A Sliding-Window Approach to Automatic Creation of Meeting Minutes

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Fei Liu

Computer Science Department
University of Central Florida, Orlando, FL 32816

{jjkoay,alexroustai,xd.zangyiwu}@knights.ucf.edu
feiliu@cs.ucf.edu

Abstract

Meeting minutes record any subject matters discussed, decisions reached and actions taken at meetings. The importance of minuting cannot be overemphasized in a time when a significant number of meetings take place in the virtual space. In this paper, we present a sliding window approach to automatic generation of meeting minutes. It aims to tackle issues associated with the nature of spoken text, including lengthy transcripts and lack of document structure, which make it difficult to identify salient content to be included in the meeting minutes. Our approach combines a sliding window and a neural abstractive summarizer to navigate through the transcripts to find salient content. The approach is evaluated on transcripts of natural meeting conversations, where we compare results obtained for human transcripts and two versions of automatic transcripts and discuss how and to what extent the summarizer succeeds at capturing salient content.

1 Introduction

Meetings are ubiquitous across organizations of all shapes and sizes, and it takes a tremendous effort to record any subject matters discussed, final decisions reached and actions taken at meetings. With the rise of remote workforce, virtual meetings are more important than ever. An increasing number of video conferencing providers including Zoom, Microsoft Team, Amazon Chime and Google Meet allow meetings to be transcribed (Martindale, 2021). However, without automatic minuting, consolidating notes and creating meeting minutes is still regarded as a tedious and time-consuming task for meeting participants. There is thus an urgent need to develop advanced techniques to better summarize and organize meeting content.

Meeting summarization has been attempted on a small scale before the era of deep learning. Previous work includes efforts to extract utterances and keyphrases from meeting transcripts (Galley, 2006; Murray and Carenini, 2008; Gillick et al., 2009; Liu et al., 2009), detect meeting decisions (Hsueh and Moore, 2008), compress or merge utterances to generate abstracts (Liu and Liu, 2009; Wang and Cardie, 2013; Mehdad et al., 2013) and make use of acoustic-prosodic and speaker features (Maskey and Hirschberg, 2005; Zhu et al., 2009; Chen and Metze, 2012) for utterance extraction. The continued development of automatic transcription and its easy accessibility have sparked a renewed interest in meeting summarization (Shang et al., 2018; Li et al., 2019; Koay et al., 2020; Song et al., 2020; Zhu et al., 2020; Zhong et al., 2021), where neural representations are explored for this task. We believe the time is therefore ripe for a reconsideration of the approach to automatic minuting.

It may be tempting to apply neural abstractive summarization to meetings given its remarkable recent success on summarization benchmarks, e.g., CNN/DM (See et al., 2017; Chen and Bansal, 2018; Gehrmann et al., 2018; Laban et al., 2020). However, the challenge lies not only in handling hallucinations that are seen in abstractive models (Kryscinski et al., 2019; Lebanoff et al., 2019; Maynez et al., 2020) but also the models' strong positional bias that occurs as a consequence of fine-tuning on news articles (Kedzie et al., 2018; Grenander et al., 2019). Neural summarizers also assume a maximum sequence length, e.g., Perez-Beltrachini et al. (2019) use the first 800 tokens of the document as input. With an estimated speaking rate of 122 words per minute (Polifroni et al., 1991), it indicates that the summarizer may only process a relatively short transcript – about 5 minutes in duration.

In this paper, we instead study an extractive meeting summarizer to identify salient utterances from the transcripts. It leverages a sliding window to navigate through a transcript of any length and a neural abstractive summarizer to find salient local content. In particular, we aim to address three key questions: (1) what are suitable window and stride sizes? (2)

Proceedings of NAACL-HLT 2021: Student Research Workshop, pages 68–75
June 6–11, 2021. ©2021 Association for Computational Linguistics

can the abstractive summarizer effectively identify salient local content? (3) how should we consolidate local abstracts into meeting-level summaries? Our approach is intuitive and appealing, as humans make a sequence of local decisions when navigating through very long recordings. It is evaluated on transcripts of natural meeting conversations (Janin et al., 2003), where we obtained human transcripts and two versions of automatic transcripts produced by the AMI speech recognizer (Hain et al., 2006) and Google Cloud's Speech-to-Text API.[1] Our contributions in this paper are as follows.

