R-BASS: Relevance-aided Block-wise Adaptation for Speech Summarization

Roshan Sharma[1*], **Ruchira Sharma**[2], **Hira Dhamyal**[1], **Rita Singh**[1] and **Bhiksha Raj**[1,3]
[1]Carnegie Mellon University, [2]University of Massachusetts, Amherst
[3]Mohammed bin Zayed University of AI, Abu Dhabi
roshansh@cmu.edu

Abstract

End-to-end speech summarization on long recordings is challenging because of the high computational cost. Block-wise Adaptation for Speech Summarization (BASS) summarizes arbitrarily long sequences by sequentially processing abutting chunks of audio. Despite the benefits of BASS, it has higher compute time due to sequential processing of all blocks, regardless of whether they are relevant to the final summary. In this paper, we propose R-BASS, a new relevance-aware block-wise adaptation method. First, we introduce two approaches to automatically estimate block relevance based on lexical and semantic similarity between the block-level transcript and the summary. Experiments on the How2 dataset show that using ground truth relevance during inference improves efficiency by 63.9 % by dropping irrelevant blocks. Finally, we incorporate relevance scores into training using a novel relevance loss and relevance predictor, and the proposed R-BASS model makes it possible to drop 86.3 % of the blocks while retaining comparable performance, resulting in a 2.2x speedup over BASS.

1 Introduction

Generative models (Lakhotia et al., 2021; Brown et al., 2020) have revolutionized the field of artificial intelligence. Speech summarization (Hori et al., 2002; Rezazadegan et al., 2020; Murray et al., 2010; Palaskar et al., 2019; Li et al., 2019; Shang et al., 2018) is the task of taking in long input recordings, identifying parts of the speech with essential information, and generating a short textual summary that concisely conveys the important information. End-to-end speech summarization (Sharma et al., 2022; Matsuura et al., 2023; Jung et al., 2024) has been shown to improve performance over cascade models that first transcribe long recordings, and then summarize transcripts (Palaskar et al., 2019, 2021). However, such models are difficult to train on very large inputs owing to compute restrictions (Kano et al., 2023).

To address the challenge of long inputs, Block-wise adaptation for Speech Summarization (BASS) (Sharma et al., 2023) chunks the long input speech into blocks. These blocks are then processed independently, with the semantic context being passed across blocks to facilitate remembering information from past blocks. Though BASS has better performance and lower computational cost over training directly on long sequences and can process arbitrarily long sequences by updating summaries based on new acoustic information, processing of all relevant and irrelevant blocks is computationally inefficient. In this paper, we introduce *R-BASS*, a relevance-aware block-wise model that first predicts whether the new block of acoustic information is relevant to the summary before integrating new information only from relevant blocks into the semantic context.

To decide whether a given block is relevant or not, we analyze the acoustics of the block and the generated summary thus far. If the acoustic information within a new block possesses higher semantic similarity with the previously produced summary, we deem such blocks to be relevant. Then, we examine automatic methods to label the relevance of blocks based on lexical and semantic similarity with the block transcript. Lexical similarity involves looking at the number of words in the transcript of a given block that are present in the final summary. Semantic similarity is assessed by calculating the similarity between BERT (Kenton and Toutanova, 2019) embeddings of the given block's transcript and the summary. Finally, we devise a *relevance loss* that can be used to guide model predictions of relevance to be similar to the ones obtained by automatic annotations. From experiments on How2, *R-BASS* improves efficiency while retaining comparable performance.

[*] Author is now at Google

Findings of the Association for Computational Linguistics: NAACL 2024, pages 848–857
June 16-21, 2024 ©2024 Association for Computational Linguistics

2 Block-wise Adaptation for Speech Summarization (BASS)

BASS is implemented as an attention-based encoder-decoder model and comprises three main blocks : (1) Encoder, (2) Updater, and (3) Decoder. The input sequence X is represented as a sequence of T abutting blocks of length B, i.e., $X = [X^1, X^2, X^3 \cdots X^T]$. Given the i-th block of input X^i, the encoder computes a high-level latent representation H^i of the input. A semantic embedding S^i is used to represent semantic context from all blocks until the i-th block. An updater considers the previous semantic embedding S^{i-1} and the new acoustic information represented by H^i and produces the updated semantic embedding S^i for the current block. The decoder finally uses the updated semantic embedding S^i to obtain a hypothesis for the summary $\hat{Y}^i$.

