Automatic Minuting: A Pipeline Method for Generating Minutes from Multi-Party Meeting Transcripts

Kartik Shinde[†], Tirthankar Ghosal[‡], Muskaan Singh[*], and Ondřej Bojar[‡]
[†]Indian Institute of Technology Patna, Bihta, Bihar, India
[‡]Charles University, Faculty of Mathematics and Physics, ÚFAL, Czech Republic
[*]Speech and Audio Processing Group, IDIAP Research Institute, Martigny, Switzerland
kartik_1901ce16@iitp.ac.in, msingh@idiap.ch
(ghosal,bojar)@ufal.mff.cuni.cz

Abstract

Automatically generating meeting minutes is a challenging yet time-relevant problem in speech and natural language processing. Nowadays, meeting minutes seem more crucial than ever due to the manifold rise of online meetings. However, *automatic minuting* is not straightforward for various reasons: obtaining transcriptions of sufficient quality, summarizing long dialogue discourse, retaining topical relevance and coverage, handling redundancies and small talk, etc. This paper presents our investigations on a pipelined approach to automatically generate meeting minutes using a BART model (Bidirectional and Auto-Regressive Transformers) trained on multi-party dialogue summarization datasets. We achieve comparable results with our simple yet intuitive method with respect to previous large and computationally heavy state-of-the-art models. We make our code available at https://github.com/ELITR/minuting-pipeline.

1 Introduction

Ever since most of our interactions went virtual, the need for automatic support to run online meetings became essential. Due to frequent meetings and the resulting context switching, people are experiencing an information overload (Fauville et al., 2021) of epic proportions. Hence a tool to automatically summarize a meeting transcript would be a valuable addition to the virtual workplace. *Automatic Minuting* is the task of generating bullet-point meeting minutes from multi-party meeting transcripts. The AutoMin shared task at Interspeech 2021 (Ghosal et al., 2021) is a community-wide effort in this direction. Organizers of AutoMin (Ghosal et al., 2022a) released a medium-scale annotated corpus (Nedoluzhko et al., 2022) of transcript-minute pairs for conducting the shared task.

Automatic Minuting is close to summarization but not the same; subtle differences exist. Summarization aims at generating a concise and coherent text summary. It often purposely removes some less critical information; minuting is more inclined towards adequately capturing the entire contents of the meeting (*coverage is probably more significant than coherence and conciseness*).

Summarizing spoken multi-party dialogues comes with challenges: incorrect or noisy automated speech recognition (ASR) outputs, long discourse, topical shifts, the dialogue turns, redundancies and small talk, etc. Hence we deem automatic minuting to be more difficult than text summarization.

Due to the variety of sub-problems associated with this task, we adopt a pipelined approach. Our method encompasses (i) pre-processing the ASR-generated meeting transcripts to drop redundancies and noise, followed by (ii) unsupervised topical segmentation, and finally (iii) summarizing each segment of the discourse with a BART model (Raffel et al., 2019) pre-trained on a large-scale dialogue summarization dataset. Our initial investigation yields encouraging results. The obtained minutes resemble the human gold standard in terms of readability and coverage. Our main contribution lies in developing a lightweight, easy-to-implement, and efficient automatic minuting pipeline by leveraging pre-trained Transformer-based language models fine-tuned on large-scale dialogue summarization datasets.

2 Related Work

Although meeting summarization is a well-studied problem in the summarization literature (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Celikyilmaz et al., 2018; Chen and Bansal, 2018; Zhong et al., 2019; Xu and Durrett, 2019; Liu and Lapata, 2019; Lebanoff et al., 2019; Cho et al., 2019; Wang et al., 2020; Xu et al., 2019; Jia et al., 2020), automatic minuting is defined as a task relatively recently (Ghosal et al., 2021). We survey some of the relevant meeting summarization research in this section.

Early studies like Chen and Metze (2012) used intra-speaker topic modeling to summarize meetings. Later, several approaches (Zhao et al., 2019; Liu and Chen, 2019; Liu et al., 2019) documented the efficacy of hierarchical methods in learning the inherent structure of conversations. Li et al. (2019) utilized a multi-modal hierarchical attention mechanism across the topic, utterance, and word levels for the task. However, their method depends on manual annotation of topical segments and visual attention of the participants in the meetings, which are not commonly available. Zhu et al. (2020) introduced a hierarchical network *HMNet* for end-to-end training with cross-domain flexibility, which is now one of the state-of-the-art models for meeting summarization but is very resource-intensive. Recently, Liu and Chen (2021) proposed a dynamic sliding window strategy for abstractive summarization, achieving a close to state-of-the-art performance. Along similar lines, Zhong et al. (2021) presented a pre-training approach for long dialogue understanding and summarization with window-based denoising. Zhang et al. (2021) introduced a flexible multi-stage framework for longer input texts, combining a multi-stage greedy transcript segmentation with end-to-end training. Singh et al. (2021) tested several baseline text summarization models for automatic minuting and concluded that *off-the-shelf* summarization models are not suited for the concerned task.

Most of the above deep neural models are resource-heavy. The hierarchical model, HMNet, requires 4 Tesla V-100 GPUs with 32G memory on each. Our proposed pipeline approach is straightforward and consists of separate stages for each sub-task in the pipeline: pre-processing, redundancy elimination, transcript segmentation, summarization, and post-processing. Each stage has a unique problem, with specified target outputs, culminating in the final objective, i.e., minutes generation. We would also like to point out that the earlier methods do not aim for automatic meeting minutes generation; instead, they strive to generate coherent meeting summaries in the form of paragraphs. Our motivation is to generate meeting minutes in the form of bullet points that adequately capture the contents of the meeting.

