iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.53353

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Automatic Generation of Meeting Minutes Using NLP

Nikhil Khodake[1], Shweta Kondewar[2], Swarda Bhandare[3], Chinmay Haridas[4]
[1, 2, 3, 4]*Department of Computer Engineering, PVG's COET &GKPIM, Affiliated to Savitribai Phule Pune University, India*

Abstract: ***The COVID-19 pandemic exposed the globe to a different time period. Due to the widespread adoption of the work-from-home culture, business meetings are increasingly taking place online. Minutes Of Meeting(MOM) maintains a record of the topics discussed in online business meetings. Meeting minutes is a written document used to inform participants and non-participants of what happened during the meeting. The main issue with meeting minutes is that the person should personally attend the meeting and it takes a lot of time to create MOM manually. We proposed a system to get fluent, concise, and adequate minutes of meeting from a given transcript. We observe better results than our baseline models and achieve eval_rouge1 score of 47.39 and an eval_loss of 1.56 for summarization tasks .***

Keywords: Natural Language Processing, Minutes Of Meeting, Summarization, Transcript, Transfer Learning

I. INTRODUCTION

Due to covid-19 pandemic many people have adopted work from home culture, so business meetings are happening in online mode. In those business meetings mom keeps a record of what was discussed at the meeting. The main problem with meeting minutes is that the person should attend the meeting personally and after that it takes much time to write them down properly. In order to generate accurate minutes of meeting we used various steps such as redundancy elimination, linear segmentation, summarization and various post-processing techniques.

II. DATASET

We referred to the samsum dataset for training of the model. The Samsung Research and Development Institute in Poland created the Samsum dataset, which is made available for research. About 16k messenger-like interactions with summaries can be found in the Samsum dataset. Linguists with English fluency created and recorded conversations. Linguists were instructed to generate dialogues that reflected the proportion of themes of their real-life messenger discussions and were similar to those they write on a daily basis. There are many different conversational styles and registers, including informal, semi-formal, and formal ones. Slang phrases, emojis and typos may also be used. Then, the conversations were annotated with summaries. It was assumed that summaries should be a concise brief of what people talked about in the conversation in third person.

Table 1: Statistics of the Dialogue and Meeting Dataset - SAMSUM

SET	DIALOGUE
TRAINING	14,732
TESTING	819
VALIDATION	818
TOTAL	16,349

III. METHODOLOGY



Fig. 1 System Implementation Steps

A. *Redundancy Elimination*

Stop words: stop words are words that are frequently used but that search engines have been programmed to ignore, both when indexing entries for searching and when retrieving them as the outcome of a search query. We don't want these words to take up any unnecessary storage room or processing time in our database. To do this, we can simply delete them by keeping a collection of words you believe to be stop words. A list of stop words is stored in nltk (natural language toolkit) in python and is available in 16 distinct languages. The nltk data repository is where you can locate them.

B. *Linear Segmentation*

Long recordings and meeting transcripts that are text documents are typically divided into topically coherent text segments that each contains a certain number of text sections. One would anticipate that the word usage would exhibit more consistent lexical distributions within each topically coherent section than it does across segments. For text analysis tasks like passage retrieval, document summarization, and discourse analysis, natural language processing (NLP), more specifically, a linear partition of texts into subject segments can be used. In this exercise, we'll go over how to create a python script to pre-process a collection of transcripts and turn them into numerical representations that subject segmentation algorithms can use.

C. *Summarization*

For summarization we proceeded with an abstractive approach. The process of creating a brief and succinct summary that encapsulates the key ideas of the original text is known as abstractive text summarization. The generated summaries might include novel words and phrases that aren't in the original text. These techniques can produce fresh sentences, which sharpens a summary's emphasis, cuts down on repetition, and maintains a high compression level.

