Team Iterate @ AutoMin 2023 - Experiments with Iterative Minuting

František Kmječ
Faculty of Mathematics and Physics
Charles University, Czech Republic
frantisek.kmjec@gmail.com

Ondřej Bojar
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Czech Republic
bojar@ufal.mff.cuni.cz

Abstract

This report describes the development of our system for automatic minuting created for the AutoMin 2023 Task A organized by Ghosal et al. (2023). As a baseline, we utilize a system based on the BART encoder-decoder model paired with a preprocessing pipeline similar to the one introduced by Shinde et al. (2022). We then further explore the possibilities for iterative summarization by constructing an iterative minuting dataset from the provided data, finetuning on it and feeding the model previously generated minutes. We also experiment with adding more context by utilizing the Longformer encoder-decoder model (Beltagy et al., 2020), finetuning it on the SAMSum dataset (Gliwa et al., 2019). Our submitted solution is of the baseline approach, since we were unable to match its performance with our iterative variants. With the baseline, we achieve a ROUGE-1 score of 0.368 on the ELITR minuting corpus (Nedoluzhko et al., 2022a) development set. We finally explore the performance of Vicuna (Chiang et al., 2023) 13B quantized language model for summarization.

1 Introduction

Meeting minuting is the process of writing down the important contents discussed while reducing the overall length. It is generally necessary to create minutes to keep people who were not able to attend up to date and to have a reference to come back to. However, writing meeting minutes is a tedious process requiring a lot of concentration. Additionally, most meetings lack a dedicated notetaker, therefore the additional cognitive load is placed on the meeting participants who are already under stress. Since the COVID pandemic hit, many meetings have shifted to the online space, and with the rise of the large language models, it is becoming technically possible to automate the tedious and taxing minuting process.

Numerous approaches for automatic minuting were shown at the first AutoMin (Ghosal et al., 2021), most of them utilizing a pre-trained transformer model like BART (Lewis et al., 2020) or PEGASUS (Zhang et al., 2020). Such models however have their limitations, especially with their input size being constrained to 512 or 1024 tokens by the quadratic complexity of the attention mechanism.

In our approach, we explored possible solutions to the issue of short context length, namely iterative summarization and the Longformer model. We utilized a solution inspired by the winning one at AutoMin 2021 by Shinde et al. (2021) as a baseline. Finally, we experimented with the new Vicuna models, but we were unable to obtain the results by the task deadline.

2 Related works

In AutoMin 2021, the approaches with best results were of Team Hitachi (Yamaguchi et al., 2021) and Team ABC (Shinde et al., 2021). Both of these teams solved the problem of limited model input length in a different way. Team ABC splits the transcript into fixed-size segments, preprocesses them with a rule-based system, then summarizes each segment separately using a BART model. They then filter the output to remove redundancies and concatenate the result. On the other hand, Team Hitachi utilize a segmenter based on the Longformer architecture with a LSTM recurrent network on top which assigns utterances to different topics. These topics are then summarized using a BART model and results are concatenated to form the final minutes. The approach from Team Hitachi scores slightly higher on adequacy while the system of Team ABC is ranked higher in fluency and grammatical correctness. Notably, neither of these systems used the ELITR minuting corpus data for training.

We also list some notable systems that were not a part of the AutoMin 2021. Summ^N by Zhang et al. (2022) works by generating a coarse summary

Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges, pages 114–120
September 11–15, 2023. ©2023 Association for Computational Linguistics

in multiple stages and then generating a final summary from them. It has a variable-length input as it can scale its number of stages. QMSum by Zhong et al. (2021) utilizes a locate-then-summarize approach, which works by first locating parts of the transcript with a common topic and then summarizing them separately. In this, the approach is similar to Team Hitachi's.

3 Baseline system

We use a baseline approach inspired by system of Team ABC from 2021. We use a pipeline with a BART model finetuned on the XSum (Narayan et al., 2018) and SAMSum datasets with a simple rule-based preprocessing system. The transcript is first cleaned of filler words and less common characters are removed to make the summary more fluent with the preprocessing code of Shinde et al. (2022). To satisfy the input length limitation of the BART model, the pipeline then splits the transcript into chunks of roughly 512 tokens. Each of those chunks is then summarized into a separate bullet point. The resulting minutes are a concatenation of the chunk summaries.

