End-to-End Modeling for Abstractive Speech Summarization

Submitted in partial fulfillment for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

by

Roshan Sharma

B.Tech. Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham
M.S. Electrical and Computer Engineering, Carnegie Mellon University

CARNEGIE MELLON UNIVERSITY
Pittsburgh, PA
March 2024

Acknowledgements

In the words of Nelson Mandela, "The greatest glory in living lies not in never falling, but in rising every time we fall.". I am grateful to have had many pillars of support who helped me rise. Thanks, Prof. Florian Metze for seeing something in me and taking me on as a Ph.D. student in 2019, and for remaining an important mentor since. Thanks, Prof. Ian Lane for stepping in to advise me and teaching me to focus on long-term research. It would not have been possible to complete this thesis without Prof. Bhiksha Raj and Prof. Rita Singh, who have been advisors, mentors, and sounding boards over these past two years, and helped me grow as a researcher and individual. I also owe a debt of gratitude to Prof. Shinji Watanabe, an amazing mentor and inventive collaborator.

I am extremely grateful to my thesis committee comprising Prof. Bhiksha Raj, Prof. Rita Singh, Prof. Shinji Watanabe, and Dr. Florian Metze for all their time, effort, and valuable feedback on my work. I would like to thank Sony, Japan, the United States Army, and the Defense Science Technology Academy (DSTA) in Singapore for their generous support of my research.

Collaboration is truly the currency of today, and I owe a lot to my collaborators for teaching me so much. My thanks to Prof. Alan W. Black, Dr. Suwon Shon from ASAPP, Prof. Karen Livescu from the Toyota Technological Institute at Chicago, Prof. Hung-yi Lee from National Taiwan University, Mr. Daniel Leong and Mr. Jee Wen Jie from DSTA Singapore, and Dr. Takatomo Kano, Dr. Marc Delcroix, and Dr. Atsunori Ogawa from NTT Japan. I am immensely grateful for the amazing internship experiences I had with Meta - thanks to Dr. Suyoun Kim, Dr. Ozlem Kalinli, Dr. Kaustubh Kalgaonkar, Dr. Xin Lei, Dr. Weipeng He, Dr. Ju Lin, and many others.

My thanks to friends in the Sin-bad group and LTI for sage advice and positive conversations - Dr. Shruti Palaskar, Dr. Siddharth Dalmia, Dr. Xinjian Li, Dr. Juncheng (Billy) Li, Tejas, Vikas Raunak, and Dr. Sai Krishna Rallabandi. I am also grateful to friends from the Sense of Wonder group - Mr. Naoyuki Kanda, Mr. Yifan Peng, Mr. Muqiao Yang, Ms. Xuandi Fu, and Dr. Guan-Lin Chao for their companionship. I am grateful to WavLab members for their friendship and collaborative spirit - Mr. Siddhant Arora, Mr. Xuankai Chang, Mr. Yifan Peng, Dr. Soumi Maiti, Dr. Jeeweon Jung, Mr. William Chen, Mr. Jiatong Shi, and many others. I would have been hard-pressed to do all I did without the unrelenting support and unconditional friendship of my family from the Robust Machine Learning for Signal Processing Group - Ms. Hira Dhamyal, Mr. Ahmed Shah, Dr. Raphael Olivier, Dr. Tyler Vuong, Mr. Yangyang Xia, Mr. Mark Lindsey, Mr. Patrick Conrey, Mr. Ankit Shah, Mr. Soham Deshmukh, Mr. Xiang Li, Mr. Muqiao Yang, Dr. Benjamin Elizalde, Dr. Anurag Kumar, Mr. Joseph Konan, Ms. Dareen Alharthi, Mr. Michael Kuhlmann, Mr. Hazim Bukhari, Ms. Massa Baali, and many others. I am grateful to my friends in the ECE department for their constant support and friendship - Mr. Srinivasa Pranav, Mr. Shreyas Chaudhari, Dr. John Shi, Mr. Harry Dong, Mr. Diogo Cardosa Mendez, and many others. I am also grateful to friends from other universities - Dr. Florian Kreyysig, Dr. Desh Raj, Dr. Tiantian Feng, and the soon-to-be doctors – Mr. Zhenyu Wang, Mr. Pavel Denisov, Mrs. Ankita Pasad, Mr. Wangyou Zhang, and many others for the camaraderie we share and for motivating me through your incredible work. Thanks to my dear friends Mr. Varun Haritsa, Mr. Sriram Ganesh, Mrs. Sahana Anbazhagan, and Mr. V Sayee Sunder for remaining connected and constant despite the barriers of distance.

I owe all I am to the hard work, prayers, blessings, and unrelenting support of my family - my dear father whose unwavering faith in me helped scale the tallest heights, my beloved mother whose belief in the transformative power of education and vision for a better life brought me here today, and my brilliant sister and collaborator Ms. Ruchira whose steadfast support helped me power through challenging tasks and unwind. Humble pranams to all my gurus for their guidance and blessings which have helped me be here today.

I would like to dedicate this thesis to the memory of my dear departed grandparents - from humble beginnings in the villages of Tamil Nadu, India to the first doctoral student studying abroad in the family, we have truly come a long way.

Abstract

In our increasingly interconnected world, where speech remains the most intuitive and natural form of communication, spoken language processing systems face a crucial challenge: they must do more than just categorize speech, they need to truly *understand* it to generate meaningful responses. One key aspect of this understanding is speech summarization, where a system condenses the important information from spoken input into a concise summary. This thesis delves into the challenge of generating abstractive textual summaries directly from speech.

The classical approach involves cascade systems that realize speech summarization by first transcribing speech, and then summarizing the resulting transcript. However, this comes with many challenges including computational efficiency, domain mismatches, and error propagation. In this thesis, we propose an alternative—an end-to-end framework that directly optimizes a single sequence model for speech summarization. To implement such end-to-end models with constrained computing resources, we address challenges such as abstract learning, learning global acoustic context, dealing with paucity of data, and improving the quality of summaries using multiple references. We also shed light on observations from human annotation for speech summarization.

We present multi-stage training using speech transcription as a pre-training task to address abstract learning and facilitate improved performance of end-to-end models. We describe multiple solutions to address the problem of global acoustic context—restricted self-attention, replacing self-attention with the Fourier transform, and two block-wise adaptation solutions BASS and R-BASS that reframe speech summarization through the lens of block-wise processing. To address the challenge of data paucity, we introduce work on two new datasets—SLUE-TED and Interview for abstractive speech summarization. An exploration of human annotation provides insights into best practices and the nature of differences between speech-based and transcript-based summaries. Finally, we propose a novel method called AugSumm to improve the diversity and fluency of speech summaries by leveraging auxiliary references from generative text models.

Contents

List of Figures

List of Tables

Abbreviations

Chapter 1

Introduction

Speech is the most widely used form of interaction among humans that enables the exchange of information. Therefore, one of the aspirations of artificial intelligence – the notion that machines can be imbued with human-like characteristics – is the ability to listen to the speech, identify the unique sounds present, and understand the information conveyed. Though modern artificial intelligence that uses deep neural networks has made tremendous progress in attaining near human parity in simpler tasks like clean speech recognition [Xiong et al., 2017], it remains a significant challenge for machines to mimic human understanding.

Truly understanding speech gives humans the ability to extract and summarize useful information conveyed through long audio recordings. This is particularly important because speech in the wild is often verbose, i.e., the "essential" information is contained in a few speech frames rather than in the entire recording. With the rise in the amount of available data that humans consume in daily life- videos, music, podcasts, meetings, lectures, and more, building artificial intelligence that can summarize speech has gained importance. This has driven the development of methods for Automatic Speech Summarization (SSUM), i.e., to automatically generate condensed textual representations called "summaries" from long input audio.

These summaries can be obtained by extracting key frames, i.e., *extractive summarization*, or by generating human-like summaries - *abstractive summarization*. Extractive summaries comprise words or phrases from the speech transcript concatenated to be lexically and grammatically correct. Abstractive summaries, on the other hand, are generated by humans who often paraphrase from the speech, and are thus considered to represent higher-level information. An important distinction we make between speech summarization and other tasks like speech recognition is the notion that summarization requires *global acoustic context*. For speech recognition, local acoustic context is important since the mapping between the input speech and output transcript is direct and monotonic, whereas, for summarization, knowledge of the entire speech signal is required to either extract keyframes or generate abstractive summaries. Extractive summarization can be considered to be less complex than abstractive summarization since the latter involves *abstract learning*. Abstract learning involves learning indirect relationships like the one between

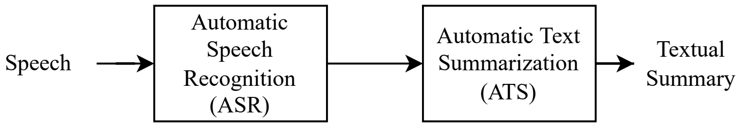

Speech
Automatic Speech Recognition (ASR)
Automatic Text Summarization (ATS)
Textual Summary

FIGURE 1.1: The traditional cascade approach for Speech Summarization uses an Automatic Speech Recognition model to transcribe speech, and an Automatic Text Summarization model to generate the summary from the text.

an abstractive summary and the input speech signal. Abstractive summaries are created by compressing, re-arranging, and paraphrasing the content of the input speech, and hence more challenging to generate.

Traditional approaches to summarization follow a cascade framework- input speech is first transcribed using Automatic Speech Recognition (ASR), and the resulting transcript is used by an Automatic Text Summarization (ATS) model to generate a summary (see Figure 1.1). However, such cascade frameworks suffer from error propagation, i.e., seemingly minor errors in speech transcription get amplified in the summarization model leading to irrelevant, incorrect or incomplete summaries. This is the main challenge we address within this thesis.

1.1 Thesis Overview and Contributions

End-to-End learning, which involves optimizing the cascade jointly with a learning objective to generate more accurate targets has been shown to improve such error cascades previously on other tasks [Gu et al., 2019, Graves and Jaitly, 2014, Bérard et al., 2016, Serdyuk et al., 2018]. Motivated by such approaches, in this thesis, we formulate the task of speech summarization as the end-to-end optimization of a sequence model to produce better summaries directly from speech. To realize end-to-end models in practice, two key challenges need to be resolved - *abstract learning* and *global acoustic context*. Drawing inspiration from curriculum learning, abstract learning can be addressed by an end-to-end speech summarization model that can be first trained to produce the speech transcript, an easier problem, before being fine-tuned to do the more challenging task of producing an abstractive summary from speech. To enable end-to-end speech summarization models to be trained over longer input sequences, restricted self-attention can be used. In training models with these strategies, it is observed that end-to-end models outperform cascade models on abstractive video summarization.