From a probabilistic perspective, at each block, the BASS model estimates the probability of the summary at the i-th block Y^i given all prior speech features $X^{1:i}$ $\mathbb{P}(Y^i|X^{1:i}, Y^{1:i-1})$, which can be decomposed as shown in Equation 1. The terms in the decomposition $P(S^i|X^i)$ and $P(Y^i|S^i)$ are modeled by the encoder and decoder respectively.

$$\mathbb{P}(Y^i|X^{1:i}) = \mathbb{P}(Y^i|S^i)\mathbb{P}(S^i|S^{1:i-1}, X^i) \quad (1)$$

To make training tractable over arbitrarily large inputs, backpropagation is performed block-wise rather than utterance-wise. In the former, only tensors that pertain to the current block of acoustic input remain in the computational graph and GPU memory while the latter is infeasible for long inputs because all the tensors corresponding to all blocks in the recording would have to be stored in the computational graph and GPU memory. To perform block-level optimization, block-level targets are required to compute a loss. The i-th block X^i produces a block-level output Y^i which is compared to the reference summary for the entire recording Y to obtain the loss, and backpropagation follows.

To combine information from the prior block and the current encoded output, we first use *Concatenation*, which is a simple approach. In this the previous semantic embedding is concatenated along the sequence (time) dimension with the current acoustic embedding to produce the current semantic embedding. This mechanism preserves more information but can be less efficient than using a fixed size of semantic embedding for all blocks.

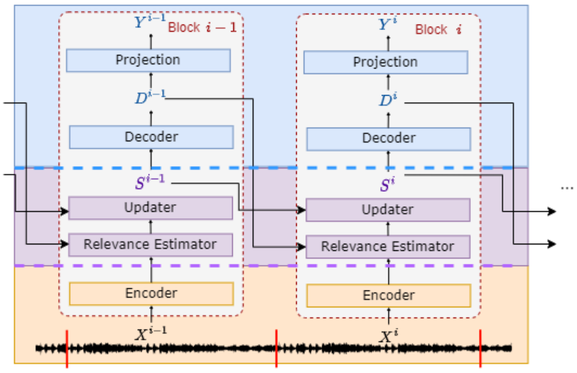

Figure 1: Proposed Relevance-Aware Block-wise Adaptation for Speech Summarization

BASS not only allows one to use standard self-attentions and avoid approximation errors, but it can also be efficient for streaming speech summarization, where summaries are expected to be updated given new acoustic information. However, it can result in longer training and inference times than methods that use efficient self-attention for offline training and inference. This can be mitigated by using the fact that all blocks are not equally useful to the summary. Since processing all blocks to generate the final output summary at the last block is computationally expensive, we propose *R-BASS*, a method to predict and use block-level relevance.

3 Proposed R-BASS

3.1 Overview

The fundamental idea behind *R-BASS* is to develop a mechanism to help the model learn when new acoustic information is relevant. When the acoustic information in a new block is relevant, we can update the semantic context to incorporate this information, and otherwise retain the same semantic context. This approach (1) saves time and memory and (2) ensures that the context we use across blocks is comprised solely of relevant information.

Since we aggregate context in the semantic space for BASS, decisions on relevance need to be made before updation. Figure 1 shows the model architecture for *R-BASS* where we insert a new relevance estimator in the semantic space. The goal of the relevance estimator is to predict whether the acoustic information present in the current block is relevant to the summary. During training, when we have access to the ground-truth summary, all we need to do is estimate the similarity between the ground-truth summary and the encoded speech representations. However, during inference, we do not have access to the reference summary to make decisions about relevance. Therefore, we approximate relevance during both training and inference by using the similarity between the summary from the previous output block $\hat{Y}^{i-1}$ and the output of the encoder for the current block, i.e., $Enc(X^i)$. Equation 2 shows how we compute relevance R^i of the i-th block using new speech information X^i, where Sim. stands for a similarity function.

