Team JU_PAD @ AutoMin 2021: MoM Generation from Multiparty Meeting Transcript

Palash Nandi, Sarthak Pan, Dipankar Das

Dept. Computer Science and Engineering, Jadavpur University, Kolkata, India

sondhanil1@gmail.com, sarthakpan@gmail.com, dipankar.das@jadavpuruniversity.in

Abstract

Use of online meeting platforms for long multi-party discussion is gradually increasing and generation of Minutes of Meeting (MoM) is crucial for subsequent events. MOM records all key issues, possible solutions, decisions and actions taken during the meeting. Hence the importance of minuting cannot be overemphasized in a time when a significant number of meetings take place in the virtual space. Automatic generation of MoM can potentially save up to 80% of time while revisiting. In this paper, we present an abstractive approach for automatic generation of meeting minutes. It aims to deal with problems like the nature of spoken text, length of transcripts and lack of document structure and conversation fillers.

The system is evaluated on a test dataset. The evaluation score is calculated by both manual and automatic systems. Text summarization metrics ROUGE-1, ROUGE-2, ROUGE-L [1] are used for automated scoring and metrics Adequacy, Grammatical Correctness, Fluency are used for manual scoring. The proposed model achieved 0.221, 0.046, 0,125 for ROUGE-1, ROUGE-2 , ROUGE-L respectively in automated evaluation and 3.5/5, 3/5, 3/5 for Adequacy, Grammatical Correctness, Fluency respectively in manual evaluation.

Index Terms: minute of meeting, POS tagging, abstractive summarization, extractive summarization

1. Introduction

Manual revision of the events of a previously held meeting is a hectic, time consuming and labored job. But often it becomes crucial to step forward. The scenario gets even more complex with increasing meeting duration and occurrence. Therefore summarizing the important events of a meeting can be helpful in the long run.

Automated summarization of texts is arguably one of the most challenging task in Natural Language Processing (NLP). It requires the automated system to identify important information and represent them concisely with minimum loss of information and redundancy. There are two kinds of text summarization tasks [2]. First, extractive summarization aims to create a summary by selecting a subset of the sentences in the input text that maximizes the coverage of important content while minimizing redundancy.

In contrast, the second one, abstractive summarization aims to create an abstract representation of the input text and use natural language generation techniques to generate a summary. Early approaches towards abstractive summarization includes sentence compression [3] which generates a grammatical summary of a given sentence, sentence fusion [4] which uses bottom-up local multisequence alignment to identify phrases conveying similar information and statistical generation to combine common phrases into a sentence, sentence revision [5], which generates sentences not found in the input and synthesizes information across sentences. In recent time, apart from neural oriented approaches, most of the abstractive summarization technique either contains three subtasks performed in a pipeline fashion: information extraction [6] [7] [8], content selection [9] [6] and surface realization or graph based representation [2] [10] [11] [7].

There are differences between minuting a meeting and a summarization of a text. To the best of our knowledge the field of minute generation of a meeting is still unexplored. AutoMin [12] is an ‘Interspeech 2021 satellite event’ endorsed by the ‘International Speech Communication Association’ (ISCA). There are three tasks (A, B, C) available for the year 2021. This paper focuses only on the shared task A (for English language). Task A is about automatic creation of minutes from multiparty meeting transcripts. Automatic minuting is challenging in the sense that there is no universal framework for creating minutes as well as highly dependent upon meeting type, subjects, and objectives.

2. Shared Task A

2.1. Task Definition

Shared task A is to generate minutes for provided multi-party transcripts.

2.1.1. Data

Provided data for shared task A consists of text files only. For training and validation each training transcript file is assisted with multiple MoM, annotated by different annotators. For testing only transcript files are provided. Table 1 depicts information about the dataset provided for Task A.

Table 1: *Statistics of dataset for Task A.*

Dataset	Number of transcript files
Train	85
Validation	10
Test	18

2.1.2. Data Analysis

On thoroughly analyzing transcripts following are the find outs:

- Speaker name is enclosed within “[]” or “()”.

- There are some expressions which are enclosed in “<>” like, <laugh>, <parallel_talk>¿, etc.
- Speakers have used conversational fillers like, “Umm, huh, mmm, so” and many more.
- Sometimes speakers have stopped in the mid-way of speaking a word like “[Person1] atta- attach the documents”.
- “Ok”, “No” and “Yes” are said in various forms like “Okay, Yeah, yup, nah”.
- There is a repetition of punctuation marks like “. . . ”.
- There is a repetition of words sometimes like “always there there is a way”.
- There is use of non-ASCII characters.
- Use of contractions like “didn’t, I’ll” etc.

2.1.3. MOM analysis

The MOM provided for testing varies annotator to annotator. There is no particular pattern followed for the preparation of minutes. The expected length of MoM to be 15-25% of the original transcript.