4 Iterative approaches

One of the biggest challenges for summarization transformer language models is the limited input length. This naturally limits the amount of context the model can process and therefore can severely interfere with the quality of the generated minutes, especially for conversations with a common topic that span several thousand tokens. There are approaches that try to counter this, notably the Longformer mechanism, which modifies the attention mechanism to reduce the complexity, and others mentioned in section 2.

For humans, a natural approach to creating meeting minutes is an incremental one. A notetaker listens to the conversation taking place and writes down the agreed-upon points, all the while keeping in mind what he has already noted. Our intention was to imitate such a process. The summarization model would be fed a chunk of a transcript together with several previously generated minute points to both satisfy the input length constraint of the transformer models while providing the needed context for the minutes.

4.1 Data pre-processing

To the best of our knowledge, there are no datasets publicly available for transcript summarization where there would be known alignment between a minute bullet point and a transcript chunk. Therefore, we needed to fabricate our own training dataset from available data.

We preprocessed and used data from the English part of ELITR minuting corpus (Nedoluzhko et al., 2022b) provided as a part of the competition. The dataset contains 120 meetings, each with at least one transcript and at least one minute. The average length of the transcripts is around 7000 words while the minutes are on average 373 words long. The corpus is split into four sets: `train`, `dev`, `test` and `test2` with 84, 10, 18 and 8 meetings respectively. We utilize `train` for training and `dev` for a development set.

We cleaned the transcripts of fillers and stopwords using the same preprocessing approach as with the baseline model. We then split each transcript into 512 token chunks with 256 token overlap between neighbouring chunks, dividing the chunks between utterances so as to preserve fluency. We also split the corresponding minutes into sequences of three consecutive bullet points.

We then aligned the minute chunks to the transcript chunks. We explored two approaches, one using document similarity metric from the Spacy library introduced by Honnibal et al. and the other one using ROUGE-1 precision scores. In both cases, for every minute chunk we calculated the metric between it and every transcript chunk and picked the piece of transcript that maximized the metric. By manual inspection of a sample of aligned chunks, we found the ROUGE-1 alignment to be more reliable.

The resulting dataset had the last bullet point of the minute chunk as the target and the concatenation of two previous bullet points and the transcript as the input. The dataset statistics can be found in table 1.

5 Methodology

5.1 Iterative BART

We utilized the same BART model weights as in the baseline. We finetuned on our created dataset with learning rate $\alpha = 2 \cdot 10^{-5}$ and with weight decay of 0.01 for one epoch.

After training and testing the model on some development transcripts, we found out that we are

dataset	n. samples	transcript	prepended minutes	target minutes
train	6014	189.21 ± 123.67	19.05 ± 15.47	9.95 ± 9.74

Table 1: Iterative dataset statistics. The transcript, prepended minutes and target minutes columns give the average amount of words in the respective categories and the standard deviation.

unable to prevent the model from infinitely repeating the past outputted minutes, effectively being stuck in a loop. We attribute this to two factors. Firstly, there was not much training data, with our dataset creation process yielding about 6000 samples. Secondly, the training data quality was not very good and probably unsuitable for the limited context length of the BART model input. Many of the target bullet points consisted of information that cannot be obtained from a short chunk of the transcript, like the list of participants, purpose of the whole meeting or a purpose of a large section of a meeting.

5.2 Iterative LED

To counteract the input length limits of the BART model, we experimented with the LED model for iterative summarization. LED stands for Longformer Encoder Decoder and is a modification of the BART model. It utilizes the Longformer attention mechanism as a drop-in replacement of the classic self-attention mechanism, allowing it to take input up to 16384 tokens in length, which is in most cases longer than the transcript provided as part of ELITR minuting corpus.

We utilized the LED-large model pretrained on Arxiv long document dataset introduced by Cohan et al. (2018). We then finetuned on the SAMSum dataset for 1000 steps with learning rate $5 \cdot 10^{-5}$ with the Adam optimizer.

For further finetuning, we modified the iterative dataset, utilizing the entire transcript instead of transcript chunks as input. We then trained following the same procedure as for the BART model. However, while testing the model, we found it did not provide the improvement we hoped for, as the LED was still looping and generating the same minutes all over again, rendering the approach unusable for practical applications. Overall, we found the iterative solutions to be infeasible, especially because of the lack of suitable training data.

5.3 Non-iterative LED model

As we did not manage to pass the baseline or get to a functional solution with our iterative approaches, we turned towards using the SAMSum-finetuned LED model in a manner similar to the BART baseline. We then generated the minute points by first feeding the model the first whole transcript, then the transcript without first 1024 tokens, then without 2048 tokens, and so on. We cut off parts of the transcript do distinguish the inputs and force the model to focus on something new in the next summary point. The results were promising, with roughly comparable ROUGE and BERT scores to the ones posed by the baseline. However, the system produced a summary whose bullet points were a lot less compact. We assume this is due to the fact that the LED model was not pretrained on the XSum dataset, therefore it did not learn to shorten the input as well as the BART model.