Due to the paucity of labeled data and tools for speech summarization, we introduce two new datasets *SLUE-TED* and *Interview* for generating abstracts from TED Talks and multi-party meeting summaries from NPR interviews. Benchmarking end-to-end and cascade models for multiple speech summarization datasets shows that end-to-end models can outperform cascade models given sufficient amounts of labeled data and strong long-term acoustic context modeling.

Next, we consider the process of human annotation for speech summarization. High-quality annotations are particularly important for building models, and there is no consensus on best practices to annotate summaries. Creating a dataset of human expert and non-expert annotations, it is seen that when annotators read transcripts of spoken content, resulting summaries tend to be more informative, fluent, and coherent. On the other hand, when annotators listen to audio recordings and annotate on that basis, resulting summaries are more information-selective, and have higher inter-annotator agreement. These observations could be useful to develop annotation-aware approaches to speech summarization.

Finally, we describe approaches to improve end-to-end summarization, both in the ability to handle long-term context and produce more diverse summaries.

To improve global context modeling, we introduce *XNORformer*, a new generalized linear transformer that improves performance over existing methods while retaining comparable efficiency. Though improvements to transformer efficiency can help process longer input contexts, such models cannot process arbitrarily long contexts. Therefore, we re-frame the process of end-to-end speech summarization using block-wise processing in a method we call *Blockwise Adaptation for Speech Summarization (BASS)*. Though BASS improves model performance significantly while being able to process arbitrarily long inputs, it also has an increased computational cost over linear transformers. To bridge this gap in efficiency, we introduce a *Relevance-Aware Blockwise Adaptation for Speech Summarization*, which improves the efficiency of BASS, while retaining its performance.

Summaries produced by speech summarization models are limited in lexical diversity and grammatical structure, inhibiting the application of such models *generally*, i.e., across all domains. This results primarily from using one reference summary during training and evaluation, rather than modeling a distribution of valid references. Large language models are leveraged to generate additional summaries, and multiple methods are explored to use these summaries during training and evaluation. Experiments demonstrate that the use of multiple summaries in training can indeed improve the coherence, fluency, factual consistency, and relevance of summaries based on automatic and human measures.

In conclusion, in this thesis, I advance the state of research in speech summarization by proposing end-to-end modeling as a viable alternative to cascade models. I address multiple challenges in making such models feasible to train and infer automatically, including building models that can handle arbitrarily long contexts and produce diverse and general summaries.

1.2 Thesis Statement

Speech contains all of the information present within the transcript, and end-to-end models for speech summarization are a viable, simpler, and more efficient alternative to the classical cascade approach to summarization.

1.3 Thesis Organization

This thesis is organized into three parts – Part I(Background), Part II (Our End-to-End Approach), and Part III Improving End-to-End Modeling.

Part I develops the background and analysis for this thesis. Chapter 2 expounds on the nature of summaries, the task of speech summarization, and how summaries may be evaluated. It also presents an outline of the state of the field until 2022, when work on this thesis begins. Chapter 3 focuses on reviewing prior approaches to speech summarization, and lists their advantages and disadvantages. In Chapter 4, we provide perspectives on the process of human annotation, and whether humans obtain different kinds of information by reading speech transcripts as opposed to listening to audio recordings.

Part II introduces our end-to-end approach for speech summarization, and develops benchmarking efforts that compare end-to-end and cascade approaches. Chapter 5 introduces our formulation of end-to-end speech summarization and presents solutions to abstract learning and global acoustic context for video summarization. Chapter 6 tackles challenges with labeled data by introducing two new datasets, and a multi-domain speech summarization benchmark.

Part III proposes multiple approaches to improve the performance of end-to-end speech summarization models. Chapter 7 introduces the XNORformer for efficient and performant global acoustic context. It also introduces BASS and its more efficient variant R-BASS which are block-wise methods to summarize arbitrarily long recordings. Chapter 8 examines the challenge of unitary references for speech summarization that leads to less lexically diverse summaries and introduces AugSumm, an approach to obtain and use multiple references during training and evaluation.

Finally, Chapter 9 summarizes the main conclusions of this thesis and proposes directions for future research in the area.

Part I

Background: Speech Summarization

Chapter 2

Speech Summarization

The goal of audio summarization is to generate a *summary* – a text, audio, or video snippet that relates most or all of the key information in the recording.

Summaries may represent different kinds of key information within the audio - for example, content information in speech, acoustic events from the scene, speaker information from speech, or prosody information in music. Audio summaries can be a combination of all of these different attributes. In speech-centric audio, i.e., where one or multiple people are communicating through speech, it is useful to focus on summarizing content information embedded in speech, which we term speech summarization.

There may be similarities between speech summarization and other tasks like speech recognition where spoken content is transcribed as text, or audio event detection, where acoustic events are identified and classified. However, the primary distinction between summarization and all other tasks is that the summary is always a "condensed" representation of the input, and does not contain all the information in the input.

Condensed content summaries can take multiple forms - audio clips, images, videos, lists or sequences of events, text, or more complex forms including combinations of the aforementioned. The choice of representation for a summary is application-dependent. For applications like key-term detection, summaries could simply be a list of key terms, while for YouTube news, summaries could simply be a salient video frame or short video clip combining salient portions. In general, video summaries are often more engaging to users, while audio summaries enable hands-free communication and are useful for users with reading disabilities. Textual summaries may be preferred in many cases owing to the reduced cognitive load of skimming text as opposed to listening to a recording or watching a video. Though each output modality has its advantages and disadvantages, in this thesis, we focus on the task of producing textual output summaries.

2.1 Summaries: Categories, Application and Characteristics

There exist multiple categorizations of textual summaries based on the goal that underlies summarization. Summaries can be *abstractive* if the goal is to present information from the input like how humans may summarize, or *extractive* by being composed of key phrases or sentences from the input. *Indicative* summaries are constructed as guides for where to find the most important parts of the input, while *informative* summaries are meant to convey the most important information within the input. *Descriptive* summaries capture the tone of the input while containing figurative language. Depending on whether any additional constraints are provided for summarization, a distinction can be made between *generic* summarization and *query-based summarization*. Of these, the former has no constraints on the summary, while in the latter, the summary is required to be pertinent to a given query. In this thesis, we focus our effort on the challenging task of *abstractive generic* speech summarization.

Speech summarization has many practical applications. Humans consume a myriad of audio multimedia in their daily lives, ranging from listening to the morning news, music, and podcasts, to participating in meetings and lectures. A characteristic of all these interactions is that the audio in most cases is very verbose and that the essential information is concentrated within a few select frames as opposed to within the entire recording. Therefore, such long recordings can be condensed into summaries comprising only the essential information. News broadcasts can result in news headlines and news reports, lectures, and meetings can be converted into notes, and talks can be summarized into succinct titles and abstracts. Apart from the utility of these summaries for human consumption, summaries can be used as additional inputs to other downstream tasks. For example, a summary can be used for keyword extraction, retrieval, topic detection, and also as input for question answering or natural language inference.

The application of summarization decides the extent and type of "essential information" that summaries ought to focus on, and the desired characteristics of summaries. Summaries can be used to index and retrieve multimedia based on highly specific queries. For example, summarization can be used to extract keywords like "Spanish rice" or "black beans" from videos. Based on such keywords, the search query "look up instructional videos on making Spanish rice with black beans" yields a few relevant videos describing the process of making Spanish rice with black beans as opposed to a large list of videos that explain how to make brown rice. For such instructional videos, summaries are often short and factual and comprise keywords that can be used for indexing. Therefore, any missing or incorrect content words in the summary would result in poor retrieval. Abstractive summaries can also be used to encourage viewers to consume multimedia such as TED talks, and podcasts. The emphasis in such summaries is to build logical, cogent, and persuasive arguments, so incorrect grammar and incoherent summaries may dissuade users from perusing the videos or talks. Meetings can be summarized using extractive and abstractive summaries to convey progress updates, and discussion outcomes and demarcate the next steps for meeting participants. In such cases, missing, inaccurate, or misleading information, especially in entity names, pronouns, and co-references may pose significant challenges to utility.

2.2 Automatic Speech Summarization (SSUM)

Automatic Speech Summarization (SSUM) aims to automatically generate summaries given speech recordings. It is desirable to develop automatic methods for speech summarization for multiple reasons:

1. Improved Information Accessibility: Automatic methods to summarize long recordings save time and effort in consuming long-form content such as speeches and lectures. Further, such techniques can make audio content accessible in summary form to people with hearing impairments or those who prefer to read text. It can also enable efficient indexing and retrieval of important information within long-form audio. Widespread access to such technology can enable open and easy access to information across the world, bridging barriers of language through associated technologies like automatic translation.
2. Increased Productivity: By producing detailed automatic summaries with relatively low annotator bias for long-form audio, the productivity of humans and entities can improve by enabling focus on other tasks.

To mathematically define the task of automatic speech summarization, let $X = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots \mathbf{x}_N]$ represent a sequence of speech feature vectors. Let Y = $[y_1, y_2, y_3 \cdots y_M]$ represent the sequence of text tokens representing the speech summary. The goal of a parametric automatic speech summarization method is to estimate the optimal parameters Θ_{SSUMM} that maximize the likelihood of predicting the correct summary given the sequence of speech feature vectors X. This is shown in Equation 2.1:

$$\Theta_{\text{SSUMM}} = \text{argmax}_{\Theta} P(Y|X;\Theta), \tag{2.1}$$

where $P(Y|X;\Theta)$ refers to the conditional probability of Y given X. The θ after the semicolon indicates that this probability is being characaterized as a parameterized function with parameters θ (Here, and in the rest of this document, we will continue to use the notation $P(.)$ to represent probabilities, and $P(.;.)$ to represent parameterized models for probabilities, where the parameters are represented by the symbol following the semicolon).

Automatic Speech Summarization can be addressed using different approaches that we discuss in detail in Chapter 3 and Chapter 5.