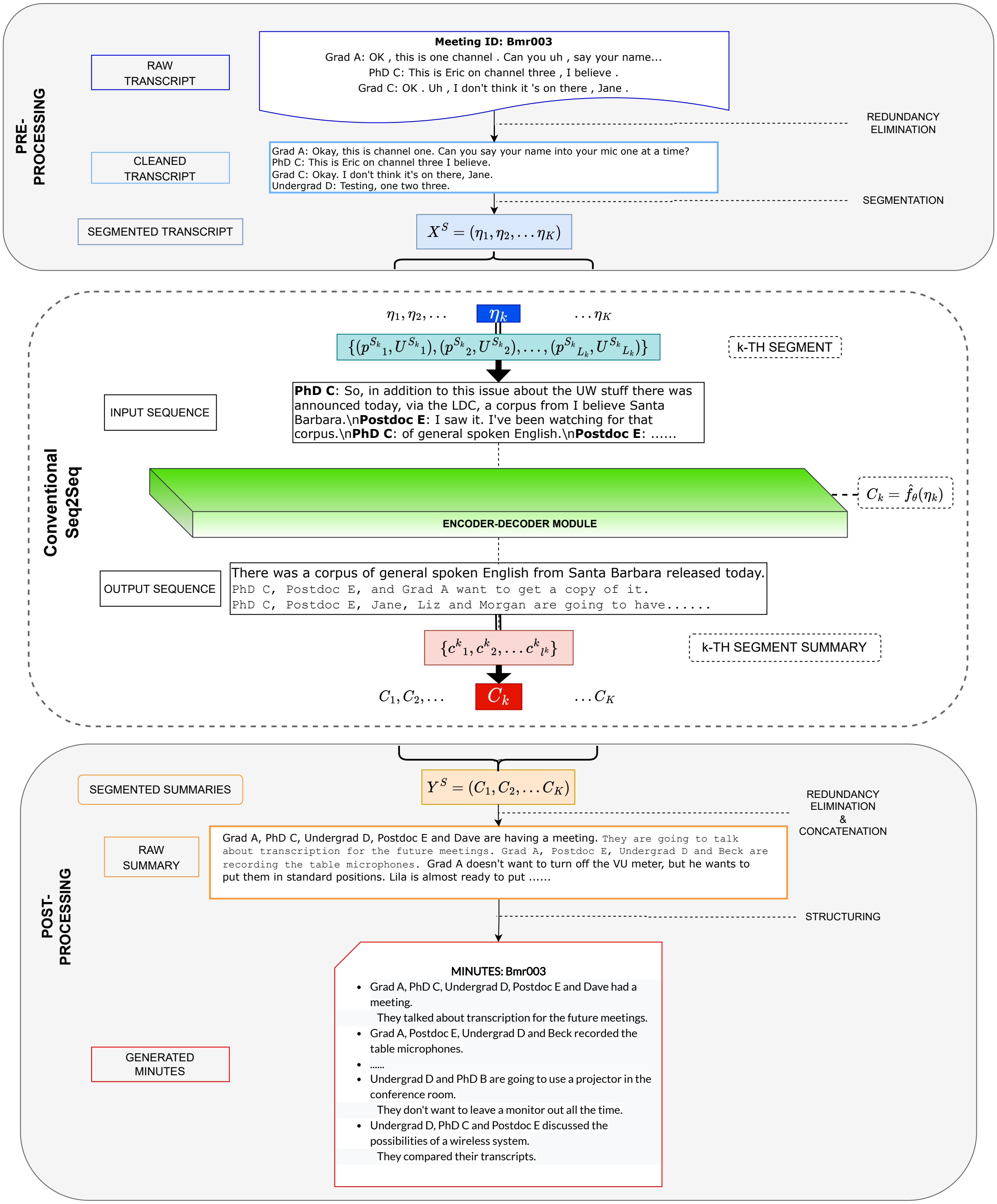
3 Methodology

Our current approach is inspired by one of the system submissions (Shinde et al., 2021) in the AutoMin shared task (Ghosal et al., 2021). Initially, we pre-process the transcripts as described in Section 3.1, later utilize the fine-tuned dialogue summarization model (Section 3.2), and finally, we post-process the outputs (Section 3.3). We describe the datasets used for the fine-tuning and evaluation in Section 4. We also provide automatic and human evaluation discussions and error analysis in Section 5. Kindly refer to Figure 1 for the entire system architecture.

3.1 Pre-Processing

Raw transcripts (directly from the ASRs) would require a good amount of pre-processing before one can proceed with the downstream tasks (automatic minuting in our case). In our experiments, the raw transcripts were already processed by human annotators to remove any inconsistencies during the respective corpora development. We discuss the steps employed for our use case on the already processed datasets.

Redundancy Elimination. Since current summarization models are not trained to eliminate redundancies and are often capped to specific input lengths, they struggle to process a long sequence of multi-speaker utterances and the dispersed information that comes with them (Ghosal et al., 2022b). We leverage specific pre-processing methods and em-



Meeting ID: Bmr003
Grad A: OK , this is one channel . Can you uh , say your name...
PhD C: This is Eric on channel three , I believe .
Grad C: OK . Uh , I don't think it 's on there , Jane .

Grad A: Okay, this is channel one. Can you say your name into your mic one at a time?
PhD C: This is Eric on channel three I believe.
Grad C: Okay. I don't think it's on there, Jane.
Undergrad D: Testing, one two three.

PhD C: So, in addition to this issue about the UW stuff there was announced today, via the LDC, a corpus from I believe Santa Barbara.\n**Postdoc E**: I saw it. I've been watching for that corpus.\n**PhD C**: of general spoken English.\n**Postdoc E**: ......