In this summarization architecture, initially we started experimenting with various pertained model such as BART. We fine-tuned every model using samsum dataset and analysed performance of each. For fine-tuning the underlying summarization module, we use the following configurations: ‘max input length’ = 512, ‘min target length’ = 128. As mentioned, the training data for this method is the samsum corpus. We use a learning rate of 2e −5 and a batch size equal to 4 during the training. However, the performance of distilbert-base-uncased model was better which a pre trained transformer model is. The model is mainly intended to be fine-tuned on tasks, like sequence classification, token classification, or question answering, that use the entire sentence to make decisions.

D. *Post-Processing*

This process is followed after text summarization to get more accurate and short length summary which is an important task followed for generating minutes of meeting.in which we perform various tasks like keyword extraction, removing redundancy from summarised data, adding pronouns instead of using nouns, eliminating grammatical inconsistencies to generate the minutes of meeting. for post-processing we have also used various algorithms such as Textrank, tf_idf score to generate more accurate summaries.

1) *Textrank:* it is a graph based ranking algorithm used for text ranking. Textrank works the same as Google’s PageRank algorithm works. The Textrank algorithm firstly converts the text into a graph. Various attributes of text are used as vertices of a graph, such attributes may include words, sentences etc. After that it ranks the text and collects the text with the highest score to generate the summary.
2) *TF_IDF:I*t is obtained by multiplying term frequency and inverse document frequency. Term frequency is the raw count of instances a word appears in a document and inverse document frequency can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the algorithm.

TF(T, D) = TOTAL APPEARANCE OF THE TERM ′ T ′IN A DOCUMENT / TOTAL TERMS IN THE DOCUMENT ′ D′
IDF(T) = LOG TOTAL NUMBER OF DOCUMENTS IN A DOCUMENT SET / DOCUMENT FREQUENCY OF THE TERM ′ T ′ TF−IDF(T, D) = TF ∗ IDF

IV. EXPERIMENTAL SETUP

A. *Model Architecture-Distilbert-Base-Uncased*

Distilbert, a transformers model that was pre trained on the same corpus in a self-supervised manner using the bert basic model as a teacher, is smaller and faster than BERT.

This paradigm is uncased, meaning that it does not differentiate between the two forms of English. The model is mainly intended to be fine-tuned on tasks, like sequence classification, token classification, or question answering, that use the entire sentence (possibly masked) to make decisions.

Three objectives on which it was pertained -

1) The model was trained to return the same probabilities as the bert base model, which results in distillation loss.

2) Masked language modelling (mlm) is a portion of the bert basic model's initial training loss. When taking a sentence, the model randomly masks 15% of the words in the input then runs the entire masked sentence through the model and has to predict the masked words. This contrasts with conventional recurrent neural networks (rnns), which typically perceive the words sequentially, and autoregressive models like gpt, which inwardly conceal the upcoming tokens. This makes it possible for the model to acquire a two-way representation of the sentence.

3) Cosine embedding loss-the model was also trained to produce hidden states that were as similar to the bert base model as feasible.

B. Data Preparation

Standard text cleaning steps like removing numbers, special characters, punctuation, accidental spaces, etc. From a given transcript. Stop words were removed using the nltk library. Also, stemming, lemmatization and part of speech tagging (POS). Stemming is defined as reducing words to their root form while lemmatization is used to convert words to their base or dictionary form.. All these techniques help in reducing dimensionality of the data and improving the accuracy of our model.

C. Training Setup

We used the pretrained 'distilbert-base-uncased' model from the hugging face transformers 1 library. All other modules used in our methodology were built using pytorch. The model was trained using an automodelforseq2seqlm with a learning rate of 2e-5.Additionally, early stopping was used if the validation loss does not decrease after 10 successive epochs. The train_batch_size and Eval_batch_size was set to 8 and 16 respectively, weight_decay of 0.01

A sagemaker provided by aws was used to fine-tuned the pretrained model, the model pipeline was created using serverless framework for future enhancement of accuracy of the model. Data processing, training, evaluation-loss, validation, and register/create model steps comprise the pipeline.the successful completion of each stage is required in order to register and save a new model. New models should be more accurate than older versions of models when being stored.

