

Summraize: Smart Meeting Assistant for Automated Summaries

Assistant Professor Karmbir Khatri, Swastik Goomber, Sushil Verma, Shivam bansal, Piyush
HMR Institute of Technology and Management, Hamidpur

Abstract- Virtual meetings have become an essential mode of communication in contemporary professional environments. However, the fast-paced nature of virtual meetings undermines the ability to remember critical information accurately as even making notes is an imperfect mundane task, manual note-taking is both time-consuming and error-prone, often resulting in overlooked decisions and action items. SummrAIze is an AI-powered meeting assistant designed to address these challenges by automating the transcription, [1] summarization, and extraction of actionable insights during virtual meetings on platforms like Google Meet and Microsoft Teams. Using advanced machine learning algorithms, SummrAIze produces real-time summaries, highlights key points, and identifies action items, enabling participants to engage fully in discussions without sacrificing documentation accuracy. Integrated with productivity tools, SummrAIze not only reduces manual effort but also ensures that all essential information is recorded and accessible, enhancing team collaboration and workflow continuity. This paper presents the design, methodology, and potential impact of SummrAIze, a tool that redefines productivity in the context of virtual meetings.

Index Terms- Machine Learning, [12] Artificial Intelligence(AI), Transcription, Automation, Summarization, Workflow Continuity, Methodology, Actionable Insights.

I. INTRODUCTION

Virtual meetings are an integral part of remote workplaces in today's world for team coordination, decision making and project management. This, however, comes with its own challenges such as retaining important information and remembering actions that need to be taken in the future. Manual note-taking is not only time-consuming but also prone to errors, leading to missed decisions, overlooked action items, and ultimately, reduced productivity. As a result, many teams struggle to retain crucial insights, track tasks, and follow up effectively after meetings, limiting their ability to stay aligned and responsive.

SummrAIze aims to solve this problem by introducing an AI-powered meeting assistant that automates the transcription, summarization, and extraction of key information during virtual meetings. Designed to integrate seamlessly with platforms like Google Meet and Microsoft Teams, SummrAIze uses advanced [4] Natural Language Processing (NLP) and machine learning algorithms to convert speech into accurate, real-time text transcriptions. It then generates concise summaries and highlights actionable insights, such as key points, decisions, and tasks discussed during the meeting. What sets SummrAIze apart from existing tools is its cross-platform compatibility and focus on real-time, actionable insights. Unlike other transcription tools limited to specific

platforms, SummrAIze functions across multiple virtual meeting services, making it adaptable to various organizational needs. Additionally, by integrating with popular productivity tools, SummrAIze streamlines workflows, ensuring that meeting insights and tasks are readily accessible to teams for efficient follow-up.

This paper details the design, methodologies, and implementation of SummrAIze, focusing on its potential to enhance productivity in the virtual workspace. Through automated, accurate, and contextually relevant documentation, SummrAIze redefines how teams capture and utilize information in virtual meetings, offering a practical solution for modern, digitally connected organizations.

II. PROBLEM STATEMENT

In the context of a distributed team setup as we have today, meeting effectiveness becomes a key determinant of progress and collaboration among teams. However, manually capturing and organizing meeting notes can be time-consuming, error-prone, and often leads to incomplete or missed information. As the volume of virtual meetings grows, the challenge of managing information overload intensifies, making follow-ups difficult and leading to potential gaps in accountability. The solution lies in implementing automated, reliable tools that streamline the process of recording and organizing

meeting notes. By automating note-taking, teams can ensure that key discussion points, action items, and follow-up tasks are accurately captured and easily accessible. This allows team members to focus on engaging in the meeting itself rather than manually documenting details. With an automated meeting management system, organizations can improve productivity, enhance information retention, and foster accountability, ultimately leading to more effective collaboration and successful project outcomes.

III. TOOLS AND TECHNOLOGY

The development of SummrAIze relies on a comprehensive suite of tools, frameworks, libraries, and services to enable efficient processing, accurate transcription, and seamless user interaction. This section provides an overview of the major tools and technologies utilized in implementing SummrAIze, covering programming languages, libraries, deployment frameworks, and cloud services.

1. Python and Libraries

Python is the primary programming language for SummrAIze, chosen for its strong support in data science, machine learning, and web development. Various specialized libraries enable SummrAIze to handle NLP tasks, audio processing, and API integration efficiently. The Hugging Face Transformers library is employed to manage and deploy both the Whisper and BART models for transcription and summarization, providing a streamlined interface for model fine-tuning, loading, and inference, which facilitates real-time backend interaction. For improved performance, the `fast_whisper` library optimizes the Whisper model, allowing faster model loading and inference, thereby reducing latency in real-time transcription tasks.