There was a corpus of general spoken English from Santa Barbara released today.
PhD C, Postdoc E, and Grad A want to get a copy of it.
PhD C, Postdoc E, Jane, Liz and Morgan are going to have......

Grad A, PhD C, Undergrad D, Postdoc E and Dave are having a meeting. They are going to talk about transcription for the future meetings. Grad A, Postdoc E, Undergrad D and Beck are recording the table microphones. Grad A doesn't want to turn off the VU meter, but he wants to put them in standard positions. Lila is almost ready to put ......

MINUTES: Bmr003

- Grad A, PhD C, Undergrad D, Postdoc E and Dave had a meeting.
  They talked about transcription for the future meetings.
- Grad A, Postdoc E, Undergrad D and Beck recorded the table microphones.
- ......
- Undergrad D and PhD B are going to use a projector in the conference room.
  They don't want to leave a monitor out all the time.
- Undergrad D, PhD C and Postdoc E discussed the possibilities of a wireless system.
  They compared their transcripts.

Figure 1: Architecture of the proposed *Automatic Minuting* pipeline.

ploy utterance cleaning and redundancy elimination based on thresholds to tackle this issue.

Consider a transcript with speaker-utterance pairs, $X^0 = \{(p_1^0, U_1^0), (p_2^0, U_2^0), ..., (p_L^0, U_L^0)\}$, where $p_j^0 \subset P$, $1 \leq j \leq L$, is the j-th speaker and $U_j^0 = (w_1^j, w_2^j ..., w_{l_j}^j)$ is the tokenized sequence of the j-th utterance; where $\{w_i^j\}$ represents the i-th token from the j-th utterance. For the j-th tokenized utterance, $U_j^0 = (w_1^j, w_2^j ..., w_{l_j}^j)$ from the transcript, we generate a cleaned sequence, $U_j^c = (W_1^j, W_2^j ..., W_{L_j}^j)$, by eliminating repetitions, pauses and known special symbols for unarticulated sounds, unintelligibility, disfluency markers, and similar disruptions. We filter the utterances using custom stopwords set S that we define from various meeting transcripts from currently available corpora like AMI (McCowan et al., 2005), ICSI (Janin et al., 2003), and the dataset from AutoMin (Nedoluzhko et al., 2022). By this, we obtain the filtered utterance $U^f = (U^c \backslash S)$ and the corresponding context ratio R, which expresses how much the utterance was shortened by dropping stopwords compared to the cleaned version:

$$R = |U^f| / |U^c| \tag{1}$$

Ultimately, our processed transcript X' comprises utterances U_i^c where the ratio of non-stop-words R_i is big enough, i.e. $R_i \geq \alpha$ (α being a predefined threshold ratio).

Linear Segmentation. Current summarization models limit the length of input sequences they can process (Singh et al., 2021), so they cannot process the full-length transcripts in our data. Our approach here is simple: it breaks the transcripts into blocks with a uniform token length. We experiment with token lengths: 512, 768, and 1024, respectively.

Topical Segmentation. The linear segmentation technique is problematic whenever important information on a topic falls into the subsequent segment. To address this limitation, we experiment with two methods for topic-aware segmentation: Depth-Scoring (adopted from Solbiati et al. (2021)) and the TextTiling algorithm by (Hearst, 1993).

For Depth Scoring, we use a window of k_w segments, capping each segment to $\hat{L} = 60$ words and setting topic change threshold τ to 0.5 (these are tunable hyperparameters, kindly refer to Solbiati et al. (2021) for details). Let us consider Figure 2. For a transcript with N turns, we obtain their contextualized embeddings from an encoder. We apply max pooling on this embedding space.

For a pair of neighboring windows of segments, one consisting of turns $k - k_w$ till k, and the other of turns k till $k + k_w$, we obtain the cosine similarity, sim_k between the embeddings pooled across all segments in the respective windows. For a series of neighbouring window similarity scores $\hat{s} = (sim_{k_w}, ...sim_{N-k_W})$, we compute the depth scores as $dp_k = \frac{hl(k)+hr(k)-2sim_k}{2}$ where $hl(k)$ and $hr(k)$ are the highest similarity score on the left and right side of the k^{th} element in the series of similarity scores. We deduce the topic change indices with the help of the obtained window-similarity scores and depth scores. Following are the variations one can use while determining the topic change indices.

- **Segment-window capping.** With this approach, we compute the topic change indices as:

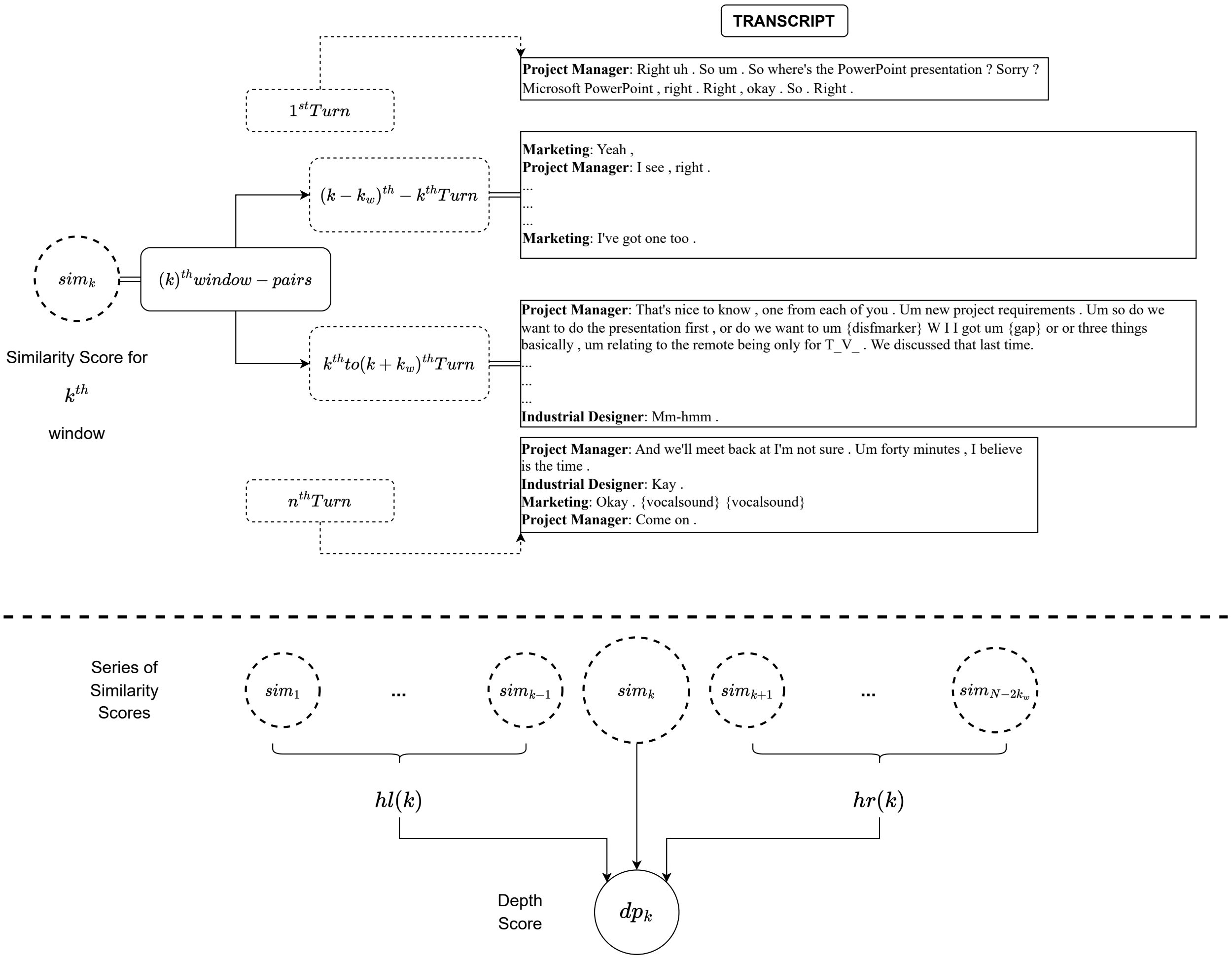
  $$T_{ds} = \{i \in [0, M] | sim_{k_w+i} \leq \mu_s - \sigma_s\} \tag{2}$$

  where T is the set of topic-change indices μ_s and σ_s are the mean and variance of the sequence, $M = N - k_W$ is the number of windows, $sim_{k_{w+i}}$ is the similarity score of the $i^t h$ window.

- **TextTiling.** TextTiling is a method to subdivide texts into multi-paragraph units representing passages or subtopics by leveraging lexical co-occurrence and distribution patterns. Here, we use TextTiling to identify major subtopic shifts. After computing the window similarity scores, we use the TextTiling method to compute the segments in a transcript. For a series of depth scores $D = (d_1, d_2, ...d_{N-k_w})$, we compute the topic change indices as:

  $$T_{tt} = \{i \in [1, M] | d_i \geq \tau\} \tag{3}$$

Through one of the three approaches (linear, depth-scoring, or text tiling), we obtain the segmented transcript $X^S = (\eta_1, \eta_2, ...\eta_K)$ where $\eta_k = \{(p_1^{S_k}, U_1^{S_k}), (p_2^{S_k}, U_2^{S_k}), ..., (p_{L_k}^{S_k}, U_{L_k}^{S_k})\}$ is the sequence of speaker-utterance pairs belonging to that segment.



Project Manager: Right uh . So um . So where's the PowerPoint presentation ? Sorry ? Microsoft PowerPoint , right . Right , okay . So . Right .

Marketing: Yeah ,
Project Manager: I see , right .
...
...
...
Marketing: I've got one too .

Project Manager: That's nice to know , one from each of you . Um new project requirements . Um so do we want to do the presentation first , or do we want to um {disfmarker} W I I got um {gap} or or three things basically , um relating to the remote being only for T_V_ . We discussed that last time.
...
...
...
Industrial Designer: Mm-hmm .

Project Manager: And we'll meet back at I'm not sure . Um forty minutes , I believe is the time .
Industrial Designer: Kay .
Marketing: Okay . {vocalsound} {vocalsound}
Project Manager: Come on .

Figure 2: Illustration of segment-windows, and calculation of similarity and depth scores referred in Section 3.1

We concatenate the speaker labels $p_{\bullet}$ with the corresponding utterances $U_{\bullet}$ and then across all items in the given segment back to the form of a single dialogue transcript. We then pass each of these plain text segments to one of the root summarization modules (see Section 3.2).

3.2 Summarization

We choose the pre-trained BART model (Lewis et al., 2019) in the summarization module in our pipeline. BART performs best among the other summarization models we tested, generating fluent and readable meeting minutes. Other summarization models include T5 (Raffel et al., 2019), Pegasus (Zhang et al., 2020), and RoBERTa2RoBERTa (Rothe et al., 2020). We fine-tune all these models on popular dialogue summarization datasets before integrating them into our pipeline.

BART is a denoising autoencoder for pretraining sequence-to-sequence models. The model is trained by corrupting text in an arbitrary noising function and then teaching it to reconstruct the original text. BART's ability to use source-side bi-directionality when operating on sequence generation tasks encourages its use for text summarization.