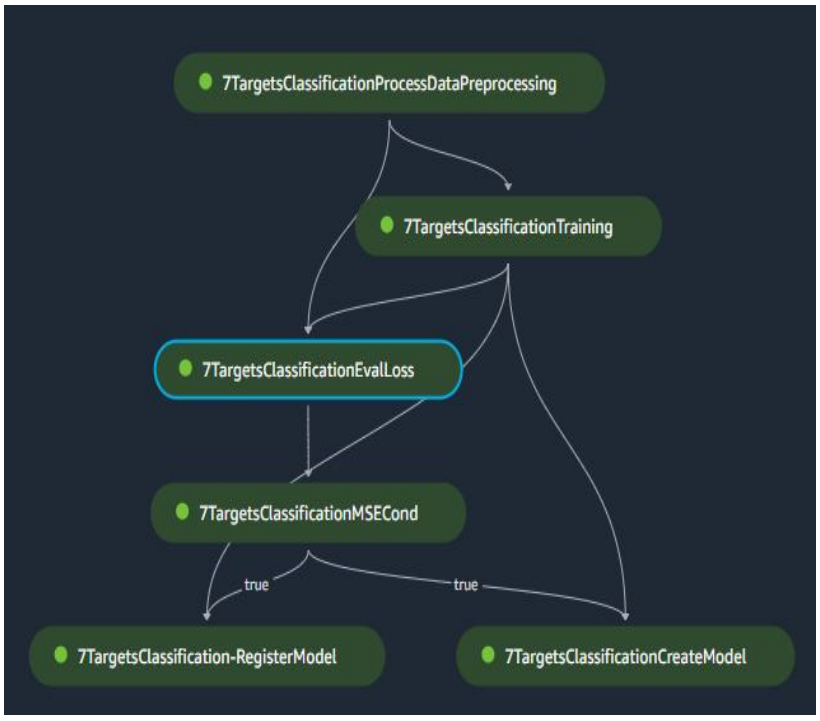


Fig. 2 Summarization Model Building Pipeline

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

V. RESULTS AND DISCUSSION

A. *Evaluation Metrics*

We evaluated the performance of the distilbert-base-uncased model using standard evaluation metrics commonly employed in text summarization tasks. The metrics used include rouge (recall-oriented understudy for gisting evaluation) scores, which measure the overlap between the generated summaries and the reference summaries in terms of n-gram recall, precision, and f1 score.

B. *Comparison with Baseline Models*

We compared the performance of the distilbert-base-uncased model with several baseline models, including extractive methods and other abstractive models. The results clearly demonstrated the superiority of the distilbert-base-uncased model in terms of overall summarization quality and coherence.