$$R^i = \text{Sim.}(Y, X^i) \approx \text{Sim.}(\hat{Y}^{i-1}, Enc(X^i)) \quad (2)$$

Similarity, in general, can be computed using a myriad of mechanisms including cosine distance, however, since the previous summary $\hat{Y}^{i-1}$ and the current acoustic encoding $Enc(X^i)$ belong to different distributions, additional parameters are required to transform the vectors into a common space before computing similarity. We utilize a cross-attention mechanism between the previous summary and the current acoustics and obtain an attention-based context vector. Since relevance is modeled at the block level, we first obtain the temporal mean of this attended context. The mean attended context vector is then projected down to a single value that represents the probability that the current block is relevant.

Since backpropagation is performed at the block level for BASS, the previous semantic embedding is detached from the computational graph while processing the current block. That is, gradients do not flow through the past summary while computing relevance. To ensure that the encoder representations do not degrade when computing relevance, we detach the encoder representation $Enc(X^i)$ from the computational graph as well. In this way, the trainable attention and linear projection parameters used for computing relevance are the only parameters updated.

To ensure that model predictions of relevance are reasonable, we develop methods to automatically tag blocks as relevant and irrelevant. Then, we use these labels along with a relevance loss to fine-tune BASS so that it learns to accurately predict the relevance of blocks.

3.2 Labeling Relevance and R-BASS-Inf

To automatically label the relevance of blocks, we compare the reference summary with the ground-truth block-level transcript, rather than the input speech. Since both representations will be in the textual space, we can leverage textual similarity metrics to assess relevance. Humans generally annotate relevance by looking for: (a) common keywords between the transcript and summary, and (b) related sentences based on semantics. If the block-level transcript under consideration has words that are present in the summary, then the block may be considered relevant - we refer to this idea as *lexical similarity*. If the block-level transcript is related in intent or meaning to the summary, then the similarity between semantic embeddings of the block-level transcript and the summary is high, and the block is relevant – this is *semantic similarity*. We remove stop words using NLTK (Bird and Loper, 2004) before computing similarity metrics to avoid basing similarity on stop words.

Lexical Similarity: One of the ways to capture relevance is to assess word overlap. We calculate the ratio of the number of words in the current block's transcript that occur in the reference summary to the number of words in the reference summary. This ratio reflects the degree of lexical similarity. If the i-th block's transcript is denoted as T^i, and the reference summary is represented as Y, then the lexical similarity $LS(T^i, Y)$ can be written as shown in Equation 3.

$$LS(T^i, Y) = \frac{\#(y \in T^i | y \in Y)}{\#(y \in Y)}. \quad (3)$$

The ratio $LS(T^i, Y)$ represents the *degree of relevance*. However, in *R-BASS*, we focus only on whether or not a given block is relevant. Therefore, we apply a threshold $\tau = 0$ to convert $LS(T^i, Y)$ to a binary value.

Semantic Similarity: This metric captures similarity in the semantic space between the block-level transcript T^i and the reference summary Y. We extract BERT (Kenton and Toutanova, 2019) embeddings from the transcript and reference summary. The cosine similarity between the two embeddings is our measure of semantic similarity $SS(T^i, Y)$. This computation is described in Equation 4, where $\mathcal{B}()$ represents the BERT embeddings of the given text. We use $\tau = 0.4$ based on the data distribution to get binary values

$$SS(T^i, Y) = \text{cos-sim}(\mathcal{B}(T^i), \mathcal{B}(Y)) \quad (4)$$

To evaluate the quality of obtained pseudo-labels, we can use these relevance pseudo-labels directly during inference (*R-BASS-Inf*) to produce summaries using only relevant blocks. We compare this to a baseline that randomly selects a fraction of blocks without relying on any relevance metric, and another baseline that does not use relevance to validate whether the lexical and semantic similarity approaches accurately capture relevance.