We pass the input sequence obtained from the preprocessing module through the summarization module. Again, for k-th segment, it returns a summary $C_k = \{c_1^k, c_2^k, ...c_{l_k}^k\}$, where c_i^k is the i-th summary line of the k-th segment. We rejoin all the segment summaries $Y^S = (C_1, C_2, ...C_K)$ to get the raw summary text.

Experimental Configuration We do not train any models from scratch but finetune most of them on

Datasets	# Dialogues	# Turns	# Speakers	# Turn Len.	# Len. of Dialogue	# Summary Len.	Compression
SAMSum	16.4K	11.2	2.4	9.1	124.0	23.4	81.12%
DialogSum	13.5K	9.5	2.0	15.8	168.5	25.8	84.70%
MediaSum	463.6K	30.0	6.5	49.6	1553.7	14.4	99.00%
AMI	137	535.6	4.0	10.4	5,570.4	321	94.24%
ICSI	59	819.0	6.3	10.5	8,567.7	576	93.28%
ELITR Corpus	124	254.4	5.8	9.7	8,890.8	387	95.65%

Table 1: Statistics of the dialogue and meeting summarization datasets we employ in our experiments. The top part is the larger summarization datasets we use for fine-tuning our models, the bottom part is theb meeting summarization datasets we use for model selection and testing. The reported statistics are averages across entire corpora. Lengths are in words. The compression ratio indicates how much the dialogue is shortened into the summary.

Datasets	Instances	Doc. Len.	Summ. Len.	% Comp.	%novel unigram
XSum	226.0K	488	27	94.5%	37.8%
CNN/DM	311.0K	906	63	93.0%	16.9%
R-TIFU	7.9K	641	65	89.9%	43.8%

Table 2: Document summarization datasets used for fine-tuning.

the data described in Section 4 below. For most models, a single Tesla K80 GPU is sufficient. Few larger models like BART-large and T5-large require multi-GPU training on NVIDIA GTX 1050 Ti or single GPU training on the NVIDIA A100-PCI-E-40GB variant. Training for individual finetuning procedures takes less than 2 hours, while warm-starting takes approximately 0.5 hours, depending on the dataset used. The hyperparameters and model configurations are consistent with the default values used during the pretraining of respective models. We set the finetuned BART on inference and generate our text with $num\_beams = 4, top\_k = 0.5$ and no limit on 'max_length'. We provide the hyperparameters and model configuration details in our code repository.

3.3 Post-Processing

After the main summarization, we use sentence compression methods, including swapping shortened phrases and pronouns and splitting longer sentences into two for improved readability. In our proposed pipeline, for each summary line, we filter out a set of unique entities (speaker names, project/corporation names, and location details). Further, we use a token-count threshold τ_{token} of 10 to include only those summary-sentences which are quantitatively informative enough (i.e., consisting of a minimum of τ_{token} number of tokens).

4 Dataset Description

Our work uses two types of data sources (see Table 1): (1) for fine-tuning summarization models, see Section 4.1, and (2) for the choice of the best setup and final evaluation of the minuting task, see Section 4.2.

4.1 Datasets for Fine-tuning Summarization Module

Here, we choose from some of the popular abstractive summarization datasets. Primarily, we use the dialogue summarization corpora: SAMSum (Gliwa et al., 2019), DialogSum (Chen et al., 2021), and MediaSum (Zhu et al., 2021).

Additionally, we use document summarization datasets XSum Narayan et al. (2018), CNN/DM Nallapati et al. (2016) and R-TIFU Kim et al. (2018), see Table 2. Their high compression ratio ("% Comp.") can potentially train the models to generate sequences more selectively, thus automatically eliminating redundancies.

4.2 Target Datasets: Automatic Minuting/Meeting Summarization

We primarily use ELITR Minuting Corpus (Nedoluzhko et al., 2022) for comparison with other systems. We further experiment on popular meeting summarization datasets: AMI (McCowan et al., 2005) and ICSI (Janin et al., 2003), due to similarities in the two tasks. AMI and ICSI come from staged product design meetings in companies, academic group meetings in schools, and similar arrangements. Each instance has a transcription of the entire dialogue and is annotated with a meeting summary and human-identified topic boundaries (except for ELITR Minuting Corpus). These meeting transcripts are extremely long, have a

	AMI			ICSI		
Models/Metrics	R-1	R-2	R-SU4	R-1	R-2	R-SU4
(A) Baselines and Comparing Systems						
Random	35.13	6.26	13.17	29.28	3.78	10.29
Cluster Rank (Garg et al., 2009)	35.14	6.46	13.35	27.64	3.68	9.77
Extractive Oracle	39.49	9.65	13.20	34.66	8.00	10.49
PGNet (See et al., 2017)	40.77	14.87	18.68	32.00	7.70	14.46
(B) Our best-performing setups						
bert2bert-cnndm-samsum	40.72	10.10	**27.13**	35.03	7.35	**24.48**
bart-xsum-dialogsum	42.40	10.34	17.67	36.95	6.94	13.68
t5-dialogsum	42.71	11.05	18.34	37.01	7.48	13.68
bart-xsum-samsum*	**45.17**	**13.30**	20.33	**38.75**	**8.51**	14.98
(C) State-of-the-art systems in Meeting Summarization						
HMNet (Zhu et al., 2020)	53.02	18.57	**24.85**	46.28	10.60	**19.12**
DialogLM (Zhong et al., 2021)	**53.70**	**19.60**	-	**49.50**	**12.50**	-
Summ^N (Zhang et al., 2021)	53.40	20.30	-	48.80	12.20	-

Table 3: ROUGE-1, ROUGE-2, ROUGE-SU4 scores of generated summaries on AMI and ICSI datasets. *→'bart-xsum-samsum' stands for our proposed model finetuned on the XSum corpus, further finetuned on the SAMSum corpus. Results in (C) are reproduced from the respective papers.

turn-based structure, and have multiple occurrences of redundant words and utterances.