5.4 Experiments with Llama quantized models

In early 2023, Llama models were proposed by Touvron et al. (2023). Llama is a family of decoder-only foundational language models similar in architecture to GPT (Radford and Narasimhan, 2018). The architecture includes optimizations from subsequent successful models like GPT-3 (Brown et al., 2020) or PaLM (Chowdhery et al., 2022). Due to the successes of models with similar architecture, for example by Hájek (2021) with GPT-2 for Czech summarization, we were intrigued to try the models for minuting. Because the weights are public, many open-source modifications are available. Recently, with the help of the GPT4All library (Anand et al., 2023), it has become easy to generate outputs from such large language models using quantization.

We experimented with prompting the 4-bit quantized 13 billion parameter Vicuna model. Vicuna is a version of the Llama model specifically finetuned on user-model conversations from ShareGPT.[1] It is meant to follow users' instructions, functioning as a chatbot. The model has a limited context length, therefore the same preprocessing and splitting into chunks as with the baseline model is needed.

We used the prompt of "`Please summarize the following transcript with 2 bullet points starting with *. Write just the bullet`

[1] `https://sharegpt.com/`

points, nothing more." The input chunk length chosen was 768 tokens at maximum. The results were promising, with most minutes being more relevant and fluent than the ones generated by the baseline. The Vicuna model sometimes does not listen to the prompt instruction, instead generating a response like "I am sorry, but I cannot write a response to this prompt as it is incomplete and I am not sure what the prompt is asking for. Please provide a complete and clear prompt, so I can assist you.", but in the majority of responses, the task is fulfilled correctly. However, we were unable to compute the results by the task deadline, therefore we did not submit it to the competition.

6 Evaluation and output samples

Commonly used approaches for automatic evaluation include ROUGE and BERTScore, but these often fail to represent the real quality of a meeting minute, as they are unable to fully represent the informational content. We therefore fall back to a combination of manual evaluation (coarsely assessing the relevance, coverage and fluency of the generated minutes) and the automatic metrics of ROUGE and BERTScore. We place most emphasis on the manual qualitative evaluation on the development set of ELITR. We also ran automatic evaluations on the test and test2 sets.

We found the baseline model to perform better than LED in all the automatic metrics we computed, as shown in table 2. The baseline also generates more concise summaries. We observed that the LED model has a tendency to refrain from drawing conclusions and only generating sentences with questionable informational content. An example of this is They need to organize it according to the dates, the names of the sessions and then complement it with the with the minutes and then discuss what in what way do they need to rename it or rename it. The sentence is neither grammatically correct nor very informative. This is indicated also by the smaller ROUGE recall scores. Notably, the model also struggled with copying words from the input, with the word ORGANIZATION commonly misspelled as ORGANATION in the output. Due to all these factors, we submitted the baseline model as our competition one.

After the competition deadline, we also evaluated the outputs of the pipeline with the Vicuna model prompted for meeting summarization. We found out it was similar to the the other models in BERTScore but performed worse on ROUGE-1 precision and a bit better on ROUGE-1 recall, being more consistent across all the evaluation datasets. Qualitatively, the result sample we examined seemed more informative than the output of our other models. Had the results been available by the submission deadline, we would be submitting them. However, sometimes the outcome of the Vicuna model was not summarization at all; sometimes it ignored the transcript completely and generated a generic outcome. This could be automatically filtered or, better, fixed with better prompting or finetuning. It should also be noted that the results from the Vicuna model were on average considerably longer than of the baseline model, as also shown in table 2.

We would like to point out that all the generated minutes differ in essence to the target minutes provided as a part of the dev set, as they often focus more on what is happening at the current time in the transcript instead of summarizing the outcomes and agreed upon points. We find this to be a major disparity that we believe will only be fixed with much increased context length. A comparison of samples of successfully generated summary points for the same transcript chunks can be found below.