- We study the feasibility of a sliding-window approach to automatic generation of meeting minutes that draws on a pretrained neural abstractive summarizer to make local decisions on utterance saliency. It does not require any annotated data and can be extended to meetings of various types and domains.
- We examine results obtained from human transcripts and two versions of automatic transcripts, and show that our summarizer either outperforms or performs comparably to competitive baselines given both automatic and human evaluations. We discuss how and to what extent the summarizer succeeds at capturing salient content.[2]

2 Background: The BART Summarizer

BART (Lewis et al., 2020) has demonstrated strong performance on neural abstractive summarization. It consists of a bidirectional encoder and a left-to-right autoregressive decoder, each contains multiple layers of Transformers (Vaswani et al., 2017). The model is pretrained using a denoising objective that, given a corrupted input text, the encoder strives to learn meaningful representations and the decoder reconstructs the original text using the representations. In this study, we use `BART-large-cnn` as a base summarizer. It contains 12 layers in each of the encoder and decoder and uses a hidden size of 1024. The model is then fine-tuned on the CNN dataset for abstractive summarization.

There are two obstacles that should be overcome in order for BART to generate meeting summaries from transcripts. Firstly, BART is trained on written text, rather than spoken text. The pretraining data contain 160G of news, books, stories, and web text. It remains unclear if the model can effectively identify salient content on spoken text and, how it is to reduce lead bias that is not as frequent in spoken text as in news writing (Grenander et al., 2019). Secondly, a transcript can far exceed the maximum input length of the model, which is restricted by the GPU memory size. This is the case even for recent variants such as Reformer (Kitaev et al., 2020) and Longformer (Beltagy et al., 2020).

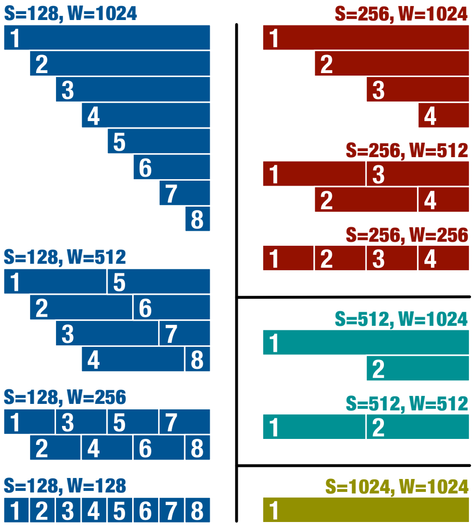

Figure 1: A total of 10 combinations of window (`W`) and stride (`S`) sizes examined in this study. A small stride allows a text region to be repeatedly visited by the summarizer. The numbers (`1-8`) indicate local windows.

3 Our Approach

A sliding-window approach to generating meeting minutes is appealing because it breaks lengthy transcripts into small and manageable local windows, allowing a set of "mini-summaries" to be produced from such windows which are then assembled into meeting-level summaries. There are two essential decisions to be made when using a sliding window. Firstly, one must decide on the size of the local window. Our window size is bounded by the maximum sequence length of BART as the utterances in a window are concatenated into a flat sequence that serves as input to it. We consider a number of window sizes with `W`={128, 256, 512, 1024} tokens. Secondly, a transcript may be partitioned into non-overlapping or partially overlapping windows. We set the stride size to be `S`={128, 256, 512, 1024} tokens to support both ($\texttt{W} \geq \texttt{S}$). When they are of equal size, a transcript is divided into a sequence of non-overlapping windows.

