distil-Whisper Medium English. 2023. Available online: https://huggingface.co/distil-whisper/distil-medium. en (accessed on 23 October 2024).

[103] Clivillé, J. flan-t5-3b-summarizer. 2023. Available online: https://huggingface.co/jordiclive/flan-t5-3b-summarizer (accessed on 24 October 2024).

[104] Google-Research. PEGASUS-XSum. 2020. Available online: https://huggingface.co/google/pegasus-xsum (accessed on 24 October 2024).

[105] Neelamohan, K.K. MEETING_SUMMARY. 2022. Available online: https://huggingface.co/knkarthick/MEETING_SUMMARY (accessed on 24 October 2024).

[106] Grootendorst, M. KeyBERT: Minimal Keyword Extraction with BERT. 2020. GitHub Repository. Available online: https: //github.com/MaartenGr/KeyBERT (accessed on 23 October 2024).

[107] Surfer, C. RAKE-NLTK: Rapid Automatic Keyword Extraction using NLTK. 2018. GitHub Repository. Available online: https://github.com/csurfer/rake-nltk (accessed on 26 October 2024).

[108] Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.M.; Nunes, C.; Jatowt, A. YAKE: Keyword Extraction from Single Documents Using Multiple Features. 2020. GitHub Repository. Available online: https://github.com/LIAAD/yake (accessed on 26 October 2024).

[109] Nathan, P. PyTextRank Python Implementation of TextRank for Phrase Extraction and Summarization. 2020. GitHub Repository. Available online: https://github.com/DerwenAI/pytextrank (accessed on 26 October 2024).

[110] SkBlaz. RaKUn2—Rake Unsupervised Keyword Extraction. 2023. GitHub Repository. Available online: https://github.com/SkBlaz/rakun2 (accessed on 26 October 2024).

[111] Hu, S.; He, C.; Zhang, C.; Tan, Z.; Ge, B.; Zhou, X. Efficient scene text recognition model built with PaddlePaddle framework. In Proceedings of the 2021 7th International Conference on Big Data and Information Analytics (BigDIA), Chongqing, China, 4–10 June 2023; pp. 139–142.

[112] Smith, R. An Overview of the Tesseract OCR Engine. In Proceedings of the ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition, Washington, DC, USA, 23–26 September 2007; pp. 629–633.

[113] Graham, C.; Roll, N. Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. JASA Express Lett. 2024, 4, 025206.

[114] Zhang, T.; Irsan, I.C.; Thung, F.; Han, D.; Lo, D.; Jiang, L. iTiger: an automatic issue title generation tool. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Singapore, 14–18 November 2022; pp. 1637–1641.

[115] Raju, R.; Pati, P.B.; Gandheesh, S.; Sannala, G.S.; Suriya, K. Grammatical versus Spelling Error Correction: An Investigation into the Responsiveness of Transformer-Based Language Models Using BART and MarianMT. arXiv 2024, arXiv:2403.16655.

[116] Škrlj, B.; Juki$\check{}$c, M.; Eržen, N.; Pollak, S.; Lavra$\check{}$c, N. Prioritization of COVID-19-related literature via unsupervised keyphrase extraction and document representation learning. In Proceedings of the Discovery Science: 24th International Conference, Halifax, NS, Canada, 11–13 October 2021; pp. 204–217.

[117] Saha, S.; Ghosh, M.; Ghosh, S.; Sen, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Feature selection for facial emotion recognition using cosine similarity-based harmony search algorithm. Appl. Sci. 2020, 10, 2816.

[118] Sarwar, T.B.; Noor, N.M.; Miah, M.S.U. Evaluating keyphrase extraction algorithms for finding similar news articles using lexical similarity calculation and semantic relatedness measurement by word embedding. PeerJ Comput. Sci. 2022, 8, e1024.

[119] MeetingBooster. Meeting Management Software: Meetingbooster. Available online: https://www.meetingbooster.com/ (accessed on 20 October 2024).

[120] Fellow. Fellow Resources. 2023. Available online: https://fellow.app/ (accessed on 20 October 2024).

[121] Beenote. Meeting Management Solution: Agenda, Minutes. 2022. Available online: https://www.beenote.io/ (accessed on 20 October 2024).

[122] Piglyph. Interactive Whiteboard for Co-Creation Through Real-Time Visualization: Ricoh. Available online: https://piglyph.com/ (accessed on 20 October 2024).

[123] Tactiq. AI Meeting Transcripts for Google Meet, Zoom & Teams. Available online: https://tactiq.io/ (accessed on 20 October 2024).