Table 1 shows the relevant statistics of the dialogue and meeting summarization datasets that we use in our experiments.

5 Evaluation

In this section, we present the evaluation of our proposed pipeline in terms of automatic metrics in Section 5.1 and human evaluation metrics in Section 5.2. We compare our proposed pipeline with different summarization algorithms, finetune on combinations of abstractive summarization datasets, and report our performance on ELITR Minuting Corpus, AMI, and ICSI meeting summarization datasets.

5.1 Automatic Evaluation

For automatic evaluation, we make use of popular text summarization evaluation metrics. We report ROUGE (Lin, 2004) variants, namely ROUGE-1, ROUGE-2, ROUGE-SU4, which measures the overlap of unigrams, bigrams, and unigrams plus skip-bigrams (with max. skip of 4), respectively. We also provide METEOR (Banerjee and Lavie, 2005) scores which reward matching stems, synonyms, and paraphrases and not just exact matches.

5.2 Human Evaluation

To evaluate the quality of our output, we carry out a human evaluation of our minutes and compare it with the best-performing model outputs from the AutoMin 2021 shared task. Since we were the AutoMin shared task organizers, we had access to the human evaluators who also evaluated the system submissions in AutoMin. Six human evaluators rated our minutes in terms of *Adequacy, Grammaticality and Fluency* scores on a Likert scale of 5 (we report the average scores) (Ghosal et al., 2021). Because automatic metrics for text summarization evaluation have various shortcomings and are not apt to judge the quality of meeting minutes (Ghosal et al., 2022b), we attribute more importance to human evaluation, although the annotators were judging only our outputs in this run, without immediate comparison to AutoMin system outputs.

5.3 Results and Analysis

We discuss the experimental results and analyze the performance of our system in this section.

Table 3 compares the ROUGE scores of earlier models with our best setup (bart-XSum-samsum with linear segmentation). With no prior fine-tuning on AMI and ICSI meeting datasets, our pipeline outperforms several earlier approaches, including the popular Pointer Generator network (See et al., 2017) and the Extractive Oracle. However, the state-of-the-art models: HMNet (Zhu et al., 2020), DialogLM (Zhong et al., 2022) and SUMM-N (Zhang et al., 2021) are still superior in terms of the quantitative metrics.

Table 4 compares the automatic and human eval-

	Automatic Evaluation			Human Evaluation		
Model	R-1	R-2	R-L	Adequacy	Grammatical	Fluency
Ours-bart-xsum-samsum (Current Model)	**0.40±0.09**	**0.11±0.02**	0.18±0.03	**4.46/5.00**	**4.45/5.00**	**4.18/5.00**
Team ABC (Shinde et al., 2021)	0.33±0.08	0.08±0.04	**0.19±0.06**	3.98±0.73	4.45±0.37	4.27±0.55
Team Hitachi (Yamaguchi et al., 2021)	0.26±0.09	0.08±0.03	0.14±0.05	4.25±0.46	4.34±0.41	3.93±0.57

Table 4: Performance of our pipeline in comparison to the two best-performing participating systems at the **AutoMin Shared Task on the newly released ELITR Minuting Corpus.**

Model	R-1	R-2	R-SU4	BERTScore	METEOR
bart-xsum-samsum	**45.2**	**13.3**	**20.3**	**0.60**	**20.6**
bart-xsum-dialogsum	42.4	10.3	17.7	0.59	18.6
bart-base-samsum	39.9	11.2	16.1	**0.60**	15.1
bart-base-mediasum	33.2	7.0	11.3	0.55	14.0

Table 5: Comparison of the BART-based model setups with different finetuning datasets on the AMI test set.

Model	Pk	WinDiff	ROUGE-1	METEOR
Random	0.61	0.75	-	-
TextTiling	0.39	0.41	43.40	18.1
Capped	**0.34**	**0.35**	42.50	16.7
Linear (768)	0.44	0.50	**45.17**	**20.6**

Table 6: Comparison of the 'bart-xsum-samsum' model with different segmentation methods on AMI dataset

uation scores of AutoMin participating systems and our proposed model on ELITR Minuting Corpus (Nedoluzhko et al., 2022). Our model outperforms others on each of the metrics, confirming our pipeline's effectiveness. However, we must mention here that the scores of AutoMin systems were taken directly from Ghosal et al. (2021) and not remeasured in our annotation. The annotator pool was almost the same, and they had access to the AutoMin participant minutes. However, it is unlikely that they compared the AutoMin participants' outputs with the current system-generated output, so their evaluation scales could have shifted.

Table 5 shows the performance of our pipeline when used with different summarization models based on BART on the AMI test set. Our best-performing combination outscores the next by almost 3 points in terms of ROUGE-1; however, other model variants still perform close to the proposed approach. From the setups we tested, the best finetuning procedure starts with XSum and continues with the SAMSum dataset.

As we mentioned earlier, our model fine-tuned on the SAMSum corpus offers a better generation quality than those trained on other datasets. We attribute this to the fact that the dialogues in the SAMSum dataset are relatively simplistic and much more straightforward than those in DialogSum and MediaSum. The conversations are comparatively shorter and better reflect a conventional multiparty dialogue situation, leading to a better match between the training and testing conditions.

We also notice the differences caused by the training datasets used before the finetuning phase. Having a high compression ratio and novel word percentage, datasets like XSum demonstrate an extremely abstractive nature of summarization. Although the source text in XSum (Narayan et al., 2018) is longer than the dialogue instances from datasets like SAMSum (Gliwa et al., 2019), the summaries are relatively shorter. We observe a similar difference when we train the model on the XSum dataset compared to other datasets like the Reddit-TIFU (Kim et al., 2018) and the CNN/DailyMail (Nallapati et al., 2016). The generated minutes are relatively short, with more novel words and paraphrased sentences, qualities that are critical in abstractive summarization.