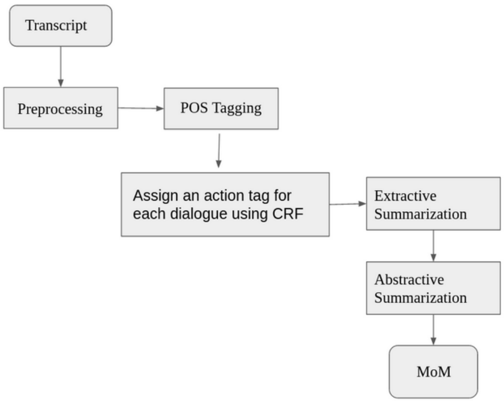
3. Our Approach

Since the length of the transcript is large (i.e., more than 5000 words), they cannot be directly used to generate a summary. Hence, we have to reduce the length to get an abstract summary out of the transcript. We have decided to go with multiple stacked models together to get the final outcome.
Steps we followed are,

- Pre-processing
- POS tagging
- Sentence/Dialogue Processing
- Extractive Summarization
- Abstractive Summarization
- Minute generation

Figure 1 depicts the sequential stages of MoM generation.

Figure 1: *Diagrammatic representation of workflow*


Transcript
Preprocessing
POS Tagging
Assign an action tag for each dialogue using CRF
Extractive Summarization
Abstractive Summarization
MoM

3.1. Pre-processing

The pre-processing focuses on resolving the issues discussed in section 2.1.2. In the preliminary step, the speaker is identified and the dialogue is transformed from the sentence into a columnar format where the first column is the name of the speaker and second column is for the dialogue. We have also prepared a list of contractions and substituted them with their full form. A full stop is added to the end of sentence in absence of a punctuation mark.

3.2. POS tagging

It attaches each word with it’s part of speech. Since we are dealing with conversational data it plays an important role. A false POS tag can greatly affect the processes. “pos-fast” pretrained model of Flair[1] is used for our POS tagging purpose. The python Flair package has simple implementation which deals with sequence tagging and prediction of pos. While saving pos we have added ‘/’ in beginning of pos, after the original word to identify our pos tags. A csv temporary file is generated with speakers in 1st column and pos tagged sentence in the 2nd column.

Table 2: *Details of POS tags.*

Tag	Meaning
CC	Coordinating conjunction
DT	Determiner
EX	Existential there
FW	Foreign word
IN	subordinating conjunction
JJ	Adjective
NN	Noun, singular or mass
NNP	Proper noun, singular
PRP	Personal pronoun
RB	Adverb
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
WDT	Wh-determiner
WP	Wh-pronoun

3.3. Sentence or Dialogue Processing

A CRF [4][13] model trained on “The Switchboard Dialog Act Corpus” data which consist of 1115 conversations, containing 205,000 utterances and 43 different discourse tags is used for sentence/dialogue processing. CRF model for our sequence labeling considers start of conversation, change of speakers, all the words, all POS tags of the words as different features.

For example, [’1’, ’0’, ’TOKEN_Hi’, ’Hi’, ’TOKEN_[PERSON11]’, ’[PERSON11]’, ’TOKEN_.’, ’.’, ’POS_UH’, ’POS_NNP’, ’POS_.’]. The CRF model will predict the dialogue act for each sentence. Table 3 describes all valid dialogue acts for the CRF model.

Apart from the tags mentioned in the above table all sentences with other tags are ignored because they will not contribute anything for minute preparation.
Once the correct sentences are selected, we apply some rules

Table 3: *CRF accepted acts.*

Tag	Meaning
nn	No answer
na	Non-negative answers
qy	Yes-No Question
qw	Wh-Question
qh	Rhetorical-Questions
aa	Agree or accept
ba	Appreciation

to resolve discourse relations. We have identified questions and their respective answers. The "yes/no" questions and their agreement/disagreement were also coupled together. Appreciation of some discussed topics by speakers was also identified. "I, me, my, you, your" are also replaced with proper speaker names. While replacing grammar is also taken care like "I am eating" should be converted to "Person1 is eating". At the end we will get a cleaned and simplified version of the transcript for further processing.

3.4. Extractive Summarization

For extractive summarization the TextRank [5] algorithm is used. TextRank algorithm considers each embedded sentence as a node and assigns a weight for each edge based on similarity of two sentences. Then sentences are scored by their centrality in the graph. The text generated from the previous step is taken as input for the TextRank algorithm. The python package summa is used for this project. The extractive transcript summary length ratio is set to 0.5.

3.5. Abstractive Summarization

For the purpose of abstract summarization, the BART [7] model is used. We have used the 'facebook/bart-large-cnn' pre trained model from Hugging Face. This model has been shown to have state of the art performance on other summarization tasks. Along with that we have also tested output of a BERT based model and with human evaluation, results don't seem to be intimidating.
We have followed the summarization technique discussed in [14] using BART. Later, K-Means clustering is used to choose sentences closest to the centroid. We have restricted our function to generate 45 sentences for the summary.