In Figure 1, we enumerate all 10 combinations of window and stride sizes. For example, we ex-

[1] `https://cloud.google.com/speech-to-text`

[2] Our transcripts and system outputs are released publicly at `https://github.com/ucfnlp/meeting-sliding-window`

Input	System	ROUGE-1			ROUGE-2			Summary Len	
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	%Uttrs	#Wrds
Human	KL-Sum	57.2	31.9	40.8	19.0	10.6	13.6	19.6	754
	SumBasic	61.6	67.1	62.4	24.8	28.1	25.6	19.6	1,730
	LexRank	36.8	84.3	50.9	21.2	49.2	29.4	19.6	3,528
	TextRank	28.2	91.6	42.9	19.4	63.5	29.5	19.6	4,954
	(Koay et al., 2020)	52.6	81.0	62.5	29.4	46.1	35.2	21.7	2,321
	SW (HumanTrans)	**36.5**	**90.9**	**51.9**	**23.2**	**58.4**	**33.1**	19.6	3,741
ASR	(Shang et al., 2018)	27.6	36.3	31.0	4.4	5.6	4.8	n/a	n/a
	(Koay et al., 2020)	51.3	78.6	61.3	25.7	39.9	30.9	16.7	2,224
	SW (AMI ASR)	**36.1**	**88.3**	**51.2**	**19.4**	**47.8**	**27.6**	18.2	3,514
	SW (Google ASR)	**61.9**	**65.7**	**62.9**	**26.5**	**28.1**	**26.9**	23.2	1,460

Table 1: Results on the ICSI test set using human transcripts and two versions of automatic transcripts (AMI vs. Google) as input. The length is defined as percentage of selected utterances over all utterances of the meetings and average number of words in the summaries. The sliding-window (SW) summarizer uses (S=128, W=1024).

periment with four window sizes of 128, 256, 512 and 1,024 tokens using the same stride size of 128 tokens, shown in dark blue (left). A larger window gives additional context to BART for recognizing salient content. Using a window of 1,024 and stride of 128 tokens allow each utterance of the transcript to be visited 8 times, whereas using a window of 512 tokens reduces that to 4 times.

Consolidation. BART abstracts generated from local windows cannot be simply concatenated to form meeting-level summaries as they contain redundancy. When local windows are partially overlapping, they can cause the same content to be included in different abstracts. Instead, we identify *supporting utterances* of each abstract from the transcript. Particularly, we compute the ROUGE-L scores between each utterance in the window and the abstract. If the utterance is longer than 5 tokens, achieves a recall score $r > 0.5$ and precision score $p > 0.1$, we call it a supporting utterance.[3] The same utterance can support multiple abstracts. We include an utterance into the meeting summary if it is designated as the supporting utterance for at lease one local abstract. It lends flexibility and improves ease of consolidation of local abstractive summaries produced by BART.

4 Results

Dataset. Our experiments are performed on the ICSI meeting corpus (Janin et al., 2003), which is a challenging benchmark for meeting summarization. The corpus contains 75 meeting recordings, each is about an hour long. We use 54 meetings for training and report results on the standard test set containing 6 meetings. Each training meeting has been annotated with an extractive summary. Each test meeting has three human-annotated extractive summaries, which we use as gold-standard summaries. The original corpus include human transcripts and automatic speech recognition (ASR) output generated by the AMI ASR team (Hain et al., 2006). We are able to generate a new version of automatic transcripts by using Google's Speech-to-Text API as an off-the-shelf system.[4] Comparing results on different versions of transcripts allows us to better assess the generality of our findings.