Baseline: This week PERSON11 is working on the collection is business for English. He has downloaded 1970 to 1980 pdf files which is extracted from the OTHER3 thing of the office. 15000 sentences is crawled automatic using automatic clover for English, OTHER3. He needs to clean all the data

LED: PERSON11 is trying to record the call to see if they are able to summarize it automatically. They are trying to make it sound like they are discussing the issue of censorship. They need to apply the same type of adaptation for the NLP, PROJECT2 on the 12th and the 17th of February. They also need to adapt the vocabulary of the PERSON7 talkage. They want to use English as English models. They will use it for the first dry run of a workshop on the Monday seminar.

Model	avg. words	BERT	R-1p	R-1r	R-2p	R-2r
dev set						
Baseline	471	0.785	0.225	0.368	0.06	0.106
LED	661	0.778	0.220	0.334	0.04	0.09
Vicuna	698	0.766	0.187	0.389	0.05	0.119
test set						
Baseline	543	0.750	0.156	0.287	0.03	0.06
LED	704	0.729	0.165	0.258	0.022	0.05
Vicuna	764	0.74	0.144	0.33	0.03	0.08
test2 set						
Baseline	537	0.781	0.292	0.335	0.09	0.12
LED	704	0.765	0.292	0.26	0.06	0.08
Vicuna	732	0.774	0.254	0.343	0.07	0.11

Table 2: comparison of the output lengths and metrics on ELITR **dev**, **test** and **test2** sets

Vicuna PERSON11 is working on a business project for OTHER3, which involves cleaning and organizing a large amount of data in text format.
PERSON14 is collaborating with PERSON6 on a language model for the project, and they are discussing how to use the model for organizing the data.

7 Conclusion

Although we were unable to pass the baseline with our approaches, we have several interesting findings.

- We found that although iterative summarization is a possibly promising approach, the needed training data is not yet available. Training on ELITR minuting corpus data proved difficult, mostly due to the non-incremental character of the available minutes.

- We successfully finetuned the LED model on conversation summarization and gained comparable results to the baseline on some inputs. However, we were unable to see the benefits of the larger context length it offers. We believe this is due to the character of available conversation summarization datasets, which rarely have inputs longer than a thousand tokens.

- We have shown that Vicuna models can be successfully prompted to perform summarization of transcripts, even though the results can be unreliable. We found that the results are often more fluent and relevant than outputs of the smaller BART model, even though the model has not been specifically finetuned on the summarization task.

7.1 Future work

We believe the Llama models show promise for summarization and minuting; therefore, we think further finetuning on the SAMSum and XSum datasets could improve the results by a large margin. Bigger models could be finetuned using low-rank adaptation training as proposed by Hu et al. (2021), shown in practice on the StackLLama model from Beeching et al. (2023).

We also believe that the Longformer model could be successfully used for summarization if it is adapted to a smaller subtask of the minuting. As seen in the provided training data in the ELITR minuting corpus, the minutes often have very specific sections for a general topic of the meeting, the attendees, the agreed upon next actions and tasks that are given to separate participants. Such sections cannot be well generated by an approach that only has short chunks as context. Therefore, a separate Longformer model could be trained for each of those subtasks that would take full advantage of the whole transcript context. Such an approach would be similar to the one created by Team Hitachi at AutoMin 2021.

References

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. https://github.com/nomic-ai/gpt4all.

Edward Beeching, Younes Belkada, Kashif Rasul, Lewis Tunstall, Leandro von Werra, Nazneen Rajani, and Nathan Lambert. 2023. Stackllama: An rl fine-tuned llama model for stack exchange question and answering.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Tirthankar Ghosal, Ondřej Bojar, Marie Hledíková, Tom Kocmi, and Anja Nedoluzhko. 2023. Overview of the second shared task on automatic minuting (automin) at inlg 2023. In *Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges*. Association for Computational Linguistics.

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.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spacy: Industrial-strength natural language processing in python.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Adam Hájek. 2021. Automatic text summarization [online]. SUPERVISOR: doc. RNDr. Aleš Horák, Ph.D.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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*, abs/1808.08745.

Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Tirthankar Ghosal, and Ondřej Bojar. 2022a. ELITR minuting corpus: A novel dataset for automatic minuting from multi-party meetings in English and Czech. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3174–3182, Marseille, France. European Language Resources Association.

Anna Nedoluzhko, Muskaan Singh, Marie Hledíková, Ghosal Tirthankar, and Ondřej Bojar. 2022b. ELITR minuting corpus. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

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.

Kartik Shinde, Tirthankar Ghosal, Muskaan Singh, and Ondrej Bojar. 2022. Automatic minuting: A pipeline method for generating minutes from multi-party meeting proceedings. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 691–702, Manila, Philippines. De La Salle University.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summ^n: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.