JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Automated Meeting Minutes Generator

Rajat Verma[#1], Sparsh Gupta[#1], Shubh Sharma[#1], Tanishq Aggarwal[#1],
Mahesha A.M[2]

[#1]. CSE DEPARTMENT. JSS ACADEMY OF TECHNICAL EDUCATION, NOIDA
[#2]. CSE DEPARTMENT. Assistant Professor JSSATE Noida

Abstract

Meetings are a very important aspect of everyday work life.
Because of the covid-19 epidemic, at least 60% of the meeting will take place online. Meetings with little physical interaction are notoriously difficult to succeed in [1]. Numerous research has been conducted to demonstrate how ineffective these internet meetings are. People continually ignore the point of meetings: each participant must take notes in order to remember what was discussed and what actions should be taken as a result of the meetings.

Many different types of meetings could benefit from technological assistance, such as the automatic creation of meeting minutes. To prepare reasonable automation, we must first have a thorough understanding of the various types of meetings, the linguistic properties and commonalities in the structure of meeting minutes, and the methods for automating them. We summarise the quality criteria and linguistic properties of meeting minutes in this paper, describe the available meeting corpora and meeting datasets, and propose a classification of meeting and minutes types. This also helps in saving a lot of time as it eases the process of taking notes of the meetings.

Furthermore, we examine the methods and tools for automatic minuting in terms of their suitability for use with existing dataset types. We summarise the knowledge gained in relation to our goal of designing automatic minuting and present our initial steps in this direction.

Keywords: *Meetings, Meeting notes, online meetings, meetings minutes.*

I. INTRODUCTION

The Covid-19 Pandemic impact can be seen in a variety of domains, including social, economic, tourist, and educational domains. Many schools have been closed in order to move all academic activities online [1, 2]. Users in remote areas hold important meetings via video conferencing without having to travel or meet in the same location. To improve the quality of learning, both teachers and students should use digital technology. To promote online learning, Google Classroom, Edmodo, Schoology, Zoom, and other media can be used.

Due to the pandemic, The meetings included work, schools, and other meetings. There are many platforms that came up during the pandemic to allow people to connect and have their conversations online without the need of having to meet in general. While the pandemic came up with a lot of bad and ugly things, it revolutionized the tech industry and people started finding solutions to connect online just like it was in offline meetings. But one drawback of online meetings was retention, it was really hard to retain what was discussed in the online meetings [3, 4].

We frequently miss the minutes of discussions in online meetings, so here's our solution: an automated meeting minutes generator. It is the simple online meeting assistant that provides you with automated meeting minutes in every conversation.

It will help you save time on board meetings, team management, and customer service so that you can focus on the conversation and never miss anything important.

We are reducing manual efforts and thus increasing team productivity. This will aid in removing the distraction of note-taking, reducing the number of ineffective meetings, and assisting the team in performing their best work.

This project also fits into the current online education scenario, where the majority of classes are held via online platforms such as Google Meet and Zoom. We assist students by providing quick and concise notes that help them revisit and memorize subjects more effectively.

II. RELATED WORK

Automatic Speech Recognition (ASR) systems have proliferated in recent years, to the point where free platforms like YouTube now offer speech recognition services. We contribute to the field of automatic speech recognition by comparing the relative performance of two sets of manual transcriptions and five sets of automatic transcriptions (Google Cloud, IBM Watson, Microsoft Azure, Trint, and YouTube) to assist researchers in selecting accurate transcription services. Furthermore, we identify nonverbal behaviors associated with unintelligible speech, as evidenced by high word error rates [5]. We demonstrate that manual transcriptions continue to outperform current automatic transcriptions. YouTube provides the most accurate transcription service among the automatic transcription services. In terms of nonverbal behavioral involvement, we show that when the speaker is clear, the variability of smile intensities from the listener is high (low) (unintelligible). These findings are based on interactions between participants in videoconferences [1].