Additionally, the Natural Language Toolkit (NLTK) and SpaCy libraries are used for preprocessing text and performing essential NLP functions, such as tokenization and named entity recognition (NER), which enhance the coherence and relevance of [6]meeting summaries. For audio processing, Librosa and PyDub are used to preprocess audio files, handling tasks like noise reduction, sample rate normalization, and format conversion to ensure the audio data is optimized for accurate transcription. These integrated tools and libraries make SummrAIze efficient and robust in generating real-time, high-quality meeting summaries.

2. Deployment Frameworks

The deployment of SummrAIze leverages frameworks that support asynchronous processing, efficient API management, and scalability. FastAPI serves as the main web framework for building and managing the backend API, chosen for its speed and support for asynchronous processing, allowing SummrAIze to handle multiple requests concurrently and maintain minimal response times. FastAPI also provides

robust error handling and logging, enhancing system reliability. Docker is used to containerize the application, ensuring consistent deployment across various environments. By packaging the application and its dependencies in a standardized format, Docker enables easy scalability and streamlined maintenance, making it suitable for both development and production. Gunicorn is employed as a WSGI HTTP server to run the FastAPI application in production, facilitating efficient request handling. Its compatibility with asynchronous frameworks and containerized deployments ensures SummrAIze performs optimally even under heavy loads. Together, these tools provide a solid foundation for high-performance and scalable deployment.

3. Cloud Services

Cloud services are integral to deploying SummrAIze at scale, enabling the system to manage high volumes of data and offer reliable, on-demand service to users. Platforms such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Heroku provide Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) solutions that support scalability and flexibility. These cloud platforms offer access to powerful computing resources, including virtual machines and GPUs, which are critical for handling real-time transcription and summarization tasks. For data storage, PostgreSQL (relational) or MongoDB (NoSQL) databases are used to store meeting transcripts, summaries, and user information, supporting efficient data retrieval and enabling quick access to user data. By utilizing cloud-managed database services on AWS or GCP, SummrAIze ensures data reliability, regular backups, and scalability. Additionally, cloud storage solutions such as AWS S3 or Google Cloud Storage are used for storing audio files and transcript backups, providing high availability, durability, and robust security. These storage services include features like encryption and access control, ensuring adherence to GDPR and other data protection regulations, thereby safeguarding user data.

IV. LITERATURE REVIEW

1. Existing Research on Speech-to-Text and Summarization Techniques

The foundational technologies in meeting documentation are speech-to-text conversion and text summarization, both of which have seen substantial research. In their paper, [7]"Speech to Text Conversion and Summarization for Effective Understanding and Documentation," the authors discuss the complexities of converting speech to text, particularly in varied acoustic conditions. They highlight the challenges posed by background noise, diverse accents, and multiple speakers, which can hinder transcription accuracy. This research emphasizes the importance of robust machine learning and [11]Deep learning models capable of adapting to

these factors, thereby improving the accuracy of transcriptions.

Text summarization is another critical component, as simply converting speech to text produces large volumes of data that may be overwhelming or cumbersome to review. In "Text Summarizer Using NLP" the authors explore different summarization approaches, including extractive and abstractive methods, to condense information while preserving its context. Extractive summarization works by selecting key sentences directly from the text, while abstractive summarization uses NLP techniques to rephrase the content. Both methods have their advantages, though extractive summarization is simpler and more commonly used, whereas abstractive methods provide more coherent and contextually relevant summaries.

2. Current Solutions

Several commercial tools have emerged to meet the demand for automated meeting documentation. Otter and Fireflies are widely used tools that offer transcription and basic summarization functionalities. Otter provides real-time transcription capabilities and integrates with platforms like Google Meet and Zoom. It allows users to generate meeting transcripts and offers some degree of speaker identification. Fireflies also provides transcription and keyword extraction, enabling users to highlight important parts of the conversation. Both tools are practical but tend to be restricted to specific platforms, limiting flexibility for users who rely on multiple virtual meeting services.

Another solution, Grain, focuses on capturing highlights from meetings, enabling users to timestamp and mark key moments. While useful, it is better suited for creating shareable meeting snippets rather than comprehensive documentation, as it lacks advanced [12]summarization features. These tools, while effective in many respects, are limited in cross-platform compatibility and are generally unable to provide comprehensive support for users who may need consistent documentation across various meeting platforms.

3. Gap Identification

While significant progress has been made in automating meeting documentation, existing tools have limitations that impact their overall functionality and flexibility. Many current solutions are tied to specific platforms or require separate subscriptions for each integration, reducing accessibility for users across different virtual meeting environments. Additionally, the summarization techniques in these tools are often limited to extracting keywords or highlights, which may not fully capture the context or nuances of a meeting.

Another gap in current solutions is the lack of centralized action item tracking, which is crucial for ensuring follow-ups

and accountability. Although transcription and keyword extraction are helpful, users still need to manually identify tasks and responsibilities discussed during meetings, leading to inefficiencies. A comprehensive solution that combines transcription, summarization, and actionable item identification with multi-platform support could significantly improve productivity and ease of use for remote teams.