Our baselines include both general-purpose extractive summarizers and meeting-specific summarizers. LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004) are graph-based extractive methods. SumBasic (Vanderwende et al., 2007) selects sentences if they contain frequently occurring content words. KL-Sum (Haghighi and Vanderwende, 2009) adds sentences to the summary to minimize KL divergence. We additionally experiment with two meeting summarizers. Shang et al. (2018) group utterances into clusters, generate an abstractive sentence from each cluster using sentence compression, then select best elements from these sentences under a budget constraint. Koay et al. (2020) develop a supervised BERT summarizer to identify summary utterances.

We report test set results in Table 1, where system summaries are compared with gold-standard extractive summaries using ROUGE metrics (Lin, 2004). The summary length is computed as the percentage of selected utterances over all utterances of the meetings and average number of words per test summary. This information is reported wherever

[3] The thresholds were determined heuristically on the training set by observing the resulting alignment.

[4] Due to lack of documentation, we are unable to report the word error rates of Google and AMI speech recognizers.

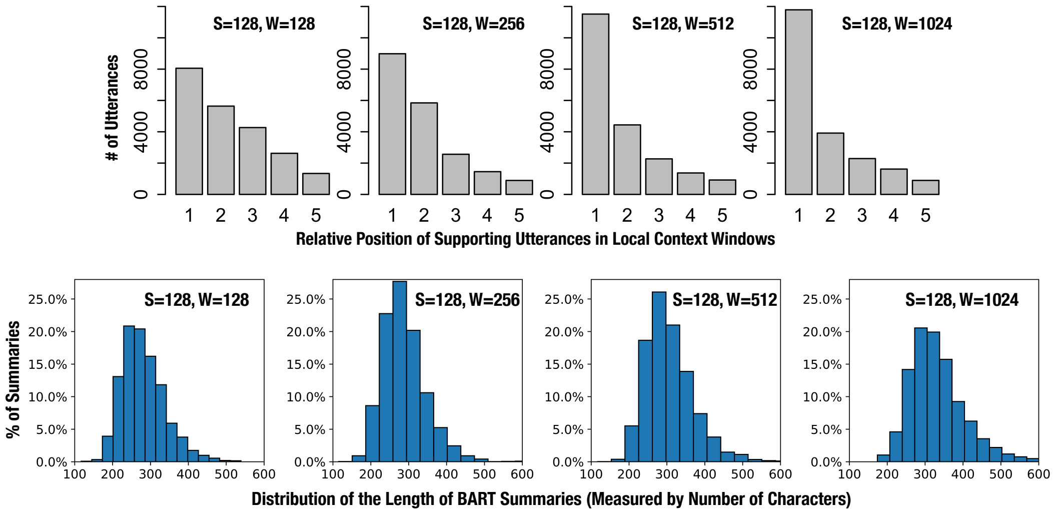

Figure 2: (Top) Relative position of supporting utterances in their local windows. We find that BART tends to take summary content from the first 150-200 tokens of the input sequence. With a large window (W=1024), summary content is likely taken from the first 20% of input. (Bottom) Length distribution of BART abstracts, measured by number of characters. Using windows ranging from 128 to 1024 tokens, the average abstract length increases from 281 to 332 characters, i.e., 56 to 66 words assuming 5 characters per word for English texts (Shannon, 1951). Results are obtained on the ICSI training set using human transcripts.

available, and baseline summarizers are set to output the same number of summary utterances as the sliding-window (SW) approach. Our SW approach can outperform or perform comparably to competitive baselines when evaluated on human and ASR transcripts. We note that Koay et al. (2020) utilize a supervised BERT summarizer, whereas our SW approach is unsupervised.[5] It does not require annotated summaries and only uses the training set to determine window and stride sizes (S=128, W=1024, details later).

A closer examination reveals that Google transcripts contain substantially less filled pauses (*um, uh, mm-hmm*), disfluencies (*go-go-go away*), repetitions and verbal interruptions. The Google service also tends to produce lengthier utterances. Table 2 provides an example comparing human, AMI and Google transcripts. The summaries produced with Google transcripts contain fewer utterances and less number of words per summary. They achieve a higher precision and lower recall when compared to those of AMI and human transcripts.