2.3 Evaluating goodness of abstractive summaries

The quality of summaries can be assessed using *intrinsic* evaluation, where summaries are compared against a reference or gold standard, or *extrinsic* evaluation, where resulting summaries are used in downstream tasks, and their performance in such tasks is used as the evaluation metric.

We primarily focus on intrinsic evaluation within this thesis, since the focus is open-domain general summarization as opposed to focusing on particular downstream applications.

The goal of abstractive summarization is to create human-like summaries, and from this perspective, the best possible evaluation for abstractive summaries is human opinion. Human opinion is solicited on primarily four dimensions - coherence, consistency, factualness, and fluency. Coherence refers to whether all sentences within a summary are organized logically, i.e., whether every sentence or paragraph builds on previous ones in a smooth progression. Consistency refers to the logical alignment of information within a summary and its factual adherence to the source. Fluency refers to the ease and naturalness of reading the summary and evaluates phrasing, grammatical errors, and sentence structure. Relevance refers to the extent to which the summary aligns with the specific needs of summarization, a proxy for which is a reference summary. These four dimensions can collectively be used to evaluate speech summaries. However, human evaluation at scale is challenging owing to the amount of time, effort, and resources involved.

Therefore, automatic metrics for intrinsic evaluation are popular within the community. Building automatic metrics for abstractive summary evaluation is a particularly challenging problem. There is no consensus on the best metric for speech summarization. However, the most popular metric for text and speech summarization currently includes Recall Oriented Understudy for Gisting Evaluation (ROUGE) and its variants. ROUGE [Lin, 2004a] is based on the machine translation metric BLEU [Papineni et al., 2002], but focuses more on recall. Essentially ROUGE and BLEU compare the extent of n-gram overlap between a hypothesis and reference summary(ies). ROUGE-L measures lexical overlap by using the longest common subsequence between the hypothesis and reference(s).

Let L_{ref} be the number of words in the reference summary, and L_{hyp} be the number of words in the hypothesis summary. Then the longest common subsequence between the two can be computed, and let its length be L_{lcs}. Then ROUGE-L can be computed as shown in Equation 2.2.

$$\text{ROUGE-L} = \frac{2L_{lcs}}{L_{ref} + L_{hyp}} \tag{2.2}$$

ROUGE-L was shown to correlate well with human judgment from the Document Understanding Conference (DUC) data and hence is widely used.

ROUGE was extended to the Basic Elements Evaluation Suite [Hovy et al., 2006] which uses Basic Elements as units instead of n-grams. Basic Elements are computed using semantic analysis as heads of major syntactic constituents or relations between heads and dependents. As this metric relies on parsing and pruning, it may not be suitable for disfluent and conversational speech.

An alternative to these is the Pyramid method [Nenkova et al., 2007, Nenkova and Passonneau, 2004] that uses variable length sub-sentential units called semantic content units (SCUs) to compare hypothesis to reference. However, obtaining SCUs requires manually annotating units for meaning, which is challenging to do in practice.

METEOR [Banerjee and Lavie, 2005a] was proposed to capture semantic similarity beyond literal n-gram overlap, and combines n-gram matches with stemming and synonym matching to accomplish this. However, it struggles with out-of-vocabulary words and prioritizes synonyms over paraphrases. Despite these limitations, METEOR remains a metric in use.

Apart from these metrics that rely on n-gram overlap, the advent of deep learning advanced the notion of model-based metrics. Mover's distance was proposed by [Mikolov et al., 2013], who used word embeddings to calculate the semantic distance between the hypothesis and reference summaries. Though this metric captures semantic similarity more effectively compared to n-gram-based metrics, evaluation depends heavily on the chosen word embeddings, making result interpretation challenging. Improving on this notion of model-based embeddings, pre-trained language models began to be employed to generate such embeddings. For example, MoverScore [Zhao et al., 2019] uses contextualized embeddings along with Earth Mover distance to estimate the quality of summaries. However, this metric considers word frequency and distribution, which results in an underestimation of similarity between texts with significant lexical differences.

BERTScore [Zhang et al., 2019] is a popular metric that relies on a greedy alignment-based aggregation of token similarity scores between the hypothesis and reference. More concretely, given a reference and hypothesis summary, they are both tokenized to obtain token embeddings from a BERT [Devlin et al., 2019a] model. RoBERTa [Liu et al., 2020] is often used as the embedding extractor. Then, pairwise cosine similarity is computed between the tokens of the hypothesis and reference summaries. Based on this pairwise measure, a greedy alignment is obtained to map the tokens of the hypothesis to the most similar tokens in the reference. Then, the cosine similarity scores of these aligned tokens are aggregated to obtain the BERTScore. SPEEDScore [Akula and Garibay, 2022] is an extension that uses sentence-level embeddings from a sentence transformer as opposed to token-level embeddings.

More recent work has focused on multiple dimensions of evaluation to assess summary quality. BLEURT [Sellam et al., 2020] combines the evaluation of semantic similarity and fluency by using pre-trained language model embeddings to extract semantic similarity scores and a fluency scorer to check the grammatical correctness and naturalness of the summary. BARTScore [Yuan et al., 2021] uses the BART-large model to evaluate fluency, relevance, and informativeness using F1, precision, and recall scores. To align these scores to different dimensions of human evaluation, UniEval [Zhong et al., 2022b] introduces a unified multi-dimensional evaluation that produces scores for relevance, coherence, consistency, and fluency. Experimental results show that UniEval improves correlation with human judgments, making it a strong multi-dimensional evaluator.

UniEval reframes multi-dimensional evaluation of these attributes as multiple binary classification tasks where the model responds to questions like "Is this a coherent summary of the input document" with "yes" or "no". The probability of the model predicting "yes" given the hypothesis summary, reference summary, and the summary source is considered as the UniEval score for that dimension. Of the four dimensions, fluency and coherence are computed with no reference, i.e., using only the hypothesis summary. Relevance is computed using the hypothesis and reference summaries, while consistency is computed between the hypothesis summary and the summary source.

Finally, there is no single best reference summary for a given source. Given the same input, human annotators often exhibit low inter-annotator agreement in the semantic units they select, and how they combine these to form abstractive summaries. In other words, there may be multiple possible valid summaries given a source document, which cannot all be enumerated. This is addressed in practice by using as many references as possible or by using reference-free evaluation.

2.4 Speech Summarization through the decades

Work in speech summarization followed decades of prior work in text summarization and gained prominence in the 1990s. Advancements in automatic speech recognition motivated research into automatically summarizing speech.

[Chen and Withgott, 1992] attempted to identify areas of emphasis within speech for summarization using F0 and energy features. Hidden Markov Models were used to select informative portions of speech in recorded interviews and telephone speech, resulting in near-human performance.

Early approaches examined summaries as abstracts and labeled them as indicative, informative [Borko and Bernier, 1975], or critical evaluative abstracts [Anderson, 1992]. At that time, summarization focused mostly on extractive applications [Mani, 1999, Mani, 2001].

Through the early 2000s, automatic extractive summarization improved and expanded, resulting in their deployment for summarizing voice mails [Koumpis and Renals, 2005], and broadcast news [Hori et al., 2002, Christensen et al., 2004]. Some approaches used lexical features [Hori et al., 2002, Christensen et al., 2004], while others used acoustic features, [Maskey and Hirschberg, 2006, Inoue et al., 2004]. Such approaches were also extended to summarize lectures [Zhang et al., 2007].

Summarization approaches used different modeling strategies including Support Vector Machines (SVMs) [Lin et al., 2009b], skip-chain Conditional Random Fields (CRFs) [Galley, 2006], Hidden Markov Models (HMMs) [Zhang et al., 2009], graph models [Lin et al., 2009a], or unsupervised approaches like Maximal Marginal Relevance [Carbonell and Goldstein, 1998] or Latent Semantic Analysis [Gong and Liu, 2001].

As extractive summarization improved, many investigated alternative forms of summarization such as abstractive summarization [Murray et al., 2010]. Research into the appropriate acoustic, lexical, prosody, and structural features continued [Zhu and Penn, 2006].

With the rise of deep learning, and sequence model architectures [Graves, 2012] for tasks like machine translation [Graves and Jaitly, 2014] and speech recognition [Chan et al., 2016], sequence models were built for abstractive text summarization [Nallapati et al., 2016b]. Most of the work on text summarization has focused on single-document summarization for domains such as news [Rush et al., 2015,Nallapati et al., 2016a,See et al., 2017,Narayan et al., 2018] and some on multi-document summarization [Woodsend and Lapata, 2012,Cao et al., 2015, Yasunaga et al., 2017] .

Recent work on abstractive summarization has used cascade approaches to first perform speech recognition on individual utterances. The resulting transcripts for these utterances are then concatenated and used to generate abstractive summaries. An example of this is found for multimodal video summarization [Palaskar et al., 2019a,Li et al., 2019], where speech transcripts in conjunction with visual representations are used for abstractive summarization. [Shang et al., 2018] describes an unsupervised approach for meeting summarization, where ASR transcriptions are pre-processed, and related utterances are identified through unsupervised clustering. These resulting "communities" are used to derive abstractive sentences and finally abstractive summaries. [Kano et al., 2021a] attempts to address the challenge of incorrect transcriptions by combining multiple hypotheses using attention, and experimental results demonstrate the efficacy of the approach.

Chapter 3

Background: Approaches to Speech Summarization

Complex tasks like Automatic Speech Summarization (SSUM) can be performed by using a cascade of multiple modules. The first module takes as input the speech recording and produces an output. The output of the first module is used as input to the second module, the output of the second module is used as input to the third module, and so on. The final module produces the desired speech summary.

Cascades can be designed in multiple ways to address the problem of speech summarization. In this chapter, we delve deeper into the traditional approach to speech summarization, i.e., the cascade approach. We examine the most common cascade formulation comprising speech recognition and text summarization and expound on the design of these modules. Then, we discuss alternative designs for cascades and compare them against the standard approach. We conclude by discussing the advantages and disadvantages of cascade models for speech summarization.

3.1 Cascade Formulation of Speech Summarization

Cascade approaches to speech summarization typically comprise two modules – speech recognition and text summarization. However, there are alternative formulations that may utilize acoustic features like prosody and pitch as additional inputs to the summarization module. [Palaskar et al., 2020] describes a three-module cascade framework for speech summarization where speech is first converted into its verbatim transcript. The transcript is then used to extract important noun phrases and verb phrases that represent the important content. These phrases, called concepts are then used to generate abstractive summaries.