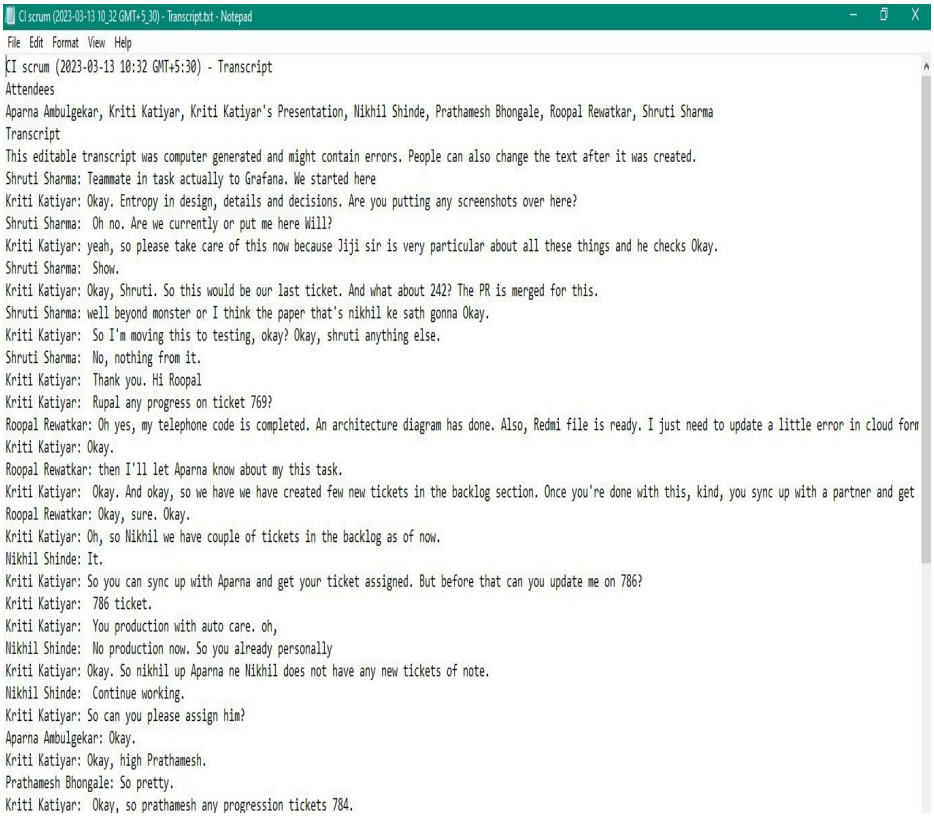
C. *Quantitative Results*

Table 2 presents the quantitative results of our experiments. The distilbert-base-uncased model achieved a rouge-1 score of 47.39, rouge-2 score of 24.09, and rouge-l score of 39.85. These scores indicate that the model successfully captured important information from the source text and generated summaries that had substantial overlap with the reference summaries. The high rouge-1 score suggests that the model effectively preserved the key content of the original text in the meeting minutes.

Table 2: Results for MOM Prediction on Validation Data

MODEL	EVAL_ROUGE1	EVAL_ROUGE2	EVAL_ROUGEL	EVAL_LOSS
Distilbert-Base-Uncased	47.3987	24.0999	39.852	1.5650

D. *Output Results*


CI scrum (2023-03-13 10_32 GMT+5_30) - Transcript.txt - Notepad

File Edit Format View Help

CI scrum (2023-03-13 10:32 GMT+5:30) - Transcript

Attendees

Aparna Ambulgekar, Kriti Katiyar, Kriti Katiyar's Presentation, Nikhil Shinde, Prathamesh Bhongale, Roopal Rewatkar, Shruti Sharma

Transcript

This editable transcript was computer generated and might contain errors. People can also change the text after it was created.

Shruti Sharma: Teammate in task actually to Grafana. We started here

Kriti Katiyar: Okay. Entropy in design, details and decisions. Are you putting any screenshots over here?

Shruti Sharma: Oh no. Are we currently or put me here Will?

Kriti Katiyar: yeah, so please take care of this now because Jiji sir is very particular about all these things and he checks Okay.

Shruti Sharma: Show.

Kriti Katiyar: Okay, Shruti. So this would be our last ticket. And what about 242? The PR is merged for this.

Shruti Sharma: well beyond monster or I think the paper that's nikhil ke sath gonna Okay.

Kriti Katiyar: So I'm moving this to testing, okay? Okay, shruti anything else.

Shruti Sharma: No, nothing from it.

Kriti Katiyar: Thank you. Hi Roopal

Kriti Katiyar: Rupal any progress on ticket 769?

Roopal Rewatkar: Oh yes, my telephone code is completed. An architecture diagram has done. Also, Redmi file is ready. I just need to update a little error in cloud form

Kriti Katiyar: Okay.

Roopal Rewatkar: then I'll let Aparna know about my this task.

Kriti Katiyar: Okay. And okay, so we have we have created few new tickets in the backlog section. Once you're done with this, kind, you sync up with a partner and get

Roopal Rewatkar: Okay, sure. Okay.

Kriti Katiyar: Oh, so Nikhil we have couple of tickets in the backlog as of now.

Nikhil Shinde: It.

Kriti Katiyar: So you can sync up with Aparna and get your ticket assigned. But before that can you update me on 786?

Kriti Katiyar: 786 ticket.

Kriti Katiyar: You production with auto care. oh,

Nikhil Shinde: No production now. So you already personally

Kriti Katiyar: Okay. So nikhil up Aparna ne Nikhil does not have any new tickets of note.