Furthermore, we carry out ablation experiments to check the effectiveness of our three segmentation methods (Table 6). We report the performance on the AMI dataset, which comes with reference segmentation of transcripts. We use the Pk (Beeferman et al., 1999) and WinDiff (Pevzner and Hearst, 2002) methods to evaluate the segmentation accuracy and report ROUGE-1 and METEOR scores on AMI transcripts.[1] We find ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) score surprisingly higher with linear segmentation (segments of at most 768 tokens). This result can be due to: with the topical segmentation schemes, several snippets of small talk and irrelevant information from the meeting make it to the final output, thereby affecting the overall scores.

[1] Ground truth segmentation of meeting transcripts are available for the AMI corpus.

Input: Transcript

Project Manager : Right uh . So um . So where's the PowerPoint presentation ? Sorry ? Microsoft PowerPoint , right . Right , okay . So . Right . Okay , so we've got uh so we've got new project requirements . Um . So basically we've got three things , and we've got forty minutes in which to uh {disfmarker} for this meeting to uh to discuss the various options . Um . Three presentations .
Industrial Designer : We have a {disfmarker} I guess we have a presentation each , 'cause I've got one . Um .
User Interface: Yeah .
Marketing : Yeah .
Project Manager : That's nice to know , one from each of you . Um new project requirements . Um so do we want to do the presentation first ,or do we want to um {disfmarker} W I I got um {gap} or or three things basically , um relating to the remote being only for T_V_ . We discussed that last time
. . .
Industrial Designer : Kay .
Marketing : Okay . {vocalsound} {vocalsound}

Output: Minutes

- Industrial Designer explains to Project Manager how the remote works.
  They have their own logo, color scheme and they have to incorporate it on the product.
  The current expansion of channels in the process of taking place, certainly the button up, down, but how many channels do we have to prepare for and how many buttons do we need?
- The Project Manager, Industrial Designer, Marketing and User Interface discussed the remote control functions.
- Industrial Designer will put delay to allow multiple numbers.
- Project Manager, User Interface, Marketing and Industrial Designer discussed the design of the remote control.
- Marketing, Industrial Designer, Project Manager and User Interface discussed the design of remotes for older televisions.
  There are 16 buttons on the front of the device: power button, standby, channel, volume, up, down, brightness, contrast, and a mute.
  On the back it will have the channel tuner, brightness and contrast.
- The project manager did the minutes of the meeting.

Figure 3: Sample output from our pipeline on a document from AMI corpus: Meeting Id-ES2014b

Figure 3 shows a sample minute generated from our pipeline approach. The transcript corresponds to 'ES2014b' from the AMI dataset. As we can see, the generated minute is coherent with the discussions from the meeting.

5.4 Error Analysis

We qualitatively examine and find that our outputs show the following categories of errors (Figure 4).

- **Made-up entities**. Anonymization of discrete entities in transcripts (e.g., LOCATION7, PERSON4, Marketing Manager) is consistent in most transcripts and minutes of our test datasets. Since no such anonymization is apparent in SAMSum, this sometimes results in the generation of made-up entities that are initially not part of that transcript.

- **Absence of context in summary**. Sometimes, the generated summary could use pronouns or other referring expressions from the transcript without ensuring that the element they are referring to is actually present in the summary. However, this issue is rare and did not occur in our final test runs.

- **Incomplete phrases**. Although less, we notice occurrences of incomplete sentences. These generally belong to those parts of the transcripts where the utterances either had missing punctuation or hesitations and interruptions on the speaker's part.

6 Conclusion

In this paper, we explore the use of large pre-trained language models fine-tuned on dialogue summarization datasets to automatically generate meeting minutes. We evaluate our proposed BART-based pipeline approach on the recently released corpus for automatic minuting (ELITR Minuting Corpus) as well as on the earlier AMI and ICSI meeting summarization corpora. We utilize existing multiparty meeting summarization datasets.

Our pipelined approach is promising and certainly puts up a case for further investigations to employ large language models for this challenging task. In future work, we would like to optimize our existing pipeline by replacing extractive filtering and utterance-level topic segmentation with an end-to-end method.

Acknowledgement

This work has received funding from the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

Case-1: Made-up entities

Instance - *“PhD A PhD F, PhD C and PhD F are discussing the encoding of things with time and data.”*

Explanation - It seems like as a normal summary line with correct grammar and readability. Consulting the transcript, we find out that ‘PhD C’ is not a real speaker, ’Grad C’ is the real speaker here. Hence, this is an error due to anonymization.

Instance - *“Marketing, Project Manager, Industrial Designer and Project Manager are meeting to....”*

Explanation - The ‘Project Manager’ was mentioned in the transcript once but it appears twice in the summary line. We attribute this to anonymization in the finetuning data, which collapses two people’s names into two very similar identifiers; the model then infers that repeating a similar (or even identical) entry is sometimes desired.

Case-2: Absence of context

Instance - *“PhD D discovered that on the wireless ones, you can tell if it’s picking up breath noises...”*

Explanation - The wording of the summary line uses a referring expression (‘the wireless ones’) without providing its referent in the surrounding lines.