Now, we describe the components of the popular two-module cascade models for summarization-speech recognition and text summarization.

3.2 Speech Recognition

The problem of automatic speech recognition is generally posed as maximum *a posteriori* estimation: finding the most likely token sequence T for given an input audio recording X. This requires knowledge $P(T|X)$, or alternately $P(T, X)$. These can generally not be known, and are modelled instead by parametric models, the parameters of which are learned from data.

The most successful current approaches, neural-network-based end-to-end systems [Chan et al., 2016] represent the posterior probability as neural networks that compute $P(T|X;\theta)$, where θ are the parameters of the network and are learned through the optimization

$$\hat{\theta} = argmax_{\theta} E_{\sim(X,T)} \log P(T|X, \theta), \tag{3.1}$$

where the statistical expectation $E_{\sim(X,T)}$ is replaced by an empirical average computed over large training sets of samples of (X, T) that are ideally drawn from the *true* distribution $P(T, X)$. $\hat{\theta}$ represents the parameters learned by the training process.

Inference, *i.e.* recognition, attempts to find the token sequence $\hat{T}$ with the maximum probability:

$$\hat{T} = argmax_{T} P(T|X, \hat{\theta}) \tag{3.2}$$

3.2.1 Speech Recognition Approaches

The classical speech recognition approach is the *hybrid* approach that divides the challenge of speech recognition into subproblems that are addressed by independently optimized modules like the acoustic model, language model, and lexicon model, etc. Earlier realizations used Hidden Markov Models (HMMs) to represent speech as a sequence of hidden states that could emit tokens with an output probability modeled by Gaussian Mixture Models (GMMs). Advances in deep learning helped replace the GMM with a Deep Neural Network (DNN) that has higher representation power and yielded better performance. Such hybrid systems obtained until recently very competitive performance within production systems. However, the primary challenges behind their deployment include the complexity of training multiple components and the possibility of a mismatch between the training of these components that degrades performance.

Around a decade ago, with the rise of sequence model architectures, *end-to-end* approaches were introduced that were much simpler than hybrid models and could achieve better performance due to end-to-end optimization. End-to-end models use recurrent or transformer-based architectures today and can be realized in three different ways: using Connectionist Temporal Classification (CTC), attention-based modeling, and using Transducers. Of these, the first two have been employed within this thesis, and we delve deeper into these methods.

Connectionist Temporal Classification (CTC): The core idea behind CTC is to estimate the probability of predicting the transcript tokens conditioned on the input speech frames such that every frame is mapped to a transcript token or a special “blank“ token. To realize this, transcript prediction is performed in two steps: (a) identifying all intermediate label representations π with frame alignments that lead to valid transcripts, (b) estimating the probability of each intermediate label representation given the speech input sequence X, and aggregating the probability values over all possible alignments.

Equation 3.3 describes this mathematically, where t' represents transcript label sequences with blanks inserted.

$$P(T|X) = \sum_{\pi \in \Phi(t')} P(\pi|X) \tag{3.3}$$

Within a sequence model, $P(\pi|X)$ is represented as a product of conditionally independent predictions across timesteps $l = 1, 2, \cdots L$. Equation 3.4 shows this process, where the function $f()$ represents the output activation of the sequence model at the l-th timestep, i.e., the probability of the l-th label in π.

$$P(\pi|X) = \prod_{l=1}^{L} f(\pi_l) \tag{3.4}$$

Attention-based Encoder-Decoder (AED):

A sequence model is comprised of an encoder and decoder, where the encoder is used to transform the input speech features into a high-level hidden representation. The decoder uses attention to identify which hidden representations are useful to autoregressively transcribe the output.

Given the D-dimensional speech feature sequence of an utterance X= $\{x_i|x_i \in \mathbb{R}^D, i = 1, 2...N\}$, the encoder produces an F-dimensional high-level hidden representation $H = \{h_j|h_j \in \mathbb{R}^F, j = 1, 2, ...M\}$. At each time step, the decoder computes an attention weight and an encoder context vector. Using the encoder context and the previous decoder states, the decoder produces an output token y_k from the output vocabulary $\mathbb{V}$ at every time step. Collecting y_k across the O decoding timesteps yields the output transcript $Y = \{y_k|y_k \in \mathbb{V}|l = 1, 2, ...O\}$.

$$\mathbb{L}_{att} = -log(P_{att}(Y|X)) \tag{3.5}$$

$$\mathbb{P}_{att}(Y|X) = \prod_{l=1}^{O} P_{att}(y_l|X, y_1, ...y_{l-1}) \tag{3.6}$$

$$H = \texttt{Encoder}(X) \tag{3.7}$$

$$P_{att}(y_l|X, ..y_{l-1}) = \texttt{AttentionDecoder}(H, y_{l-1}) \tag{3.8}$$

3.2.2 Model Architectures

End-to-end speech recognition models can be realized using recurrent [Chen et al., 2015, Chorowski et al., 2015, Watanabe et al., 2017] or transformer [Vaswani et al., 2017a] architectures. Of these, the transformer architecture is preferred due to its computational efficiency, speed, and performance.

The conformer [Gulati et al., 2020b] was introduced as a variant of the transformer for speech recognition and is commonly used within end-to-end models.

The Conformer processes the input speech sequence X using a convolution neural network (CNN) [Krizhevsky et al., 2012] as follows:

$$X' = \text{LN}(\text{ReLU}(\text{CNN}(\text{ReLU}(\text{CNN}(X))))), \tag{3.9}$$

$$\tag{3.10}$$

Here ReLU is an activation function, layer normalization LN is a linear mapping function [Ba et al., 2016], and MHAtt represents multi-head self-attention [Vaswani et al., 2017a].

The rest of the processing is similar to that of the Transformer, except that the input of $\mathrm{MHAtt}(\cdot)$ is processed with the feed-forward layer, and then the output of the $\mathrm{MHAtt}(\cdot)$ is processed with a convolutional layer before being fed to the feed-forward layer.

3.3 Text Summarization

Text Summarization can be formulated in a similar manner to speech recognition. Given the input transcript T and the reference summary Y, the goal of training is to estimate the model parameters θ_2 that maximize the posterior $P(Y|T;\theta_2)$.

$$\hat{\theta_2} = argmax_{\theta_1} E_{\sim(Y,T)} \log P(Y|T,\theta_2), \tag{3.11}$$

$$\hat{Y} = argmax_Y P(Y|T,\hat{\theta_2}) \tag{3.12}$$

Equation 3.12 represents the inference procedure for summarization, where the goal is to identify the most likely summary token sequence given the transcript sequence T and the trained summarization model with parameters $\hat{\theta_2}$. $\hat{\theta_2}$ represents the parameters learned by the training process given parameters θ_2.

Text summarization models today are implemented as encoder-decoder sequence models or decoder-only autoregressive predictors.

Within the encoder-decoder setup [Lewis, 2020, Raffel et al., 2020a], the encoder consumes text tokens and produces a high-level latent representation of these. The decoder employs cross-attention to focus on these latent representations and autoregressively produces tokens of the summary.

Decoder-only non-causal models [Liu* et al., 2018] today primarily uses the prefix language model formulation where prefix tokens that represent the prompt use non-causal bidirectional attention, while the summaries use causal attention in prediction.

Recent work in text summarization focuses on using large-scale pre-trained language models or large language models for text summarization. Popular models include BART [Lewis et al., 2019], T5 [Raffel et al., 2020b], Flan-T5 [Chung et al., 2022], and LLama2 [Ouyang et al., 2022].

3.4 Inference within the Cascade

During training, we independently optimize the constituent components, i.e., speech recognition and text summarization.

$$\text{argmax}_Y P(Y|X;\hat{\theta}_1,\hat{\theta}_2) = \text{argmax}_Y \sum_T P(T,Y|X;\hat{\theta}_1,\hat{\theta}_2) = \text{argmax}_Y \sum_T P(Y|T;\hat{\theta}_2)P(T|X;\hat{\theta}_1) \tag{3.13}$$

During inference, we need to obtain the most likely summary token Y given the input speech X and the trained models with parameters $\hat{\theta}_1$ and $\hat{\theta}_2$. Equation 3.13 shows how this process can be broken down between the two modules — speech recognition and text summarization.

$$\text{argmax}_Y \sum_T P(Y|T;\hat{\theta}_2)P(T|X;\hat{\theta}_1) = \text{argmax}_Y P(Y|T;\hat{\theta}_2)\Big[\text{argmax}_T P(T|X;\hat{\theta}_1)\Big] \tag{3.14}$$

However, while performing inference on independently trained models, we first perform inference over the speech recognizer to obtain the most likely transcript token sequence T, and then use this as input to the summarization model to retrieve the most likely summary sequence Y. This means that rather than obtaining the quantity shown in Equation 3.13, the cascade formulation instead retrieves the output of Equation 3.14.

This results in sub-optimal outputs, which can be mitigated in part by considering multiple hypotheses from the ASR module for summarization [Kano et al., 2021a]. In the next chapter, we will see that our end-to-end formulation models the quantity in Equation 3.13 rather than the one in Equation 3.14, leading to more accurate models.

3.5 Advantages and Disadvantages of the Cascade Approach

The commonly used cascade approach has some advantages and disadvantages. The most important advantage of the cascade approach lies in its *modularity* — with the speech transcript being an intermediate output, the model's performance on summarization can be assessed in terms of how accurately the intermediate transcript was produced. Further, to obtain diverse outputs from summarization, this intermediate transcript can be controlled and manipulated.

Another advantage of the cascade system is the ability to train the constituent modules independently of each other with different corpora. This allows one to train speech summarization systems despite the lack of parallel speech and summary data, which is particularly important given the relative dearth of annotated text corpora. Further, today, there exist readily available pre-trained models for both speech recognition and text summarization that one may leverage to build cascade systems easily.

With the ability to train the constituent models on different corpora, there may arise some disadvantages. The two models may be trained on data from different domains, leading to challenges in generalizability to the domain that the other module was trained on. For example, if the speech recognition model were trained on biology lectures, and the summarization model was trained on news summarization, the model may not be able to summarize biology lectures accurately due to the domain mismatch between these two models. Further, most available pre-trained summarization models do not generalize beyond their intended purpose. For example, one cannot obtain the title of a news article from a summarization model trained to generate longer news reports. This limits the utility of such pre-trained models.