Nikhil Shinde: Continue working.

Kriti Katiyar: So can you please assign him?

Aparna Ambulgekar: Okay.

Kriti Katiyar: Okay, high Prathamesh.

Prathamesh Bhongale: So pretty.

Kriti Katiyar: Okay, so prathamesh any progression tickets 784.

Fig 3. Input Transcript File

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

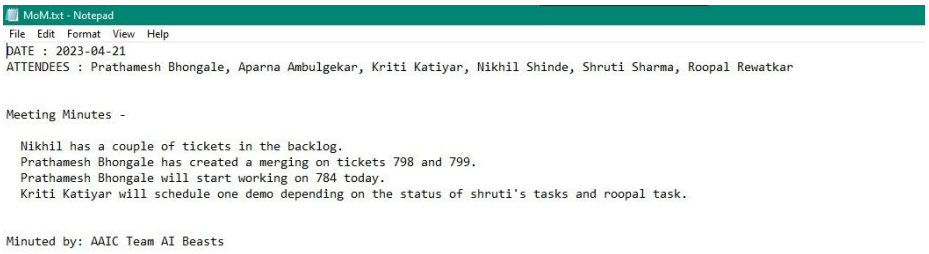


Fig 4.Meeting of Minutes Output Template

E. Quantitative Analysis

To further assess the quality of the generated meeting minutes, we conducted a qualitative analysis. We randomly selected a subset of summaries generated by the distilbert-base-uncased model and compared them with the reference summaries. Human evaluators were asked to rate the summaries based on criteria such as relevance, in formativeness, and fluency. The results of the qualitative analysis indicated that the distilbert-base-uncased model consistently produced summaries that were highly relevant to the source text, contained important information, and exhibited fluent and coherent language. The evaluators found the summaries to be concise and well-structured, effectively capturing the essence of the source document. The high grammatical correctness scores signify the merit of our selection of the distilbert-base-uncased model fine-tuned on the samsum corpus. The generated minutes are consistent in terms of fluency and adequacy.

VI. FUTURE WORK

1) *Handling Multimodal Inputs*-Meetings often involve multiple modalities, such as audio recordings, video streams, and presentation slides. Future research can focus on developing models that can effectively process and incorporate these multimodal inputs into meeting minutes generation. This could involve techniques such as audio-to-text conversion, visual information extraction, and fusion of multimodal representations to create more comprehensive and informative summaries.
2) *User Feedback and Iterative Improvements*-Collecting user feedback and incorporating it into the model's training and fine-tuning process can significantly enhance meeting minutes generation. Future work can explore methods to actively involve users in the feedback loop, enabling iterative improvements based on their preferences and specific needs.
3) *Real-Time Meeting Minutes Generation*-Real-time meeting minutes generation is an emerging area that holds great potential. Developing models and systems that can generate meeting minutes in real-time, allowing participants to access summaries during the meeting itself, could greatly improve meeting efficiency and productivity.