We are curious to know where supporting utterances appear in the local windows. In Figure 2, we discretize the position information into 5 bins and plot the distributions for four settings that use different window sizes (W={128,256,512,1024}) but the same stride size (S=128). We observe that BART tends to select content from the first 150 to 200 tokens of the input and add them to the abstract. It indicates that the model exhibits strong lead bias even for spoken text, which differs from news writing (Grenander et al., 2019). Additionally, we examine the length of BART abstracts, measured by the number of characters in an abstract. Using windows from 128 to 1024 tokens, we find that the avg. abstract length increases from 281 to 332 characters, ≈56 to 66 words assuming 5 characters per word on average for English texts (Shannon, 1951). While a larger window can lead to a longer abstract, the abstract size is disproportionate to the window

Transcription	**Human**	**AMI**	**Google**
# of utter. per meeting	1330	1410	188
# of words per utterance	7.7	7.0	33.0
(**Human**) and um There one of our diligent workers has to sort of volunteer to look over Tilman's shoulder while he is changing the grammars to English			
(**AMI**) And um And they're one of our a The legend to work paris has to sort of volunteer to Look over time and shorter what he is changing that gram was to english			
(**Google**) and they are one of our diligent workers has to sit or volunteer to look over two months shoulder while he is changing the Grandma's to English			

Table 2: Compared to human and AMI transcripts, utterances produced by Google's transcription service are lengthier and there are fewer utterances per meeting.

[5] We use pyrouge with default options to evaluate all summaries. The scores are different from that of Koay et al. (2020) which removed stopwords during evaluation by using '-s'.

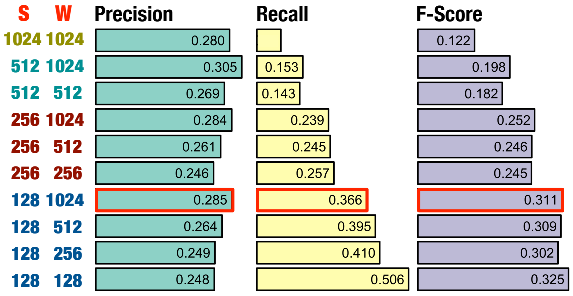

Figure 3: Precision, recall and F-scores of summary utterance selection using different combinations of stride (S) and window (W) sizes. Results are obtained on the ICSI training set using human transcripts. We find that (S=128, W=1024) attains a good balance between precision and recall, whereas using small, non-overlapping windows (S=128, W=128) yields high recall due to more utterances are included in the summary.

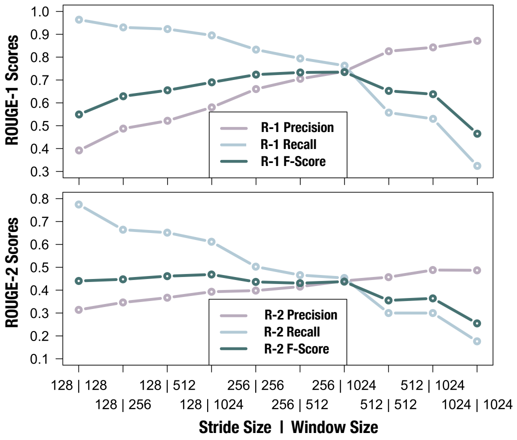

Figure 4: R-1 and R-2 scores when different combinations of stride (S) and window (W) sizes are used. Results are obtained on the ICSI training set for human transcripts. With (S=256, W=1024), we obtain balanced precision and recall scores. The best R-2 F-score is achieved with (S=128, W=1024).

size. These results are obtained on the training set using human transcripts as input.