Apart from using labels during inference, we can also train R-BASS models to predict relevance on a given block, and then use this predicted relevance to produce summaries during inference. Such a model is optimized using a relevance loss described in the next section.

3.3 Introducing Relevance Loss for Gated Attention

Now that we devised mechanisms to model relevance within the R-BASS model architecture and methods to obtain automatic annotations for relevance, we describe the *Relevance loss* used to train R-BASS. To estimate the true relevance R^i, for the i-th block, we obtain the Binary Cross-Entropy (BCE) loss between the predicted relevance $\hat{R}^i$ and the reference annotation R. In doing so, we explicitly train the model to learn the weights that capture the relevance between the block transcript and the reference summary.

4 Experiments and Results

4.1 Setup

Dataset Experiments are performed using the How2 dataset (Sanabria et al., 2018), which contains 2000h of instructional videos. More details can be found in Appendix A.1.

Model Hyperparameters Our conformer encoder (Gulati et al., 2020) - transformer (Vaswani et al., 2017) decoder models use ESPNet2 (Watanabe et al., 2018), and computational cost and hyperparameters are discussed in Appendix A.2.

Evaluation Metrics: ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al., 2020) are the most common automatic metrics for summarization.

4.2 Labeling relevance and evaluating labels using R-BASS-Inf

First, we compute relevance based on lexical and semantic similarity and examine whether they are correlated. To do this, we utilize blocks that are 10 seconds long, not too short to not have enough useful information, but not too long so that relevance measures can be fine-grained. We calculate the lexical and semantic relevance per block, creating a vector of binary relevance measures for each recording. Taking the dot product of the two vectors and averaging over all examples in the training data yields an averaged dot product of 0.7, demonstrating that semantic and lexical relevance capture similar information in the data.

Figure 2 shows the binary relevance using averaged lexical and semantic similarity scores of the training data as a function of block index. The first block is the most relevant on average, and relevance decreases as the block index increases. The plot also demonstrates that both semantic and lexical similarity have similar trends across the blocks.

Next, we utilize the obtained relevance labels for the test set to perform block-wise inference while considering only the relevant blocks. Table 1 compares two baseline models trained using FNet self-attentions and BASS with R-BASS models that use the ground-truth labels during inference. *Lex R-BASS-Inf. (GT)* and *Sem R-BASS-Inf(GT)* use relevance labels based on lexical and semantic similarity computed using the reference block transcript and reference summary. Experiments show that though both these approaches obtain the same performance as the BASS baseline, using semantic similarity-based labels leads to greater efficiency improvements, and enables one to drop 63.9% of the blocks on average. When we the more efficient *Sem R-BASS-Inf(GT)* to a corresponding random baseline *Random R-BASS-Inf.* that randomly drops 63.9% of the blocks, we note that the semantic label-based approach outperforms the random baseline in performance, showing the utility of the proposed labeling strategies.

Since these approaches use the relevance pseudo-labels during inference, any blocks that are known to be irrelevant are skipped over at the input, and incur no computational cost. Therefore, this leads to a corresponding speed-up by a factor of 2.77 in inference time on average using semantic similarity-based relevance labels.

4.3 R-BASS with Relevance Loss

Computing the relevance labels using semantic and lexical similarity assumes access to the reference summary and block-level transcripts, which are hard to obtain apriori. Therefore, in this section, we evaluate the R-BASS models trained using the relevance loss to predict block relevance.

R-BASS models can be trained using labels based on semantic similarity (*Sem R-BASS w/ Loss)*

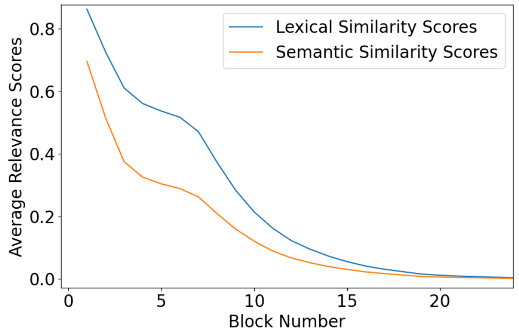

Figure 2: Binary relevance scores averaged over all training samples as a function of block index in audio recordings