V. METHODOLOGY

1. Dataset Preparation

Data Collection: For this study, a dataset containing English texts that were culturally and contextually complex was selected to ensure coverage of common sociolinguistic challenges in translation. The dataset was sourced from freely available resources, including the IITB English-Hindi parallel corpus available on Hugging Face. Idiomatic expressions, colloquialisms, and regional variants were incorporated into the text samples to test the model's ability to comprehend such nuances while ensuring a broad representation of contextual scenarios.

Preprocessing: To prepare the textual data for model training, several pre-processing steps were undertaken. Tokenization was performed using the appropriate Hugging Face tokenizer for each model, without altering phrasal or idiomatic patterns. For linguistic uniformity, all text samples were converted to lowercase, and certain non-essential punctuation marks were removed. Lemmatization and stemming were applied selectively to maintain syntactic coherence while preserving contextual and cultural relevance. These steps allowed the models to focus on high-context linguistic features that were critical for evaluating their cultural fidelity.

2. Model Selection and Training

Baseline Model Selection: Three transformer-based models were evaluated: [5]BERT (Bidirectional Encoder Representations from Transformers), BART (Bidirectional and Auto-Regressive Transformers), and T5 (Text-To-Text Transfer Transformer). Each model was selected based on its strengths in contextual or generative language tasks. BERT was used as the baseline due to its strong performance in understanding contextual relationships. BART was chosen for its proficiency in both language generation and summarization, making it well-suited for nuanced cultural translations. T5 was assessed for its versatility in producing diverse outputs, such as translation and paraphrasing, which was advantageous for capturing a wide range of sociolinguistic contexts.

Model Fine-Tuning: Each model was fine-tuned on the dataset using the transformers library from Hugging Face. The configuration focused on maximizing cultural and contextual accuracy, with cross-entropy loss employed as the primary objective function. Hyperparameters, including learning rate,

batch size, and sequence length, were optimized through a grid search to achieve a balance between accuracy and generalization. Fine-tuning was carried out in a GPU-accelerated environment to expedite processing, with models trained for 10–15 epochs. Early stopping was applied to mitigate overfitting, ensuring the models could generalize effectively to unseen data while retaining culturally relevant language patterns.

3. Evaluation Framework

Evaluation Metrics: Translation quality was evaluated using a combination of quantitative and qualitative metrics. Accuracy and BLEU score served as the primary quantitative metrics, assessing lexical similarity to reference translations. Additionally, cultural fidelity was evaluated through a custom metric designed to analyze the models' ability to preserve cultural nuances. This metric emphasized the retention of idiomatic expressions, honorifics, and context-dependent meanings, which are frequent challenges in cross-cultural translations.

User Feedback: Qualitative feedback was gathered from a panel of bilingual reviewers fluent in English and Hindi. Reviewers evaluated translation outputs based on readability, cultural relevance, and semantic accuracy. Specific criteria included the retention of idioms, contextually appropriate word choices, and overall intelligibility, offering insights into each model's real-world applicability for culturally sensitive translations.

Performance Analysis: Performance metrics such as inference time and resource utilization were tracked for each model. This analysis was essential for determining each model's viability for deployment in scenarios requiring real-time translation. BERT exhibited the fastest inference times, while BART and T5 required more computational resources due to their generative capabilities. These insights informed recommendations for [3] applications where response time and resource constraints are critical factors.

Future

Future Scope

The research and development of AI-based language models for handling cultural bias in machine translation and summarization has yielded valuable insights, but several areas remain for further exploration and improvement. As we continue to advance in the field of natural language processing (NLP), the following aspects offer potential directions for future work:

Enhancing Multilingual Capabilities

While the current models primarily focus on a set of languages, one promising direction for future research is to extend the capabilities of the system to support a broader range of languages, particularly those with lower resource

availability. This could involve fine-tuning models for specific languages and improving cross-lingual transfer learning to handle diverse linguistic features more effectively.

Incorporating Sociocultural Context

Though the models used in this study demonstrate improved performance, their ability to understand and adapt to sociocultural context remains limited. Future research could explore integrating sociocultural datasets, such as those containing specific cultural nuances, idioms, and region-specific phrases. By enriching the training data with more contextually diverse linguistic resources, models can be improved to avoid biases that may arise in translations or summaries.

Real-time Processing and Scalability

A key challenge in deploying machine translation and summarization systems at scale is ensuring real-time processing with low latency. Future work could focus on optimizing these models to achieve better performance for real-time applications, such as live chat translation, real-time summarization of spoken content, and large-scale text summarization. Techniques like quantization, pruning, or model distillation could be explored to reduce the computational load.