In Figure 3, we investigate various combinations of stride (S) and window sizes (W) and report their precision, recall and F-scores on summary utterance selection. Similarly, the results are obtained on the training set using human transcripts as input. We highlight some interesting findings. We observe that a large context window (W=1024) tends to give high precision. A small window combined with small stride yields high recall due to more utterances are selected for the summary. For example, both settings (W=512, S=128) and (W=1024, S=256) allow an utterance to be visited 4 times. The former achieves a higher recall (0.395 vs. 0.239) due to its smaller window and stride sizes. In Figure 4, we show R-1 and R-2 scores obtained on the training set for all combinations of stride and window sizes. We find that recall scores decrease substantially using large stride sizes (>=512 tokens). With (S=256, W=1024), we obtain balanced precision and recall scores. The best R-2 F-score is achieved with (S=128, W=1024) which is used at test time.

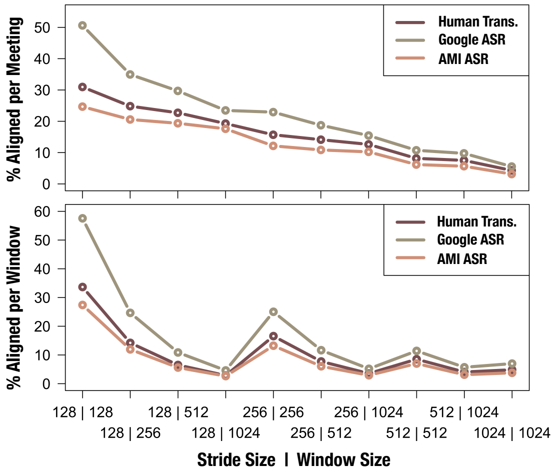

Figure 5: Percentage of supporting utterances per meeting (TOP) and per local window (BOTTOM). Results are obtained on the ICSI training set with different combinations of stride (S) and window (W) sizes, for human transcripts and two versions of automatic transcripts (Google vs. AMI).

System	Utterance Rating		
	Score-2	Score-1	Score-0
TextRank	8.58%	25.66%	65.77%
Supervised-BERT	11.35%	28.96%	59.69%
Sliding Window	**11.46%**	**26.11%**	**62.43%**

Table 3: Percentage of summary utterances rated as highly relevant (2), relevant (1) and irrelevant (0) by human evaluators. The systems for comparison are TextRank, a supervised BERT summarizer (Koay et al., 2020) and Sliding Window.

In Figure 5, we present the percentage of supporting (summary) utterances per meeting and per window, for various combinations of window and stride sizes. On human transcripts, we observe that combining small stride and window sizes (S=128, W=128) has led to ∼30% utterances to be selected per meeting. In contrast, (S=128, W=1024) selects 19% of the utterances. Human transcripts and automatic transcripts generated by AMI ASR appear to show similar behavior, but the Google transcriber breaks up utterances differently.

We further conduct a human evaluation on the six test meetings. Three human evaluators (two native speakers and a non-native speaker) are employed

Speaker	Utterance	BERT	SW	Gold
fn002	I - Hynek last week say that if I have time I can to begin to - to study	1	1	1
fn002	well seriously the France Telecom proposal to look at the code and something like that	1	1	1
me013	Mm-hmm.	0	0	0
fn002	to know exactly what they are doing because maybe that we can have some ideas	1	0	0
me013	Mm-hmm.	0	0	0
fn002	but not only to read the proposal. Look look	0	0	0
fn002	carefully what they are doing with the program and I begin to - to work also in that.	1	0	1
fn002	But the first thing that I don't understand is that they	0	1	1
fn002	are using	0	0	1
fn002	the uh log energy that this quite - I don't know why they have some	0	1	1
fn002	constant in the expression of the lower energy. I don't know what that means.	0	1	1
me018	They have a constant in there, you said?	0	1	0

Table 4: Extractive summaries produced by the sliding-window approach (SW) appear to read more coherently than those of the supervised BERT summarizer. Consecutive sentences in SW summaries are more likely to be associated with the same idea/speaker compared to supervised-BERT. "Gold" are ground-truth summary utterances.