Error propagation is a well-known challenge within cascade formulations. In this context, it refers to the fact that errors made by the first module, i.e., speech recognition transfer over to the second module text summarization, which results in vastly incorrect, irrelevant, or incoherent summaries. Error propagation effects magnify with each subsequent module that erroneous intermediates propagate to, i.e., cascades with more modules would have more severe impacts from error propagation.

Another challenge within cascade formulations centers around the effective number of model parameters. The required number of model parameters within the cascade can be obtained by summing the number of model parameters within every individual module. Often, due to the additive nature, the resulting number of model parameters for a cascade is very large. This inhibits the deployment of such models for on-device and other low-compute resource applications. Further, the independent training and a large number of model parameters have adverse implications on the amount of time required to train such models.

Chapter 4

Insights gained from human summarization

In this chapter, we consider the process of how humans summarize speech. How humans summarize speech potentially informs the quality of summaries used as human references to develop automatic methods. It also provides insights into what kinds of information humans can capture within summaries of speech, and such information can guide the development and deployment of improved automatic methods for human summarization.

The process of summarization can be approached in two ways by human annotators — by listening to the audio and writing a summary based on it, and by reading a textual transcript and writing a summary based on it. We ask whether there exists any difference in summaries that result from these two strategies. Such labeling information is not present within current datasets, and hence, we introduce a new dataset obtained by collecting expert and non-expert annotations. Annotators summarize either by listening to audio recordings or by reading the transcripts of audio recordings in such a manner that no annotator uses both these methods on the same recording.

Based on our data and analysis, we attempt to comment on best practices for human annotation of speech summaries. Our analysis reveals that the summaries based on reading the transcript and listening to the audio are different in informativeness and factual consistency.

We also examine whether expertise in annotation impacts the quality of the summary, and find that expert annotations tend to be more informative and reliable.

4.1 Overview and Research Questions

Fig. 4.1 presents an overview of our work. Humans can summarize speech when it is presented by (a) listening to the speech directly and then constructing the summary based on this, or (b)

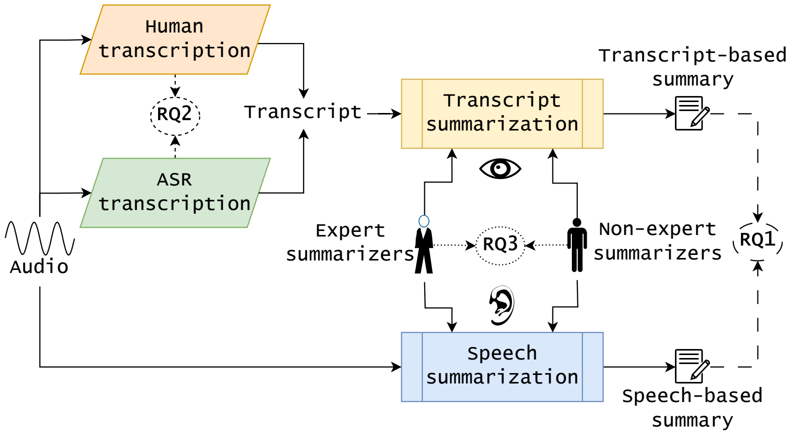


FIGURE 4.1: Overview of our proposed work for examining the ways that humans can summarize speech. Humans can summarize speech by listening to audio or reading speech transcripts, which forms Research Question 1. Differences in the generation of speech transcripts could lead to different summaries, which we address in Research Question 2. Finally, expertise of the summarizer may impact the summaries, which we study in Research Question 3.

reading a textual transcript of the content spoken in the recording, and using that to construct the summary. It is unclear whether summaries based on listening to audio differ from summaries based on reading transcripts. For transcript-based summarization, the method used to obtain the transcription, either Automatic Speech Recognition (ASR) or manual transcription, may also have a significant effect. Additionally, the quality of a summary, regardless of the source modality, is likely to be determined by the level of expertise of the summarizer. To assess the impact of source modality, transcription, and expertise, we formulate the following research questions:

(RQ1) Are summaries based on annotators listening to speech recordings different from summaries based on annotators reading textual transcripts of speech? If so, how?

(RQ2) If ASR transcripts are used instead of manual transcripts, how do recognition errors impact human summaries?

(RQ3) Do summaries written by non-experts summaries differ from those written by experts, and if so, do the differences vary based on the source modality?

Past work in automatic speech summarization suggests that there are acoustic characteristics in speech beyond the textual context that are useful for speech summarization [Kano et al., 2023a]. This implies that both transcript and raw speech carry complementary information that is useful for summarization. The fact that end-to-end models [Sharma et al., 2022a, Kano et al., 2023a, Matsuura et al., 2023b, Matsuura et al., 2023a, Sharma et al., 2023b] that use speech features as input perform differently from cascade models [Kano et al., 2021b, Palaskar et al., 2021a, Yu et al., 2021, Palaskar et al., 2019b] that use speech transcripts [Sharma et al., 2023a] suggests that there are differences in automatic summaries resulting from textual or speech input. In this chapter, we study whether the same holds for human annotators.

However, any observations on the impact of reading versus listening for annotation are difficult to make from existing corpora like How2 [Sanabria et al., 2018], SLUE-TED [Shon et al., 2023], Interview [Sharma et al., 2023a], and AMI [McCowan et al., 2005], since it is unclear how human annotations were obtained from speech. Therefore, we select 1002 recordings from the *Interview* [Sharma et al., 2023a] corpus test set and create a new evaluation set. We collect two expert human annotations for transcript-based summarization and two for speech-based summarization per recording and assess how the resulting summaries differ.

Comparing two given summaries is challenging. Broadly, we conduct three types of evaluation: (a) comparing the summary to its source transcript (*source-based evaluation*) to establish how extractive and factual the summary is, (b) evaluating the summary independently for its structure and grammar (*structure evaluation*), and (c) by comparing different summaries (*summary comparisons*) to assess relevance and which represents the input better using semantic similarity, LLM, and human evaluation. Experiments show that speech-based summaries are more information-selective and factually consistent while transcript-based summaries are preferred by human and LLM evaluators.

In real-world settings, human-annotated manual transcripts are not available, and, therefore, Automatic Speech Recognition (ASR) models may be used to generate such transcriptions. ASR transcripts may contain errors, and we investigate the impact of these errors on transcript-based summarization. To do this, we consider 100 recordings from the Interview test set, obtain speech transcriptions using Whisper [Radford et al., 2022], and perform human summarization using 2 expert annotators for each modality. Experiments show that ASR errors in the transcript degrade the quality of human summaries, resulting in lower coherence, fluency, and factual consistency with the source.

Prior work in human evaluation of abstractive text summarization highlights the importance of expertise for summary evaluations [Fabbri et al., 2021b, Gillick and Liu, 2010]. However, to the best of our knowledge, there is no prior work that examines the need for expertise in human speech summarization. Therefore, in this work, we examine whether crowd-sourced non-expert summaries are different from expert summaries. To address this question, we obtain 2 non-expert annotations using Amazon Mechanical Turk (AMT) for each of the 1002 recordings from the *Interview* test set. From these summaries, we analyze (a) differences between expert and non-expert annotations, and (b) whether any differences can be attributed to the approach taken for summary annotation. Analysis shows that non-expert annotations are less fluent and coherent compared to expert summaries but as factually consistent as expert summaries.

In summary, through this work, we find that transcript-based annotation is valuable if errors are minimal and longer, more informative summaries are desired, while speech-based annotation is desired for higher information selectivity, factual consistency, and resilience to transcription errors.

4.2 Related Work

For this study, we describe related work in three areas: (a) metrics for comparing summaries, (b) other work investigating the impact of source modality on human performance, and (c) investigations of ASR error impact on Spoken Language Understanding.

Metrics for comparing summaries: Numerous metrics have been proposed to evaluate the quality of summarization [Fabbri et al., 2021a]. Earlier works use metrics like ROUGE-H, ROUGE-L [Zhou et al., 2006], ROUGE-WE [Ng and Abrecht, 2015], BLEU [Papineni et al., 2002], sentence recall/precision is commonly used for simple sentence based extraction summaries [Hirohata et al., 2005]. Some studies have found that ROUGE and BLEU-based metrics correlate well with human judgments on summarization. However such scores are not always the best metric for summarization, especially when the input is longer like in meeting summarization and summarization of scientific articles. There have been scores of other metrics which are either variants or combinations of the earlier mentioned metrics like BertScore [Zhang et al., 2019], Sentence Mover's Similarity (SMS) [Clark et al., 2019], SummaQA [Scialom et al., 2019], SUPERT [Gao et al., 2020], CHRF [Popović, 2015], METEOR [Banerjee and Lavie,], CIDEr [Vedantam et al., 2015], or model generated ones like s3 [Peyrard et al., 2017], BLANC [Vasilyev et al., 2020], summACCY [Hori et al., 2003], which are trained to predict the quality of an input summary.

LLMs have been used as an alternative method for summary evaluation. While some work expresses doubt that LLMs have reached a human level of summary evaluation capabilities [Shen et al., 2023], other methods have yielded more optimistic results [Wu et al., 2023]. LLM evaluation methods usually take the form of automatic replacements for human evaluations such as Likert scale scoring, pairwise comparison, and binary factuality comparison [Gao et al., 2023]. LLMs can also be used to answer qualitative questions about the factual consistency of a summary and explain the reasoning behind the provided answers [Luo et al., 2023].

Source Modality - Transcript vs. Speech: We are also interested in studying whether input modalities have an impact on task performance. Cognitive psychology literature has studied this question in various settings. [Khan et al., 2022] studies cognitive performance and information retention, when note-taking is performed using voice notes as opposed to textual notes. The paper finds that when the notes are taken by recording voice notes as opposed to writing textual notes, note-taking leads to a better conceptual understanding of the topic. [Stollnberger et al., 2013] studies how the input modality affects human learning of the English language, specifically whether learning by listening to lecture audio as opposed to reading lecture notes impacts retention. In this chapter, we expand the analysis to the important task of speech summarization, which is significantly different from the tasks addressed in prior work.