Because it is impossible to interpret every point of the text in a document, text summarization is a demanding Natural Language Processing (NLP) challenge. This involves a deep examination of the text in various stages, such as semantic analysis, lexical relations, named entity recognition, and so on, all of which can be accomplished with only a significant amount of word knowledge. Because it is difficult to gather word information in numerous areas such as the meaning of a word in relation to other material, related terms, inferential interpretation, sentence creation, and so on, constructing abstracts as summaries have become challenging. This type of summary is known as abstractive summarization in NLP.

ABSTRACTIVE TEXT SUMMARIZATION

Text summarization is a tough Natural Language Processing (NLP) challenge owing to the difficulty in comprehending every point of the text in a document. This necessitates a detailed analysis of the text in several phases, such as semantic analysis, lexical relations, named entity identification, and so on, which can be achieved with just a considerable lot of word knowledge. Because obtaining word information in multiple areas such as the meaning of a word in relation to other material, related terms, inferential interpretation, sentence creation, and so on is difficult, creating abstracts as summaries have grown complex. In NLP, this sort of summarising is known as abstractive summarization.

However, an approximation, also known as extractive summarization, is more adaptable. The system must, in particular, identify the most relevant/significant parts of the text, extract them, organize them, and return them to the user. Although the extractive summarization job has been a prominent study issue since 1958 (Luhn, 1958), it remains significantly difficult to automatically summarise a text using a computer system, such as a human-created summary.

The technique of discovering a subset of a document that contains the information that resides in the complete document is known as automated text summarization (ATS). A text summarising system, according to Mani (1999), filters the important information from the original content to produce a shortened version.

III. PROPOSED WORK

The proposed project is basically to let the person concentrate on the meetings rather than thinking about the notes/minutes of the meetings. The people will like normally attend the meetings and without much of their interference, the meeting notes would be generated. The meeting notes will be emailed to the user and also visible to them on the dashboard website.

The users open the chrome extension when the meeting starts, they log in on the website as well as the chrome extension. Once that is done they attend the meeting, during the meetings our transcriber converts the speech into text which is then summarized into meetings notes using a text summarizer.

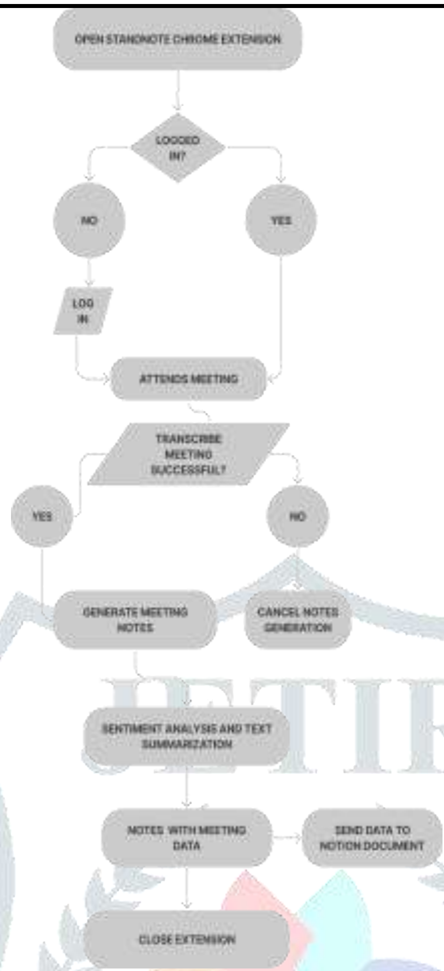

Fig 1: Workflow diagram of the project

IV. METHODOLOGY

The Automated meeting minutes generator will basically work like this: people will start the meeting, once all the people have joined the meeting one person from the team would open the chrome extension the speech to text transcriber will transcribe the meetings from speech to text on which the user can edit the converted text. Once the user is satisfied with the converted text the summarizer will run which will basically convert the complete meeting into notes.

Here are all the steps that are involved in the project:

1. Machine learning for speech to text conversion

Machine learning was utilized to transcribe the various talks into human-readable text, and the algorithm for speech-to-text conversion was tested over a large dataset. Efforts have been made to ensure that the transcribing is as efficient as feasible.