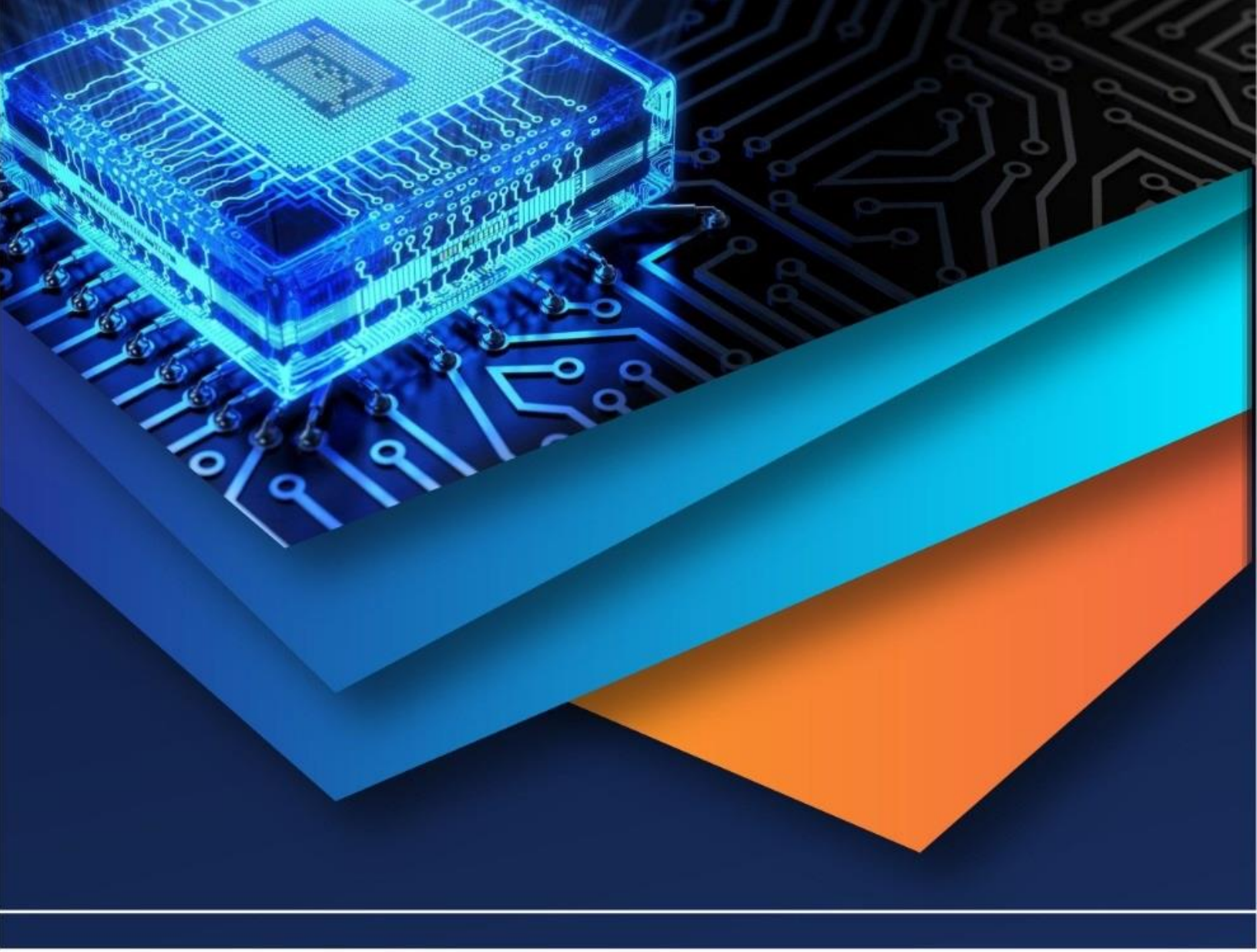
VII. CONCLUSIONS

In online business meetings, traditional method involves human intervention to generate minutes of meeting in order to solve this problem we build and model which help us to generate minutes of meetings in less time without any human involvement. The Minutes Of Meeting is a written document which is used to keep record of what was discussed in the meeting and inform participants and non-participants of what decisions have been taken during the meeting. This model generates minutes of meeting only in English language from given transcript. We observed better results than our baseline models and achieve eval_rouge1 score of 47.39 and an eval_loss of 1.56 for summarization task.

REFERENCES

[1] Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Ken-ichi Yokote and Kenji Nagamatsu.Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization.
[2] Published by Harihara Vinayakaram Natarajan, Kiran P V N N, Kunal Kasodekar, Ayushri Arora and Jaanam Haleem. Automatic Generation of Meeting Minutes from Online Meeting Transcripts
[3] Kartik Shinde1, Nidhir Bhavsar2, Aakash Bhatnagar2,Tirthankar Ghosal3. Team ABC @ AutoMin 2021: Generating Readable Minutes with a BART-based Automatic Minuting Approach.
[4] https://huggingface.co/docs/transformers/model
[5] Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Fei Liu.A Sliding-Window Approach to Automatic Creation of Meeting Minutes

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)