Impact of ASR on Downstream Tasks: Errors in automatic speech recognition are known to have an impact on downstream task performance, like emotion recognition [Schuller et al., 2009, Feng et al., 2020], spoken language understanding [Shon et al., 2023], and natural language understanding and dialog management [Serdyuk et al., 2018, Shivakumar et al., 2019]. Therefore, it is important to assess their impact on human summarization.

4.3 Data Collection

Datasets that are used for speech summarization today do not have information on which of the two approaches, i.e., reading a textual transcript of the recording or listening to the recording to summarize was employed to obtain annotation. Further, they only have one annotation per recording, which may inhibit analysis due to annotator-specific biases or tendencies. Therefore, to conduct our experiments, we curate a multi-reference dataset from an existing dataset.

We considered a subset from the test partition of the *Interview* corpus [Sharma et al., 2023a, Majumder et al., 2020, Zhu et al., 2021] for speech summarization. *Interview* is the largest open-domain corpus with around 5-minute-long speech recordings containing spontaneous speech along with real audio noises representing music, and events inside and outside the studio corresponding to discussion topics[1].

Ensuring high-quality summary annotations is crucial to making reliable observations. Therefore, we took the necessary steps to ensure a fair and reliable data collection pipeline. Initially, we consulted with an expert from the in-house data collection team and conducted small pilot studies to make design decisions.

Audio Length: We conducted in-house pilot studies comprising six questions by trying to summarize audio recordings that were 2 minutes and 5 minutes long. As expected, summarizing 5-minute recordings consumed more time and effort, and was harder due to the large amount of information compression. We concluded that limiting the audio length to 2 minutes would provide sufficient content to summarize without increasing the difficulty level of summarization significantly. Using forced alignment, we obtained audio reference transcripts up to the first 2 minutes.

Summary Length: A hard limit was not set on the length of the summary, but it was suggested that summaries should contain at least 2 sentences to convey the essence of the conversation. It was recommended that summaries contain between 50 to 80 words for the 2-minute long recordings.

Data Selection: To obtain reasonable summaries, it is important that audio recordings and textual transcripts are understandable, and have a sufficient amount of information to summarize. In cases where background noise or music is present, or when a large portion of the recording does not contain speech, these criteria are violated, leading to sub-optimal recordings for summarization. Therefore, audio recordings were transcribed using `Whisper-medium` [Radford et al., 2022], and the resulting Word Error Rate (WER) was used to remove unsuitable audio recordings from

[1] The authors acquired permission from NPR to use NPR data for this research

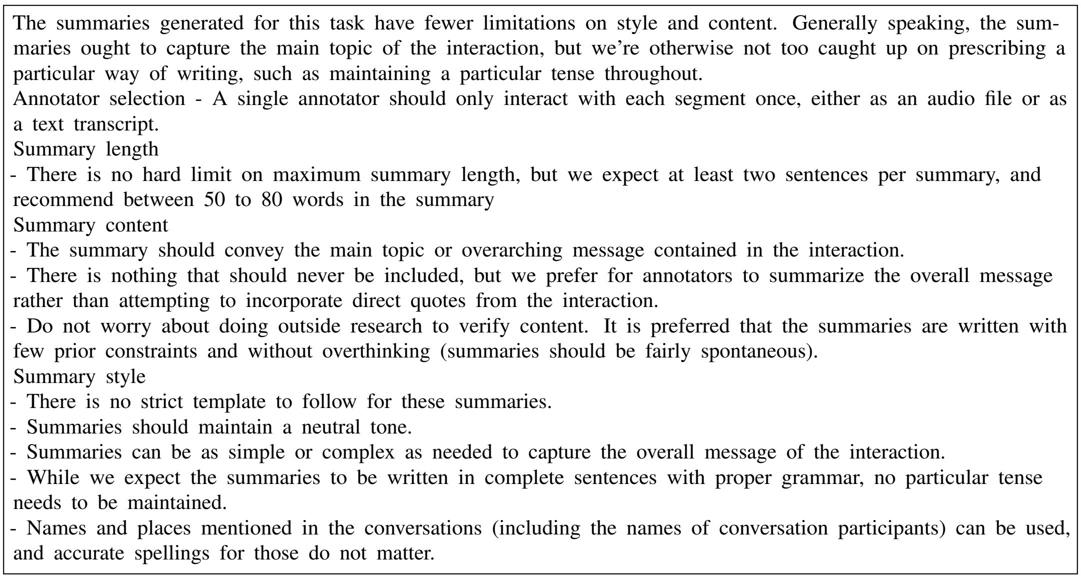
consideration. Based on manual inspection, a threshold of 25% WER was set, which eliminated recordings with only music or large amounts of noise.

Expert and Non-expert: Our data comprises 1,002 recordings in total, with each recording having a speech recording and its corresponding reference transcript. For each recording, we collected two expert summaries and two non-expert summaries by reading the text or listening to the audio, which gave us a total of 4 summaries based on the text and 4 summaries based on the audio on the same recording. Additionally, we solicited summary annotations from speech recognition transcripts for 100 of these 1002 recordings, sampled at random so the Word Error Rate with `Whisper-medium` is non-zero, i.e., the transcript has ASR errors.

Annotator Overlap: It is worth noting that each annotator was assigned to work on either the transcript or audio recording for a particular recording, but not both. This approach ensured that no annotator saw audio or transcription for a recording they had already annotated.

Expert Summary Annotation: Expert summary annotations were obtained through a third-party vendor that passed in-house qualitative assessments of summary outputs from a pilot annotation.

Figure 4.2 shows the detailed guideline we provided for expert annotation to third-party vendors.



The summaries generated for this task have fewer limitations on style and content. Generally speaking, the summaries ought to capture the main topic of the interaction, but we're otherwise not too caught up on prescribing a particular way of writing, such as maintaining a particular tense throughout.
Annotator selection - A single annotator should only interact with each segment once, either as an audio file or as a text transcript.
Summary length
- There is no hard limit on maximum summary length, but we expect at least two sentences per summary, and recommend between 50 to 80 words in the summary
Summary content
- The summary should convey the main topic or overarching message contained in the interaction.
- There is nothing that should never be included, but we prefer for annotators to summarize the overall message rather than attempting to incorporate direct quotes from the interaction.
- Do not worry about doing outside research to verify content. It is preferred that the summaries are written with few prior constraints and without overthinking (summaries should be fairly spontaneous).
Summary style
- There is no strict template to follow for these summaries.
- Summaries should maintain a neutral tone.
- Summaries can be as simple or complex as needed to capture the overall message of the interaction.
- While we expect the summaries to be written in complete sentences with proper grammar, no particular tense needs to be maintained.
- Names and places mentioned in the conversations (including the names of conversation participants) can be used, and accurate spellings for those do not matter.

FIGURE 4.2: Annotation guidelines provided to expert annotators to collect summary data

In total, there are 36 expert annotators. The expert annotators are all American national and native English speakers, proficient in the language, therefore they don't have any difficulty performing the summarization task. Among the 36, 34 are female, one identifies as non-binary and one prefers not to say.

The third-party vendor passed in-house qualitative assessments of summary outputs from a pilot annotation and contracted to the company's primary third-party vendor for summary annotation.

Non-Expert Summary Annotation Non-expert summary annotations were obtained through the Amazon Mechanical Turk (MTurk) platform. Each HIT required the annotator to accept the terms of a consent form that explained the task and requested consent to share responses for reproducible research. Each HIT included four questions, where annotators were asked to write 2 summaries by listening to audio recordings, and two summaries by reading transcript passages.

As part of the instructions, the use of generative AI was prohibited to answer the questions.

Annotators were asked to write summaries based on the provided inputs while ensuring that the response was coherent and grammatical, and written in their own words. All responses were validated manually for conformity to the instructions, and invalid responses were rejected and re-assigned to a new annotator.

Source

Towns like Bluffton say they're working to promote communities of inclusiveness, but what does that really mean, and will people of color come? Jim Hunt is president of the National League of Cities based in Washington, D.C. He heads up the group's partnership for working toward inclusive communities. Mr. Hunt joins us from West Virginia Radio in Morgantown, West Virginia..And joining us via phone is Harvard law professor Charles Ogletree. Professor Ogletree leads the law schools' Charles Hamilton Houston Institute for Race and Justice. Gentlemen, great to have you on the program. Professor, let me start with you. Give us a quick historical view, of you will, of these, quote, sundown towns..Well, thanks, Ed. As you know, when I was doing research on the Tulsa race riots from 1921, I was astounded to learn of the literally hundreds of towns - now we've discovered thousands of towns around America - in the South, but not exclusively in the South, where blacks were told leave before sundown. There were sirens, there were notices, and the consequences of staying in those towns were death or other serious bodily injury..And the good news is that that is largely historical, but it's a frightening sense that people could not live in the community. They could work there, they could visit there, but they couldn't' be there after dark because of the strongly held feelings about segregation. And sundown towns were just that: if you're black, get out of town before the sun goes down..Mm hmm. Jim Hunt, while those sirens may have gone away, there are certainly still areas in this country where when the sun goes down, minorities know that this is an area - a region you should not be in. That continues today..Right, Ed. And I think when we look around - and, obviously, the inclusive agenda is a broader agenda than just race, but clearly race plays a very significant part in the program..Mr. Hunt, why are we seeing this sense of inclusiveness? Is it totally altruistic? Are there financial benefits to these cities? Why are we seeing this rush now to try to move to include minorities?.Well, and I think when we look across the board of what's happening across the country - when we look at Dr. Richard Florida's work on the rise of the creative class, and some of the economic development needs of cities throughout America - we recognize that if we're not going to be inclusive, we're going to suffer from an economic development perspective.

Expert Speech Summary

Some towns in America used to be known as sundown towns because black people were told to leave town before sundown or they could be harmed. Some cities today have programs that focus on being inclusive to minorities.

Expert Text Summary

Jim Hunt, President of the National League of Cities, and Harvard Law Professor Charles Olgetree discuss sundown towns and why some towns are working to be more inclusive of minorities. A sundown town was a town, mostly in the south, where blacks were allowed work and visit, but were not allowed to be there after sundown. If they were, the could face serious injury or even death. While these town have mostly gone away, there are still areas in the country that minorities know they should not be in when the sun goes down. Many towns are now working to be more inclusive of minorities and part of the reason has to do with the financial benefits they'll receive.