Machine learning (ML) is a subset of artificial intelligence (AI) that enables software applications to grow increasingly effective at predicting outcomes without explicitly programming them to do so. Machine learning algorithms estimate new output values by using historical data as input.

Azure speech-to-text algorithms have been used to convert the speech to text. Here is a detailed image showing how the azure pipeline works:

ASR API	Date	CV	F	IER	LS-c	LS-o
Human					5.8	12.7
Google	2018-03-30	23.2	24.2	16.6	12.1	28.8
Google Cloud	2018-03-30	23.3	26.3	18.3	12.3	27.3
IBM	2018-03-30	21.8	47.6	24.0	9.8	25.3
Microsoft	2018-03-30	29.1	28.1	23.1	18.8	35.9
Wit.ai	2018-03-30	35.6	54.2	37.4	19.2	41.7

Fig 2: Comparison of speech to text services [10]

2. Chrome Extension for user interface

Google Chrome extensions are apps that you can install in your Chrome browser to change the way it works. A chrome extension has been used to get the meeting's data and do all the analysis and processes on that data.

The automated meeting minutes generator obtains the user's audio by utilizing the tab capture API, which Chrome makes available to all of its extensions. Using this API, we obtain the user's audio and feed it into the speech-to-text algorithm, which converts it into human-readable text.

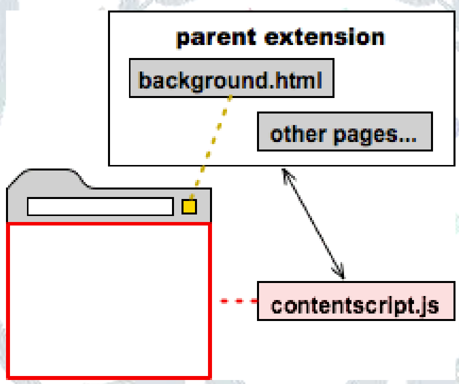

Fig 3: Chrome Extension basic working

3. Text Summarization of the Transcribed Text

The act of mechanically generating natural language summaries from an input document while keeping the important elements is known as text summarization. The main goal of this experiment is to use advanced NLP algorithms to generate grammatically correct and interesting summaries for pharmaceutical research publications. It can be utilized in financial research, newsletter summarisation, and market intelligence since it facilitates easy and quick retrieval of information.

The abstractive text summarization algorithm receives the converted text from the speech-to-text algorithm and returns the meeting notes [8]. Abstractive summarising is a technique for creating a summary of a text based on its key ideas rather than by copying the most striking lines verbatim from the text. The generated summaries may contain additional terms and sentences not found in the source material.

An abstractive text summarizer has been used to get the summary of the transcribed meeting. The summarizer receives the input from the frontend which sends the speech to text converted text. The text that comes to the summarizer is free of errors as the user has reviewed the text after it is transcribed. The Summarizer takes in the text and then depending upon the parameters creates the summary of the provided text to it.

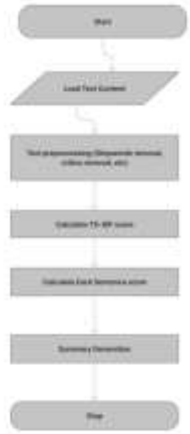

Fig 4: Text Summarizer flow

4. ReactJS Website

The notes that are generated on the backend are shown to the user on the ReactJS site, which is built using HTML, CSS JavaScript, and other technologies. The data comes from the Django REST APIs and using fetch request it is shown to the user.

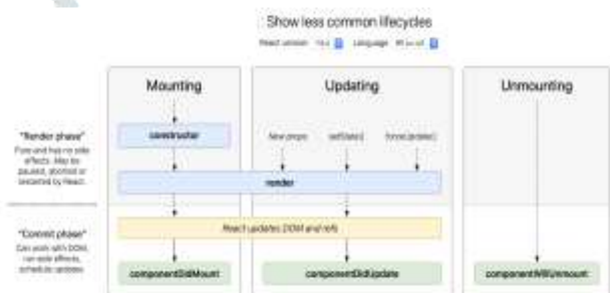

Fig 5: ReactJS state flowchart

IV. AUTHENTICATION AND SECURITY

Django REST framework has been used for user authentication. Django is a web development framework that allows you to create dynamic websites. A static website is one that only displays information and has no interaction (other than simple page requests) that is registered to a server. In a static website, the server just provides HTML, CSS, and JavaScript to the client. More capabilities necessitate a dynamic website, in which the server retains information and responds to user input in addition to simply providing the content [6]. One of the primary reasons for creating a dynamic site is to authenticate users and restrict material.