for this task. They rate each summary utterance as highly relevant (2), relevant (1) or irrelevant (0) by matching the utterance with the meeting abstract provided by the ICSI corpus. The systems for comparison are SW, TextRank and the fully supervised BERT summarizer (Koay et al., 2020). In Table 3, we report the percentage of summary utterances assigned to each category (Fleiss' Kappa=0.29). Our summarizer obtains promising results. It outperforms TextRank and performs comparably to supervised-BERT. We find that the SW summarizer navigates through the transcript in an *equally detailed* manner. It leads to coherent and sometimes verbose summaries, compared to other extractive summaries. A snippet of the transcript and its accompanying summaries are shown in Table 4.

5 Conclusion

We investigate the feasibility of a sliding-window approach to generating meeting minutes and obtain promising results on both human and automatic transcripts. The approach does not require annotated data and it has a great potential to be extended to meetings of various domains. Our future work includes, in the near horizon, experimenting with a look-ahead mechanism to enable the summarizer to skip over insignificant transcript segments.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. We would also like to thank Kaiqiang Song for helping transcribe the meetings using Google's Speech-to-Text API. Xiaojin Dai was supported by NSF DUE-1643835. We thank Amazon for partially sponsoring the research and computation in this study through the Amazon AWS Machine Learning Research Award.

References

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

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Yun-Nung Chen and Florian Metze. 2012. Two-layer mutually reinforced random walk for improved multi-party meeting summarization. In *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012*, pages 461–466. IEEE.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 364–372, Sydney, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Daniel Gillick, Korbinian Riedhammer, Benoît Favre, and Dilek Hakkani-Tur. 2009. A global optimization framework for meeting summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4769–4772.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. In *Proceedings of* *the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6019–6024, Hong Kong, China. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Iain McCowan, Darren Moore, Vincent Wan, Roeland Ordelman, and Steve Renals. 2006. The 2005 ami system for the transcription of speech in meetings. In *Machine Learning for Multimodal Interaction*, pages 450–462, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pei-Yun Hsueh and Johanna D. Moore. 2008. Automatic decision detection in meeting speech. In *Machine Learning for Multimodal Interaction*, pages 168–179, Berlin, Heidelberg. Springer Berlin Heidelberg.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 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.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Dillon Burns, Alec Kerrigan, and Fei Liu. 2020. How domain terminology affects meeting summarization performance. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5689–5695, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5135–5150, Online. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining 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.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 261–264, Suntec, Singapore. Association for Computational Linguistics.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 620–628, Boulder, Colorado. Association for Computational Linguistics.

Jon Martindale. 2021. Google meet tips and tricks. *https://www.digitaltrends.com/computing/google-meet-tips-tricks/*.

Sameer Maskey and Julia Hirschberg. 2005. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *INTERSPEECH*, pages 621–624. ISCA.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings* *of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Raymond T. Ng. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, Sofia, Bulgaria. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Gabriel Murray and Giuseppe Carenini. 2008. Summarizing spoken and written conversations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 773–782, Honolulu, Hawaii. Association for Computational Linguistics.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. Association for Computational Linguistics.

Joseph Polifroni, Stephanie Seneff, and Victor W. Zue. 1991. Collection of spontaneous speech for the ATIS domain and comparative analyses of data collected at MIT and TI. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.

C. E. Shannon. 1951. Prediction and entropy of printed english. *Bell System Technical Journal*.

Yan Song, Yuanhe Tian, Nan Wang, and Fei Xia. 2020. Summarizing medical conversations via identifying important utterances. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 717–729, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*, 43(6):1606–1618.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed H. 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*.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Xiaodan Zhu, Gerald Penn, and Frank Rudzicz. 2009. Summarizing multiple spoken documents: finding evidence from untranscribed audio. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 549–557, Suntec, Singapore. Association for Computational Linguistics.