Table 1: Performance of R-BASS-Inf and R-BASS w/ Loss using Lexical and Semantic Similarity. ROUGE-L (R-L), METEOR (MTR) and BERTScore(B.Sc.) are reported with the % of dropped blocks (efficiency gain

Updater	R-L↑	MTR↑	B.Sc.↑	% Dropped↑
Baseline-FNet	57.27	29.77	91.62	-
Baseline- BASS	57.98	31.67	91.48	-
Random R-BASS-Inf.	55.76	30.47	90.91	63.90
Lex R-BASS Inf.(GT)	57.96	31.67	91.48	42.25
Sem R-BASS Inf.(GT)	57.96	31.67	91.48	63.90
Lex. R-BASS w/ Loss	57.05	30.91	91.30	69.20
Sem. R-BASS w/ Loss	57.82	31.15	91.42	**86.31**

or lexical similarity (*Lex R-BASS w/ Loss*). The final two rows of Table 1 report the summarization performance and number of blocks dropped based on predicted relevance labels. When using R-BASS with the relevance loss, we find that the number of dropped blocks can be further increased with a small drop in performance. Training with lexical relevance enables the dropping of 69.2 % of blocks and training with semantic relevance enables the dropping of up to 86.3 % of blocks, which is a considerable improvement in efficiency over the baseline BASS approach.

Next, we consider the implications of dropping 86.3 % of blocks on speed-up at inference time. We observe that *R-BASS* with semantic relevance takes on average 7.28 seconds compared to BASS which takes 16.02 seconds, a 2.2x speed-up over BASS in inference time. By dropping 86.3 % of blocks, the expected speed-up may be computed as 7.29 (100/(100-86.3)), which is lower than the observed speed-up. This is because in R-BASS the decision about whether or not to drop an irrelevant block occurs not at the input, but after the relevance prediction by the relevance estimator. Therefore, the expected speed-up is smaller due to additional computation for the dropped blocks including obtaining the encoder output, computing the attention between the encoder output and the previous decoder states, and the projection to produce the single-dimensional relevance prediction.

Qualitative Analysis: We perform qualitative analysis by human inspection to evaluate the impact of R-BASS on summary quality, and find that the R-BASS does not degrade the quality of summaries significantly(see Table 3 in Appendix B). We also compute UniEval (Zhong et al., 2022) scores for coherence, consistency, fluency, and relevance, and find that R-BASS does not significantly degrade quality along these dimensions (see Table 4 in Appendix C).

5 Conclusion

In this paper, we address the challenge of efficiency within blockwise models for speech summarization. First, we introduce a novel model R-BASS that only processes relevant blocks rather than all blocks to produce a summary more efficiently. To realize R-BASS, a relevance estimator is used to predict whether an acoustic block is relevant based on its similarity with the previous block summary.

To obtain labels to train the relevance estimator, we propose to obtain binary relevance labels using lexical and semantic similarity between the block transcript and reference summary. Experiments demonstrate that there exist multiple irrelevant blocks, which can be ignored to improve efficiency while retaining performance.

Finally, we introduce a relevance loss to teach BASS models to predict and use relevance during inference time. Experiments show that training with the proposed semantic similarity loss enables faster processing by dropping around 86% of blocks as irrelevant, resulting in a 2.2x faster inference than BASS while obtaining relatively small performance degradations.

Acknowledgement

We would like to thank Mark Lindsey for their helpful feedback. This work used PSC Bridges2 and NCSA Delta through allocations CIS210014 and IRI120015 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

Societal Impact

We believe that our work will enable the widespread use of technologies that can summarize long recordings into condensed textual descriptions. By making existing streaming models more efficient, our work reduces the carbon footprint of such technologies and enables their use in a more diverse and inclusive set of environments. From serving people with disabilities who find it challenging to process long-form audio content, to improving industrial efficiency in information consumption and decision-making, we believe that our work can positively influence society.