Writing, installing, and managing a static website is orders of magnitude easier, less expensive, and more secure than a dynamic site. As a result, you should only develop a dynamic website if the additional capabilities of the dynamic paradigm are required for your project. Django's built-in components simplify and speed the process of developing a dynamic site. The "user account" object, like the wheel, is tempting to reinvent as one of the fundamental components of a dynamic web application, but the conventional shape is adequate for most uses. Django comes with a sophisticated user model out of the box, and in this post, we'll go over the best approach to provide secure, intuitive user authentication routines.

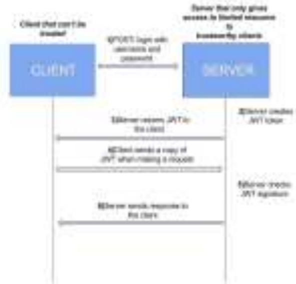

Fig 6: Django Authentication flow

For security, Django has clickjacking prevention in the form of the X-Frame-Options middleware, which can prevent a site from being rendered inside a frame in a browser that supports it [7]. It is possible to disable the protection on a per-view basis or to specify the exact header value that is transmitted.

Django has effective protections against a number of common threats, including XSS and CSRF attacks. In this paper, we've discussed how those particular threats are handled by Django on our website.

V. DESIGN OF PROPOSED SYSTEM

Different kinds of design in the proposed system are:

1. **UI Design:** For UI design, ReactJS has been used as a frontend UI library. Along with that, TailwindCSS has been used to create better.
2. **System Design:** STT has been used for speech-to-text conversion while abstractive text summarization [9, 10].
3. **Database Design:** SQL has been used for designing.

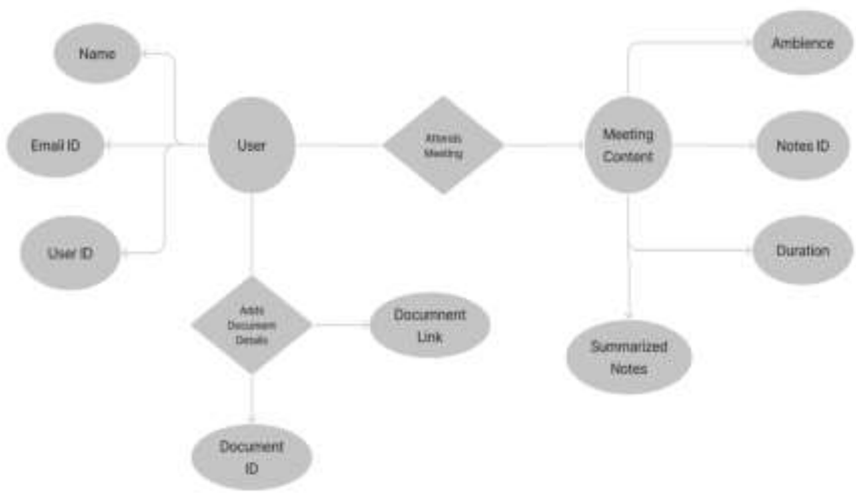

Fig 7: Relation Entity Diagram

VI. CONCLUSION

This paper suggests a method that will benefit society. It will help people to get the automated minutes of the meetings without having to do it manually. This will save a lot of manual effort. It will help you save time on board meetings, team management, and customer support such that you just focus on the conversation and never miss what's important.

Automated meeting minutes genera also fits into the current online education scenario where most of the classes take place over online platforms like Google Meet and Zoom. We help students with quick and concise notes that help them to revise and memorize the concepts better.