Non-expert Speech Summary

Jim Hunt, President of the National League of Cities, and Harvard law professor Charles Ogletree discuss the historical and contemporary implications of "sundown towns," where African Americans were forced to leave before sunset under threat of violence. While this practice is largely historical, there are still areas where minorities feel unsafe after dark. They emphasize the importance of inclusivity for economic development and societal well-being.

Non-expert Text Summary

The National League of Cities is working on inclusive communities, addressing historical issues like "sundown towns," where minorities were told to leave before dark under threats of violence. Harvard Law Professor Charles Ogletree discusses the historical prevalence of such towns, emphasizing the progress made. National League of Cities President Jim Hunt highlights the economic benefits of inclusivity in cities, linking it to overall economic development. While acknowledging past discrimination, the focus is on building inclusive communities for the future.

FIGURE 4.3: Example source and summaries.

Figure 4.3 shows some examples of the annotations we obtained.

4.4 Evaluating differences between summaries

Summary evaluation is an inherently challenging problem for general settings since there is often no clear definition of what constitutes a good summary. Within this chapter, we deem a summary good if it is factually faithful to and representative of the input source while being abstractive, coherent, and fluent. Therefore, in this section, we outline three types of evaluations that evaluate different aspects of summary quality and informativeness.

4.4.1 Source-based Evaluation

Summary Length and Compression Ratio: Summary length is measured in words and the compression ratio [Mani, 1999] is defined as the number of words in the reference transcript divided by the number of words in the summary. The greater the summary length, typically, the higher the amount of information contained in the summary, and the lower the compression ratio.

Novel Words (%): The percentage of words in the summary that are not present within the source transcript, which could be considered to measure the extent of paraphrasing within the summary.

Extractiveness: It is a measure of how extractive the summary is and can be approximated by computing lexical overlap using ROUGE-L [Lin and Och, 2004] between the source transcript and the summary.

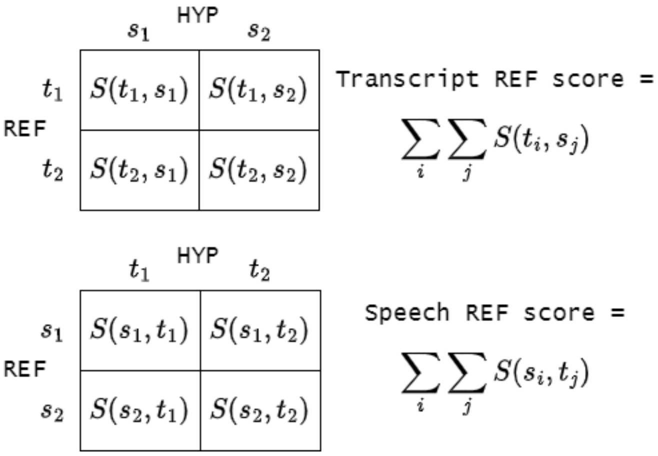
Informativeness-Entities: We propose to compare named entities predicted from the source text and summaries as a proxy for the amount of information contained within the summary. Named entities are extracted using the Entity Recognizer[2] from Spacy [Honnibal and Montani, 2017].

Informativeness-Semantic Similarity: Also, we compare semantic similarity between the source and summaries using metrics like BERTScore [Zhang et al., 2019] and BARTScore [Yuan et al., 2021].

Retrieval-based informativeness: A summary can be considered representative of or informative about the source when it can discriminate the correct source from a selection of all sources. We describe a retrieval-based measure where we compute the semantic embedding similarity (using BERTScore) between the hypothesis summary and all possible source texts. A discriminative summary should produce the highest similarity between the summary and the *correct* source text. This is measured using retrieval accuracy (RAcc). We also report the Mean Reciprocal Rank (MRR) as an indicator of retrieval performance.

Factual Consistency: We use UniEval to estimate the factual consistency of summaries with respect to the source transcript.

[2] https://spacy.io/api/entityrecognizer



$$\begin{array}{c|cc} & s_1 & s_2 \\ \hline t_1 & S(t_1, s_1) & S(t_1, s_2) \\ t_2 & S(t_2, s_1) & S(t_2, s_2) \end{array}$$

Transcript REF score =

$$\sum_i \sum_j S(t_i, s_j)$$

$$\begin{array}{c|cc} & t_1 & t_2 \\ \hline s_1 & S(s_1, t_1) & S(s_1, t_2) \\ s_2 & S(s_2, t_1) & S(s_2, t_2) \end{array}$$

Speech REF score =

$$\sum_i \sum_j S(s_i, t_j)$$

FIGURE 4.4: Pairwise Score Computation

4.4.2 Structure Evaluation

To evaluate coherence and fluency automatically, we leverage UniEval [Zhong et al., 2022b].

4.4.3 Summary Comparisons

Pairwise Similarity: We compute relevance and consistency using UniEval [Zhong et al., 2022b], and use BARTScore as a measure of semantic similarity. Pairwise scoring is illustrated in Fig. 4.4, where s_i is a speech-based summary, t_j is a transcript-based summary, and $S(a, b)$ calculates the score between reference a and hypothesis b (e.g., BART score). The idea behind pairwise scoring is to compare two types of summaries, where one type is a reference and the other a hypothesis. This process results in two scores, a speech reference score, and a transcript reference score. By comparing these two scores, we can identify cases when the information in one type of summary is present within the other type of summary.

Pairwise Inter Annotator Agreement (IAA): For text generation tasks, the inter-annotator agreement is challenging to quantify. We use a combination of lexical and semantic similarity metrics like ROUGE-L [Lin and Och, 2004], and BARTScore [Yuan et al., 2021] to automatically represent inter-annotator agreement.

Factual Consistency and **Source Representation**: We employ LLM-based evaluation and human evaluation to estimate the factual consistency of each summary, and how well each summary represents the source. These evaluation methods are described in detail below.

4.4.3.1 LLM-based Evaluation

Following the LLM summary evaluation methods listed in Section 4.2, we use ChatGPT (`gpt-3.5-turbo-0125`) as a qualitative evaluation method for the summaries collected for this work. Specifically, we designed a multiple choice-style AB test to compare summaries in

```
The following are two summaries of the given source text:
Summary 1: [Summary 1]
Summary 2: [Summary 2]
Source: [Source]
                                Factual Consistency Question
Answer the following multiple-choice question:
What is the most appropriate statement about the summaries?
(a) Summary 1 is more factually consistent with the source text than Summary 2.
(b) Summary 2 is more factually consistent with the source text than Summary 1.
(c) Both Summary 1 and Summary 2 are equally factually consistent with the source text.
Give your answer as “Answer: a”, “Answer: b”, or “Answer: c”. Then, start a new line and explain
your reasoning in a single paragraph following your answer.
                               Source Representation Question
Answer the following multiple choice question:
What is the most appropriate statement about the summaries?
(a) Summary 1 is more representative of the source text than Summary 2.
(b) Summary 2 is more representative of the source text than Summary 1.
(c) Both Summary 1 and Summary 2 are equally representative of the source text.
Give your answer as “Answer: a”, “Answer: b”, or “Answer: c”. Then, start a new line and explain
your reasoning in a single paragraph following your answer.
```

FIGURE 4.5: Template for questions presented to ChatGPT. Similar questions are used to solicit human scores.

terms of either factual consistency or accurate representation of the source transcript. The two questions posed to the LLM are displayed in Fig. 4.5, where the text in the brackets represents the summaries being evaluated and the associated source text. Each question was presented in a separate session to avoid bias from previous questions.

Each question is asked twice for each AB pairing, but with the order in which the summaries are presented switched to avoid biasing the results. For example, if we were comparing a transcript-based summary and a speech-based summary, the question would be asked one time with the transcript-based summary as Summary 1, and then another time with the speech-based summary as Summary 1. Additionally, since all summaries are collected in pairs, we compare each summary to both summaries from the other group, exhausting all possible combinations. For example, if we were comparing a pair of transcript-based summaries to a pair of speech-based summaries, the first transcript-based summary would be compared to both speech-based summaries, and then the second transcript-based summary would be compared to both speech-based summaries, resulting in four total comparisons. Considering both the order of presentation and the comparison combinations, eight questions are posed to the LLM per source sample.

4.4.3.2 Human Evaluation

To validate the expert summary comparison, we also conducted an AB test for 30 recordings with 14 annotators. We present 2 summaries based on different modalities and their source transcript to a user. We designed 2 multiple-choice questions that ask human annotators to compare summaries based on their factual consistency and how accurately they represent the source transcript. The setup of questions is similar to that in LLM evaluation.

4.5 Research Methodology

4.5.1 RQ1: Speech versus Text Inputs

We apply the methods and metrics described in Section 4.4 to explore differences between summaries derived from speech recordings and summaries extracted from text transcripts. We use the expert annotations obtained for 1002 recordings for this analysis since expert summaries are more reliable.

4.5.2 RQ2: Impact of Transcript Errors on Human Summarization

We measure the impact of ASR error propagation on the output summary by comparing a subset of 96 expert summaries given ASR-generated transcripts and ground-truth transcripts. The summaries resulting from the Whisper-generated transcripts are compared to the summaries based on ground-truth transcripts using the methods and metrics outlined in Section 4.4.

We analyze the effect of ASR errors on the quality of the summaries based on ASR transcripts by plotting informativeness measures like Entity-F1 and WER (See Figure 4.6) and find that as WER increases, Entity-F1 decreases.

4.5.3 RQ3: Expert versus Non-Expert Annotation

We use all methods and metrics from Section 4.4 to explore the effect of expertise on summary quality. Similar to RQ1, we provide an overall comparison across speech-based summaries as opposed to transcript-based summaries arising from non-expert annotations and conclude on the reliability of expert and non-expert summaries.

4.6 Experimental Results

4.6.1 RQ1: Speech versus Transcript Inputs

Evaluation across the expert summaries, including a comparison between transcript-based and speech-based summaries, is shown in the first two columns of Table 4.1.