Limitations

In our work, we assume that the first block is always relevant - this assumption is true for the dataset we use, How2, but may not be a general conclusion across all data settings. BASS and consequently R-BASS are approaches that process blocks in sequence, and R-BASS improves time and compute efficiency during training and inference over BASS, however such approaches are likely slower for offline (non-streaming) applications than end-to-end models that use limited context.

Risks

All the work in this paper was done in such a manner so as to minimize the risk of misuse and bias. However, BERT was used to extract semantic embeddings for semantic relevance, and it is possible that such computations carry impacts of the bias in BERT models.

Our models were built using instructional How2 videos mined from YouTube, and our work can enable online speech summarization to help obtain succinct summaries. On the other hand, there may be biases within the data that favor more accurate recognition and understanding of certain kinds of speech. We recommend using our models only for video summarization subject to the license constraints of the How2 dataset.

References

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

Chiori Hori, Sadaoki Furui, Rob Malkin, Hua Yu, and Alex Waibel. 2002. Automatic speech summarization applied to english broadcast news speech. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–9. IEEE.

Jee-Weon Jung, Roshan Sharma, William Chen, Bhiksha Raj, and Shinji Watanabe. 2024. Augsumm: Towards generalizable speech summarization using synthetic labels from large language models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12071–12075.

Takatomo Kano, Atsunori Ogawa, Marc Delcroix, Roshan Sharma, Kohei Matsuura, and Shinji Watanabe. 2023. Speech summarization of long spoken document: Improving memory efficiency of speech/text encoders. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. On Generative Spoken Language Modeling from Raw Audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2019. Read, watch, listen, and summarize: Multi-modal summarization for asynchronous text, image, audio and video. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):996–1009.

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.

Kohei Matsuura, Takanori Ashihara, Takafumi Moriya, Tomohiro Tanaka, Atsunori Ogawa, Marc Delcroix, and Ryo Masumura. 2023. Leveraging large text corpora for end-to-end speech summarization. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Interpretation and transformation for abstracting conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 894–902.

Shruti Palaskar, Jindřich Libovický, Spandana Gella, and Florian Metze. 2019. Multimodal abstractive summarization for how2 videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6587–6596, Florence, Italy. Association for Computational Linguistics.

Shruti Palaskar, Ruslan Salakhutdinov, Alan W. Black, and Florian Metze. 2021. Multimodal Speech Summarization Through Semantic Concept Learning. In *Proc. Interspeech 2021*, pages 791–795.

Dana Rezazadegan, Shlomo Berkovsky, Juan C Quiroz, A Baki Kocaballi, Ying Wang, Liliana Laranjo, and Enrico Coiera. 2020. Automatic speech summarisation: A scoping review. *arXiv preprint arXiv:2008.11897*.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Jean-Pierre Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. *arXiv preprint arXiv:1805.05271*.

Roshan Sharma, Siddhant Arora, Kenneth Zheng, Shinji Watanabe, Rita Singh, and Bhiksha Raj. 2023. BASS: Block-wise Adaptation for Speech Summarization. In *Proc. INTERSPEECH 2023*, pages 1454–1458.

Roshan Sharma, Shruti Palaskar, Alan W Black, and Florian Metze. 2022. End-to-end speech summarization using restricted self-attention. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8072–8076.

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. Curran Associates, Inc.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

A.1 How2 Dataset

The dataset has audio, corresponding transcripts, and a user description that is treated as the reference abstractive summary. The standard split of How2 from (Sharma et al., 2022) is used for all our summarization experiments.

Table 2: Statistics of the How-2 2000h Dataset used for model training and evaluation. The mean and maximum statistics of N- the input length in frames, and L- the output length (in tokens) is shown.

Set	N. Recordings	Max N	Mean N	Mean L	Max L
Train	72,981	145,082	9,806.58	60.54	173
Test	2,127	39,537	9,866.55	60.29	152

A.2 Model Hyperparameters and Computational Cost

Our end-to-end speech summarization models are first pre-trained on the ASR task and then fine-tuned for summarization. The encoder consists of convolutional subsampling by factor 4, followed by 12 conformer (Gulati et al., 2020) blocks with 8 attention heads and hidden size 2048. The decoder has 6 transformer (Vaswani et al., 2017) blocks, with 4 attention heads and hidden size 2048. Models have 103M parameters. Both the encoder and decoder use a dropout rate of 0.2. We use a 43-dimensional filter bank and pitch features as input to the encoder. Summarization models are first pre-trained on ASR using joint CTC-attention (Watanabe et al., 2017) and then fine-tuned for summarization (Sharma et al., 2022).