VII. RESULT

Here is the result of the meeting that we had, this is the transcribed speech into text:

Fig 7: Converted speech into text

The shown result is obtained by running on a meeting where the number of participants was 2, we were successfully able to transcribe the speech into text with an accuracy of around 60%.

VIII. REFERENCES

[1] A.M. Attorney. The Corporate Records Handbook: Meetings, Minutes and Resolutions. dElta printing solutions, inc., 2007.

[2] A.M. Attorney. Nonprofit meetings, minutes, and records. Delta printing solutions, inc., 2008.

[3] S. Banerjee, P. Mitra, and K. Sugiyama. Abstractive meeting summarization using dependency graph fusion. In Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, pages 5–6, New York, NY, USA, 2015. ACM.

[4] S. Banerjee, P. Mitra, and K. Sugiyama. Generating abstractive summaries from meeting transcripts. In Proceedings of the 2015 ACM Symposium on Document Engineering, DocEng '15, pages 51–60, New York, NY, USA, 2015. ACM.

[5] S. Banerjee, C. Rose, and A.I. Rudnicky. The necessity of a meeting recording and playback system, and the benefit of topic–level annotations to meeting browsing. In M.F. 11http://elitr.EU/ Costabile and F. Paternò, editors, Human-Computer Interaction - INTERACT 2005, pages 643–656, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[6] F. Bargiela and S.J. Harris. Managing Language: The discourse of corporate meetings. John Benjamins Publishing Company, 1997.

[7] F. Boudin and E. Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 298–305, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[8] T. Bui, M. Frampton, J. Dowding, and S. Peters. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In Proceedings of the SIGDIAL 2009 Conference, pages 235–243. Association for Computational Linguistics, 2009.

[9] Susanne Burger, Victoria MacLaren, and Hua Yu. The isl meeting corpus: The impact of meeting type on speech style. 01 2002.

[10] S.E. Butler. Mission Critical Meetings: 81 Practical Facilitation Techniques. Wheatmark, 2014.

[11] H. Chen, E. Benson, T. Naseem, and R. Barzilay. In domain relation discovery with meta-constraints via posterior regularization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 530–540, Portland, Oregon, USA, Jun 2011. Association for Computational Linguistics.

[12] J. Clarke. Global inference for sentence compression: An integer linear programming approach. J. Artif. Intell. Res., 31:399–429, 2008.

[13] J. Dowding, J.M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran. Gemini: A natural language system for spoken-language understanding. In HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993, 1993.

[14] L. Espy. Bad Meetings Happen to Good People: How to Run Meetings That Are Effective, Focused, and Produce Results. Blue Room Press, Memphis, Tenessee, 2017.

[15] R. Fernández, M. Frampton, J. Dowding, A. Adukuzhiyil, P. Ehlen, and S. Peters. Identifying relevant phrases to summarize decisions in spoken meetings. In Ninth Annual Conference of the International Speech Communication Association, 2008.

[16] R. Fernández, M. Frampton, P. Ehlen, M. Purver, and S. Peters. Modeling and detecting decisions in multi-party dialogue. In Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, SIGdial ’08, pages 156–163, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[17] K. Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In Proceedings of the 23rd International Conference on Computational Linguistics, COLING ’10, pages 322–330, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[18] M. Frampton, J. Huang, T. Bui, and S. Peters. Real-time decision detection in multi-party dialogue. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1133–1141. Association for Computational Linguistics, 2009.

[19] M. Galley and K. McKeown. Lexicalized Markov grammars for sentence compression. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pages 180–187, Rochester, New York, April 2007. Association for Computational Linguistics.

[20] Ch. Goo and Y. Chen. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. 2018 IEEE Spoken Language Technology Workshop (SLT), pages 735–742, 2018.

[21] I. Gurevych and M. Strube. Semantic similarity applied to spoken dialogue summarization. In Proceedings of the 20th International Conference on Computational Linguistics, COLING ’04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[22] K.M. Hoffman. Meeting design: For Managers, Makers, and Everyone. Louis Rosenfeld, 2018.