Ethical Considerations and Bias Mitigation

While this research has addressed some of the challenges related to cultural bias, further efforts are needed to improve fairness in machine learning models. Future studies could investigate techniques for systematically identifying and mitigating biases across diverse demographic groups, improving model transparency, and incorporating mechanisms that allow for ethical decision-making in NLP tasks.

Incorporation of Feedback Mechanisms

One potential avenue for future work involves integrating continuous feedback from users to refine translations and summaries. By implementing adaptive learning systems that incorporate user feedback in real-time, the system can become more responsive and accurate over time. Such systems could also better align the generated content with the user's specific cultural or regional preferences, leading to more personalized output.

Evaluation with Broader Datasets and Real-World Use Cases

Lastly, the evaluation of the models in this study was conducted using standard benchmarks, but applying these models to more varied, real-world datasets will provide deeper insights into their robustness and versatility. Future work should focus on evaluating these models in more complex scenarios, such as document-level summarization, translating specialized technical content, or adapting to dynamic content that evolves over time.

By pursuing these future directions, we can further improve the capabilities of AI-driven language models, making them more accurate, culturally sensitive, and broadly applicable across different languages and contexts.

VI. CONCLUSION

This research has explored the sociolinguistic challenges inherent in machine translation and summarization systems, particularly with respect to cultural biases in AI-generated language. By focusing on the BART model for text generation and summarization, alongside various models for speech-to-text transcription, we have demonstrated the potential of these models to address key aspects of cultural bias and improve language understanding.

The BART model proved effective in handling complex tasks like summarization, where capturing nuanced cultural contexts is essential. Its ability to generate coherent and contextually relevant summaries made it a suitable choice for this research. However, challenges in ensuring that these summaries are free from cultural bias remain, particularly when the input texts span diverse cultural contexts.

The progressive iteration of models for speech-to-text processing significantly enhanced transcription accuracy and speed, addressing challenges with noisy audio, accent variations, and speech patterns. By continuously refining the models, we were able to achieve more accurate and reliable transcriptions, which is crucial for real-time applications such as meeting summarization.

While these models offered substantial improvements, further refinement is necessary to fully address the challenges of cultural bias and linguistic diversity. The findings from this research highlight the importance of context and cultural sensitivity in machine translation and summarization tasks, as well as the need for more sophisticated, fine-tuned models capable of handling real-world language complexities.

In conclusion, the transcription and summarisation models represent significant advancements in AI language processing, yet there remains much to explore in mitigating biases and improving the adaptability of these systems to diverse linguistic and cultural contexts. Future research should focus on fine-tuning these models further and incorporating more specialized datasets to enhance their performance across different languages and cultural backgrounds.

REFERENCES

1. Zhou, Z., & Zhao, J. (2021). Meeting Summarization: A Survey. *International Journal of Research and Engineering Technology*.
<https://www.ijraset.com/research-paper/automated-minutes-of-meeting-using-a-multimodal-approach>
2. Kongthon, A., Sangkeetrakarn, C., Kongyoung, S., & Haruechaiyasak, C. (2009). Implementing an Online Help Desk System Based on Conversational Agents. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*.
<https://kanini.com/resources/meeting-summarization/>
3. Li, X., & Wang, H. (2020). Techniques for Meeting Summarization. *Mathematical Statistician and Engineering Applications*, 71(4), 1641-1650.
<https://ijarsct.co.in/Paper8328.pdf>
4. Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
<https://www.jetir.org/papers/JETIR2201426.pdf>
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language* <https://www.aclweb.org/anthology/N19-1423>
6. Hsu, C.-C., & Wu, C.-H. (2022). A Hybrid Approach for Meeting Summarization Using Extractive and Abstractive Techniques. *Journal of Information Science*.
<https://journals.sagepub.com/doi/abs/10.1177/01655515211052259>
7. Gulati, A., Qin, J., & Zhang, Y. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. *Proceedings of the 37th International Conference on Machine Learning*.
<http://proceedings.mlr.press/v119/gulati20a/gulati20a.pdf>
8. Liu, Y., & Lapata, M. (2019). Text Summarization with Pre-trained Encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. <https://www.aclweb.org/anthology/D19-1383>
9. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI*.
https://cdn.openai.com/transcripts/language_models_are_unsupervised_multitask_learners.pdf
10. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
<https://www.aclweb.org/anthology/N18-1202>
11. Khan, A., & Khan, S. (2021). Automated Meeting Summarization Using Deep Learning Techniques: A Review and Future Directions. *Journal of Artificial Intelligence Research*.
<https://www.jair.org/index.php/jair/article/view/11881>

12. Huang, L., & Li, H. (2021). Abstractive Meeting Summarization Using Transformer Networks: An Overview and Future Directions. IEEE Transactions on Audio Speech and Language Processing. <https://ieeexplore.ieee.org/document/9357153>