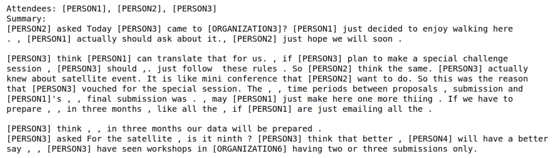
3.6. Minute generation

The output of the abstract summarization is formatted according to the required MoM format. The format is:

Attendees: List of Persons
Summary: {Point wise sentences of summary}

Since the summary generated in the form of a paragraph and splitting the sentence based on punctuation marks doesn't make it very large, we have decided to split the sentence if the sentence starts with a person's name only. The final output is stored in a text file as MoM for a particular transcript file. A small snippet of generated MoM of a test data transcript is given below.

Figure 2: *A snippet of a system generated MoM on test dataset.*



```
Attendees: [PERSON1], [PERSON2], [PERSON3]
Summary:
[PERSON2] asked Today [PERSON3] came to [ORGANIZATION3]? [PERSON1] just decided to enjoy walking here
. , [PERSON1] actually should ask about it., [PERSON2] just hope we will soon .

[PERSON3] think [PERSON1] can translate that for us. , if [PERSON3] plan to make a special challenge
session , [PERSON3] should ,. just follow  these rules . So [PERSON2] think the same. [PERSON3] actually
knew about satellite event. It is like mini conference that [PERSON2] want to do. So this was the reason
that [PERSON3] vouched for the special session. The , , time periods between proposals , submission and
[PERSON1]'s , , final submission was . , may [PERSON1] just make here one more thiing . If we have to
prepare , , in three months , like all the , if [PERSON1] are just emailing all the .

[PERSON3] think , , in three months our data will be prepared .
[PERSON3] asked For the satellite , is it ninth ? [PERSON3] think that better , [PERSON4] will have a better
say , , [PERSON3] have seen workshops in [ORGANIZATION6] having two or three submissions only.
```

4. Results and Evaluation

The system is evaluated on a test dataset. The evaluation score is calculated by both manual and automatic systems. Text summarization metrics ROUGE-1, ROUGE-2, ROUGE-L [1] are used for automated scoring and metrics Adequacy, Grammatical Correctness, Fluency are used for manual scoring. Table 4 and Table 5 represents the scores of the proposed system on the test data.

Table 4: *Automated evaluation of the proposed system on test data.*

Automatic Evaluation	Score
ROUGE-1	0.221
ROUGE-2	0.0469
ROUGE-L	0.126

Table 5: *Manual evaluation of the proposed system on test data.*

Manual Evaluation	Score
Fluency	3/5
Adequacy	3.5/5
Grammatical Correctness	3/5

5. Conclusions and Future Work

Although both extractive and abstractive summarization methods are incorporated in order to create a concise MoM, the proposed model is not able to produce satisfactory results. Therefore, the scope of improvement still remains huge.
Since we have used pre-trained models. If we can annotate dialogue act tags for the CRF model and train CRF model with new data according to our needs then more precise and accurate sentences can be generated.

6. Acknowledgements

This research would not be possible without guidance from Prof. Dipankar Das. We acknowledge all the co-authors for their contributing efforts and are thankful to the organizers Automin2021 for providing the opportunity.

7. References

[1] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries." 2004.

[2] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art." *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI Press.*, p. 9815–9822, January 27- February 1, 2019.

[3] T. A. Cohn and M. Lapata, "Sentence compression as tree transduction." *Journal of Artificial Intelligence Research.*, p. 637–674, 2009.

[4] R. Barzilay and K. R. McKeown, "Sentence fusion for multidocument news summarization." *Computational Linguistics.*, 2005.

[5] A. Tanaka, H.; Kinoshita, "Syntax-driven sentence revision for broadcast news summarization." *ACL-IJCNLP Workshop on Language Generation and Summarisation.*, 2009.

[6] P. Bing, L; Li, "Abstractive multi-document summarization via phrase selection and merging." *ACL-IJCNLP*, 2015.

[7] G. Mehdad, Y.; Carenini, "Abstractive summarization of spoken and written conversations based on phrasal queries." *ACL*, 2014.

[8] L. Wang and C. Cardie, "Domain-independent abstract generation for focused meeting summarization." *ACL*, 2013.

[9] P. Genest and G. Lapalme, "Fully abstractive approach to guided summarization." *ACL*, 2012.

[10] C. Greenbacker, "Towards a framework for abstractive summarization of multimodal documents." *ACL Student Session*, 2011.

[11] L. Li, W.; He and H. Zhuge, "Abstractive news summarization based on event semantic link network." *COLING*, 2016.

[12] T. Ghosal, O. Bojar, M. Singh, and A. Nedoluzhko, "Overview of the first shared task on automatic minuting (automin) at interspeech 2021," in *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, 2021, pp. 1–25. [Online]. Available: http://dx.doi.org/10.21437/AutoMin.2021-1

[13] K. Filippova and M. Strube, "Sentence fusion via dependency graph compression." *EMNLP*, 2008.

[14] D. Miller, "Leveraging bert for extractive text summarization on lectures. arxiv preprintarxiv: 1906.04165," 2019.