Case-3: Incomplete phrases

Instance 1 - *“they don’t match well with the operating behavior of the — Marketing, Industrial Designer, Project Manager are discussing the design of the remote control”*

Instance 2 - *“They have decided to start with the black and white version. They will use double A or triple A batteries, rubberized buttons, a plastic casing for the plastic shell, a variety of designs, — Marketing Project Manager, Industrial Designer, User Interface and Project Manager are discussing the design of a keychain.”*

Explanation - Due to interruptions in the speech, the transcripts sometimes break one speech act into several utterance — often marked with a hyphen. This reflects in the model outputs as shown with a ‘—’ separator.

Figure 4: Error instances from the pipeline-generated summaries illustrating the error cases discussed in Section 5.4.

References

[Banerjee and Lavie2005] Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

[Beeferman et al.1999] Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.

[Celikyilmaz et al.2018] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

[Chen and Bansal2018] Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*.

[Chen and Metze2012] Yun-Nung Chen and Florian Metze. 2012. Integrating intra-speaker topic modeling and temporal-based inter-speaker topic modeling in random walk for improved multi-party meeting summarization. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[Chen et al.2021] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

[Cho et al.2019] Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. *arXiv preprint arXiv:1906.00072*.

[Chopra et al.2016] Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

[Fauville et al.2021] G. Fauville, M. Luo, A.C.M. Queiroz, J.N. Bailenson, and J. Hancock. 2021. Zoom Exhaustion & Fatigue Scale. *Computers in Human Behavior Reports*, 4:100119.

[Garg et al.2009] Nikhil Garg, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani Tür. 2009. Clusterrank: a graph based method for meeting summarization. Technical report, Idiap.

[Ghosal et al.2021] Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. Overview of the First Shared Task on Automatic Minuting (AutoMin) at Interspeech 2021. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.

[Ghosal et al.2022a] Tirthankar Ghosal, Marie Hledíková, Muskaan Singh, Anna Nedoluzhko, and Ondrej Bojar. 2022a. The second automatic minuting (automin) challenge: Generating and evaluating minutes from multi-party meetings. page TBA, july.

[Ghosal et al.2022b] Tirthankar Ghosal, Muskaan Singh, Anja Nedoluzhko, and Ondřej Bojar. 2022b. Report on the SIGDial 2021 Special Session on Summarization of Dialogues and Multi-Party Meetings (SummDial). *SIGIR Forum*, 55(2), mar.

[Gliwa et al.2019] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

[Hearst1993] Marti A Hearst. 1993. Texttiling: A quantitative approach to discourse segmentation. Technical report, Citeseer.

[Janin et al.2003] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

[Jia et al.2020] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631.

[Kim et al.2018] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks. *arXiv preprint arXiv:1811.00783*.

[Lebanoff et al.2019] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. *arXiv preprint arXiv:1906.00077*.

[Lewis et al.2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[Li et al.2019] Manling Li, Lingyu Zhang, Richard J Radke, and Heng Ji. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *57th Conference of the Association for Computational Linguistics*.

[Lin2004] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[Liu and Chen2019] Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5460–5466.

[Liu and Chen2021] Zhengyuan Liu and Nancy F Chen. 2021. Dynamic sliding window for meeting summarization. *arXiv preprint arXiv:2108.13629*.

[Liu and Lapata2019] Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

[Liu et al.2019] Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F Chen. 2019. Topic-aware pointer-generator networks for summarizing spoken conversations. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821. IEEE.

[McCowan et al.2005] Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. 2005. The ami meeting corpus. In *Proceedings of the 5th international conference on methods and techniques in behavioral research*, volume 88, page 100. Citeseer.

[Nallapati et al.2016] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

[Narayan et al.2018] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

[Nedoluzhko et al.2022] Anna Nedoluzhko, Muskaan Singh, Marie HledÃkovÃ¡, Tirthankar Ghosal, and OndÅ™ej Bojar. 2022. Elitr minuting corpus: A novel dataset for automatic minuting from multi-party meetings in english and czech. In *Proceedings of the Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France, June. European Language Resources Association.

[Pevzner and Hearst2002] Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

[Raffel et al.2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

[Rothe et al.2020] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

[Rush et al.2015] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

[See et al.2017] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

[Shinde et al.2021] Kartik Shinde, Nidhir Bhavsar, Aakash Bhatnagar, and Tirthankar Ghosal. 2021. Team ABC @ AutoMin 2021: Generating Readable Minutes with a BART-based Automatic Minuting Approach. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 26–33.

[Singh et al.2021] Muskaan Singh, Tirthankar Ghosal, and Ondrej Bojar. 2021. An empirical performance analysis of state-of-the-art summarization models for automatic minuting. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 50–60, Shanghai, China, 11. Association for Computational Lingustics.

[Solbiati et al.2021] Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. *arXiv preprint arXiv:2106.12978*.

[Wang et al.2020] Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*.

[Xu and Durrett2019] Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. *arXiv preprint arXiv:1902.00863*.

[Xu et al.2019] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive text summarization. *arXiv preprint arXiv:1910.14142*.

[Yamaguchi et al.2021] Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Ken ichi Yokote, and Kenji Nagamatsu. 2021. Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 41–48.

[Zhang et al.2020] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

[Zhang et al.2021] Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H Awadallah, Dragomir Radev, and Rui Zhang. 2021. Summˆ n: A multi-stage summarization framework for long input dialogues and documents. *arXiv preprint arXiv:2110.10150*.

[Zhao et al.2019] Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, Min Yang, and Deng Cai. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, pages 3455–3461.

[Zhong et al.2019] Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. *arXiv preprint arXiv:1907.03491*.

[Zhong et al.2021] Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Dialoglm: Pre-trained model for long dialogue understanding and summarization. *arXiv preprint arXiv:2109.02492*.

[Zhong et al.2022] Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Dialoglm: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.

[Zhu et al.2020] Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. *arXiv preprint arXiv:2004.02016*.

[Zhu et al.2021] Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. *arXiv preprint arXiv:2103.06410*.