Our experiments were performed using 4xA40 48GB GPUs - ASR pretraining took 2 days, while BASS and R-BASS fine-tuning took 1.5 days and 0.8 days respectively.

B Qualitative Analysis of Summaries

C UniEval Evaluation of Summaries

Table 3: Qualitative Analysis of R-BASS results

Ground-truth	BASS	R-BASS	R-BASS-Inf
LOOKING FOR NEW LIP COLOR IDEAS ? GET TIPS FOR APPLYING LIP COLOR TO DRAMATIC MAKEUP IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST .	DRAMATIC EYE MAKEUP . GET TIPS FOR APPLYING LIP GLOSS IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST .	ADDING PINKY LIP COLOR CAN BE TRICKY . GET TIPS FOR USING PINKY LIP COLOR IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST .	DRAMATIC EYE MAKEUP . GET TIPS FOR APPLYING LIP GLOSS IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST .
COMBINE OYSTER SAUCE , SHERRY , SESAME OIL AND WATER FOR A SAUCE TO COOK THE CHICKEN IN . MAKE SAUCE FOR CANTONESE CHICKEN WITH GINGER-SCALLION FRIED RICE WITH TIPS FROM A PROFESSIONAL CHEF IN THIS FREE VIDEO ON CULINARY ARTS .	ADD THE CORNSTARCH TO THE CHICKEN MARINADE FOR THE CHICKEN MARINADE . ADD CORNSTARCH FOR GENERAL TSO 'S CHICKEN WITH FRIED MUSHROOM RICE WITH TIPS FROM A PROFESSIONAL CHEF IN THIS FREE VIDEO ON CULINARY ARTS .	THE CHICKEN MARINADE IS A MARINADE FOR THE CHICKEN . MAKE THE CHICKEN STOCK FOR GENERAL TSO 'S CHICKEN WITH FRIED MUSHROOM RICE WITH TIPS FROM A PROFESSIONAL CHEF IN THIS FREE VIDEO ON CULINARY ARTS .	ADD THE CORNSTARCH TO THE CHICKEN MARINADE FOR THE CHICKEN MARINADE . ADD CORNSTARCH FOR GENERAL TSO 'S CHICKEN WITH FRIED MUSHROOM RICE WITH TIPS FROM A PROFESSIONAL CHEF IN THIS FREE VIDEO ON CULINARY ARTS .
INTERESTED IN MAKING STAINED GLASS PROJECTS ? LEARN HOW TO LAY OUT GLASS PIECES ON PATTERNS IN THIS FREE VIDEO ABOUT PREPARING ART GLASS FOR STAINED GLASS CRAFTS .	GLASS CUTTERS NEED TO BE CUT AND THE GLASS . SEE HOW TO CUT GLASS FOR A GLASS CUTTER IN THIS FREE VIDEO .	MAKING STAINED GLASS PATTERNS IS EASY WITH THESE TIPS . GET EXPERT ADVICE ON ARTS AND CRAFTS FOR YOUR GLASS IN THIS FREE VIDEO .	GLASS CUTTERS NEED TO BE CUT AND THE GLASS . SEE HOW TO CUT GLASS FOR A GLASS CUTTER IN THIS FREE VIDEO .

Table 4: UniEval scores of the best performing R-BASS-Inf, BASS and R-BASS w/ Loss models

Unieval dimension	R-BASS Inf.	BASS	R-BASS w/ Loss
Coherence	0.69	0.69	0.67
Consistency	0.70	0.70	0.69
Fluency	0.85	0.85	0.83
Relevance	0.79	0.79	0.77
overall	0.76	0.76	0.74