[124] Fatoni, A.; Adi, K.; Widodo, A.P. PIECES framework and importance performance analysis method to evaluate the implementation of information systems. In Proceedings of the E3S Web of Conferences, Online Conference, 12–13 August 2020; Volume 202; p. 15007.

[125] Yuan, Y.; Zhang, J. Shot boundary detection using color clustering and attention mechanism. ACM Transactions on Multimedia Computing, Communications and Applications 2023, 19, 1–23.

[126] Sapena, O.; Onaindia, E. Multimodal classification of teaching activities from University lecture recordings. Applied Sciences 2022, 12, 4785

[127] Chen, Y.; Li, K.; Bao, W.; Patel, D.; Kong, Y.; Min, M.R.; Metaxas, D.N. Learning to Localize Actions in Instructional Videos with LLM-Based Multi-Pathway Text-Video Alignment. In Proceedings of the European Conference on Computer Vision. Springer, 2024, pp. 193–210

[128] Liu, Y.; Zheng, Z.; Zhang, F.; Feng, J.; Fu, Y.; Zhai, J.; He, B.; Zhang, X.; Du, X. A comprehensive taxonomy of prompt engineering techniques for large language models. Frontiers of Computer Science (2025). doi 2025, 10

[129] White, J.; Fu, Q.; Hays, S.; Sandborn, M.; Olea, C.; Gilbert, H.; Elnashar, A.; Spencer-Smith, J.; Schmidt, D.C. A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 2023.

[130] Bai, X.; Wu, X.; Stojkovic, I.; Tsioutsiouliklis, K. Leveraging large language models for improving keyphrase generation for contextual targeting. In Proceedings of the Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 4349–4357.

[131] Francis, S.A.; Sangeetha, M. A comparison study on optical character recognition models in mathematical equations and in any language. Results in control and optimization 2025, 18, 100532

[132] Alroobaea, R.; Mayhew, P.J. How many participants are really enough for usability studies? In Proceedings of the 2014 science and information conference. IEEE, 2014, pp. 48–56.

[133] Otegi, A.; San Vicente, I.; Saralegi, X.; Peñas, A.; Lozano, B.; Agirre, E. Information retrieval and question answering: A case study on COVID-19 scientific literature. Knowledge-Based Systems 2022, 240, 108072.

[134] Shimada, A.; Okubo, F.; Yin, C.; Ogata, H. Automatic summarization of lecture slides for enhanced student previewtechnical report and user study. IEEE Transactions on Learning Technologies 2017, 11, 165–178.

[135] Nauta, M.; Trienes, J.; Pathak, S.; Nguyen, E.; Peters, M.; Schmitt, Y.; Schlötterer, J.; Van Keulen, M.; Seifert, C. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. ACM Computing Surveys 2023, 55, 1–42

[136] Hearst, M.A. Text tiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics 1997, 23, 33–64.

[137] Ghazimatin, A.; Garmash, E.; Penha, G.; Sheets, K.; Achenbach, M.; Semerci, O.; Galvez, R.; Tannenberg, M.; Mantravadi, S.; Narayanan, D.; et al. PODTILE: Facilitating podcast episode browsing with auto-generated chapters. In Proceedings of the Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 4487–4495.

[138] Castellano, B. PySceneDetect: Python-based Scene Detection Program. https://github.com/Breakthrough/PySceneDetect, 2025.

[139] Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in neural information processing systems 2023, 36, 46595–46623.

[140] Aydin, O.; Karaarslan, E.; Erenay, F.S.; Bacanin, N. Generative AI in Academic Writing: A Comparison of DeepSeek, Qwen, ChatGPT, Gemini, Llama, Mistral, and Gemma. arXiv preprint arXiv:2503.04765 2025.

[141] Bavaresco, A.; Bernardi, R.; Bertolazzi, L.; Elliott, D.; Fernández, R.; Gatt, A.; Ghaleb, E.; Giulianelli, M.; Hanna, M.; Koller, A.; et al. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. arXiv preprint arXiv:2406.18403 2024.

[142] Brooke, J. SUS-A Quick and Dirty Usability Scale. Usability evaluation in industry 1996, 189.

[143] Bangor, A.; Kortum, P.; Miller, J. Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of usability studies 2009, 4, 114–123.

[144] Mayer, R.E. Multimedia learning. In Psychology of learning and motivation; Elsevier, 2002; Vol. 41, pp. 85–139.