It is immediately apparent that speech-based summaries are significantly shorter and more compressed than transcript-based summaries, indicating that speech-based summaries are highly selective. This is likely because the summarizers extract only the main points when listening to a recording, while a transcript serves as an easier reference for smaller details. This claim is supported by the low level of extractiveness in the speech-based summaries, indicated by a high percent of novel words, low ROUGE-L scores, and a low F1 score for named entities retained

Metric	Transcript Overall	Speech Overall	p-value
Summary Length	**78.58** ± **27.72**	56.95 ± 18.30	
Compression Ratio	3.76 ± 2.20	**5.20** ± **3.06**	8.83E-86
Word Vocabulary Size	20,408	16,211	-
Novel Words %	24.63 ± 13.57	**25.66** ± **13.22**	5.80E-02
Extractiveness↓	20.70 ± 10.53	**18.57** ± **10.88**	1.09E-08
Entity F1↑	**33.63** ± **29.83**	26.21 ± 26.90	1.34E-30
BERTScore ↑	**85.66** ±**2.45**	85.32 ±2.42	1.54E-05
BARTScore ↑	-3.37 ±0.47	-3.46 ± 0.45	7.96E-11
Retrieval Accuracy ↑	**68.86 +** ± **46.31**	59.04 ± 49.75	1.36E-19
Mean Reciprocal Rank ↑	70.23 ± 16.08	**75.74** ± **16.29**	8.48E-52
Factual Consistency (UniEval) ↑	0.83 ± 0.16	**0.85**± **0.16**	1.90E-04
Pairwise Rep. (LLM) ↑	**66.67**	28.52	-
Pairwise Factualness (LLM) ↑	**60.16**	33.07	-
Pairwise Rep. (Human) ↑	**48.57**	21.43	-
Pairwise Factualness (Human) ↑	**39.52**	22.62	-
Fluency (UniEval) ↑	**94.02**± **4.54**	94.01± 5.26	9.83E-01
Coherence (UniEval)↑	**91.82**± **12.97**	89.34± 15.63	5.42E-08
Pairwise BARTScore ↑	-2.81±0.51	**-2.56**± **0.52**	1.67E-101

TABLE 4.1: RQ1 Evaluation: Speech versus Transcript-based Summaries by expert annotators. ↑ signifies that a higher value of the metric is desirable, while ↓ signifies that a lower value of the metric is desirable.

in the transcript. Additionally, coherence for speech-based summaries is relatively low, likely because they are not long or detailed enough to form a very structured summary.

These results also indicate that speech-based summaries are less informative than transcript-based summaries. Again, this is likely because summarizers can extract details by referencing transcripts much more easily than by searching through the recording. Results that support this claim are high BERT- and BARTScores, high retrieval accuracy, factual consistency, and human and LLM ratings. The higher pairwise BARTScore for speech references indicates that more of the information within the speech summary is likely present in the transcript-based summary, making speech-based summaries more selective in relevant information. IAA scores computed based on ROUGE-L and BARTScore indicate that speech-based summaries promote higher consensus between raters.

The only result that did not exhibit statistically significant influence from the source modality was fluency. This is reasonable, as the fluency of the model is determined more by the language proficiency of the individual writing the summary than the content.

For **RQ1**, we find that *speech-based summaries seem to be more selective, factually consistent, and perhaps more abstractive than transcript-based summaries, while transcript-based summaries contain more information extracted directly from the source than their speech-based counterparts.*

Metric	Transcript Overall	Speech Overall	ASR Overall
Summary Length	**85.29**±**40.58**	56.94±36.89	73.67±36.89
Compression Ratio	0.40± 0.73	**0.52**± **0.81**	0.27± 0.85
Extractiveness (ROUGE-L with Transcript) ↓	21.94±10.95	**19.17**±**11.60**	19.25±9.33
Entity F1 ↑	**35.17** ±**18.93**	26.28 ± 15.65	30.24±17.67
BERTScore with Transcript↑	**85.92** ± **2.55**	85.56±2.55	85.49±2.45
BARTScore with Transcript↑	**-3.33** ± **0.48**	-3.44±0.47	-3.41 ±0.45
Retrieval Accuracy↑	**85.42**± **35.29**	71.88± 44.96	77.60± 41.68
Mean Reciprocal Rank↑	**89.09**± **27.55**	79.44 ±34.27	83.49±32.11
Factual Consistency (UniEval)↑	83.56 ± 14.45	**86.16**±**13.83**	83.48±14.25
Fluency (UniEval)↑	94.31 ± 3.62	94.23 ± 3.90	**94.36** ± **3.72**
Coherence (UniEval)↑	**91.51** ± **14.90**	88.64 ± 16.04	90.81 ± 13.34
Pairwise Rep. (LLM)↑	**40.06**	21.57	33.81
Pairwise Factualness (LLM)↑	**37.76**	22.74	32.81
Pairwise BARTScore↑	-2.81± 0.48	**-2.59** ± **0.51**	-2.79 ± 0.51

TABLE 4.2: RQ2 Evaluation: Impact of Transcript Errors on Human Summarization.↑ signifies that a higher value of the metric is desirable, while ↓ signifies that a lower value of the metric is desirable.

4.6.2 RQ2: Impact of Transcript Errors on Human Summarization

Table 4.2 compares the summaries based on the subset of 100 recordings. It is interesting to compare the summaries based on ASR transcriptions to the summaries based on manual transcriptions. Nearly all indicators of informativeness are significantly higher for the manual transcriptions than the ASR transcriptions. This is to be expected, as propagated recognition errors can directly affect measures like Entity F1, factual consistency, and semantic similarity. Coherence is similar for the two types of transcripts, which is consistent with the idea that the ability to reference text allows the summarizer to capture the details and structure of the source.

Comparing ASR-transcript-based summaries to speech-based summaries, we see similar trends to those in RQ1. Speech-based summaries are still shorter, more selective, less informative, more factually consistent, and likely more abstractive than ASR-transcript-based summaries. In general, LLM evaluation indicates that ChatGPT prefers transcript-based summaries over speech-based summaries, even if the source transcripts contain errors.

Figure 4.6 shows a scatter of Entity-F1 as a function of WER. As the WER increases, Entity-F1 decreases, showing a weak negative correlation.

For **RQ2**, we find that *transcription errors decrease the informativeness and coherence of transcript-based summaries, but do not significantly impact fluency*.

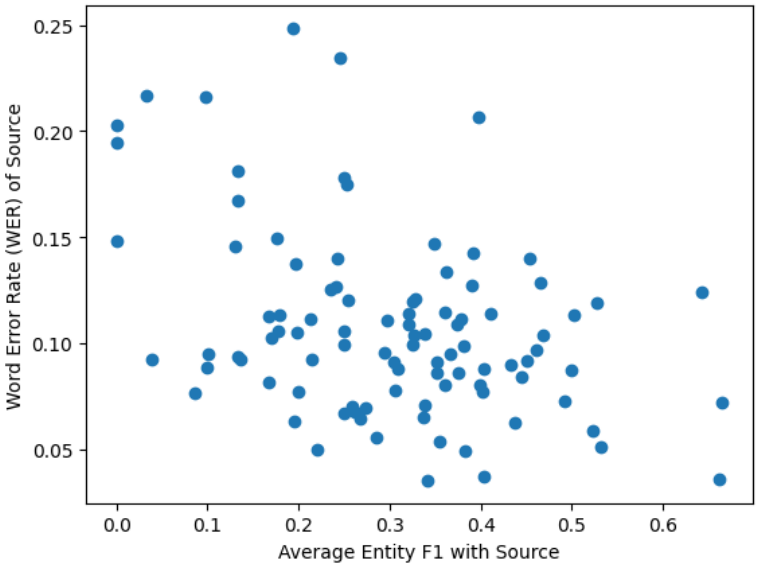


FIGURE 4.6: Scatterplot showing variance of entity F1 of summary compared to source reference transcript with Word Error Rate (WER) of audio transcription

4.6.3 RQ3: Expert Annotation versus Non-Expert Annotation

From Table 4.3, we observe that the speech-based summary length from expert annotation is significantly lower than its non-expert counterpart, while the expert speech-based and transcript-based summary lengths are very different. The word vocabulary size and the percentage of novel words are lower in the expert summaries (both speech and transcript sources) compared to the non-expert summaries.

Extractiveness is lower for non-expert speech-based and transcript-based summaries compared to the expert ones, referring to the fact that the non-experts do not tend to use phrases directly from the source whereas the experts do. Entity F1 score is higher for non-expert speech-based and transcript-based summaries, which is because non-expert summaries are on average longer than the expert ones and capture more of the entities in the source. BERTscore, BARTscore, retrieval accuracy, and MRR are higher for the expert summaries than the non-expert summaries (for both speech and transcript sources) referring to the fact that non-expert summaries are less informative than the expert summaries. Factual consistency is lower for non-expert summaries pointing to the fact that expert summaries are more factually consistent. Fluency, coherence, and pairwise BARTscores are lower for non-expert summaries demonstrating the reliability of expert summaries.

We ran t-tests to conclude whether the non-expert speech-based and transcript-based summaries belong to the same distribution and observed based on BARTScore and other metrics that the p-value was high. This potentially indicates that non-expert annotations based on speech and transcripts may be very similar and that some annotators may have used similar methods or tools for the two types of annotations. Based on a human examination, it appears that some annotators did use generative AI to respond to annotation requests through Amazon

	Expert		Non-expert	
Metric	Transcript	Speech	Transcript	Speech
Summary Length	78.58 ± 27.72	56.95 ± 18.30	75.35 ± 17.70	77.42 ± 33.85
Compression Ratio	3.76 ± 2.20	5.20 ± 3.06	3.69 ± 1.95	3.66± 1.86
Word Vocabulary Size	20,408	16,211	21511	21676
Novel Words %	24.63 ± 13.57	25.66 ± 13.22	37.47 ± 13.38	38.45± 14.21
Extractiveness↓	20.70 ± 10.53	**18.57 ± 10.88**	**18.37± 7.26**	18.38 ±11.29
Entity F1↑	**33.63 ± 29.83**	26.21 ± 26.90	**34.51 ± 17.28**	30.09± 17.72
BERTScore ↑	**85.66 ±2.45**	85.32 ±2.42	**85.28± 2.32**	85.06 ±2.76
BARTScore ↑	**-3.37 ±0.47**	-3.46 ± 0.45	**-3.41± 0.44**	-3.43± 0.53
Retrieval Accuracy ↑	**68.86 + ± 46.31**	59.04 ± 49.75	**71.82 ± 44.99**	65.78 ± 47.44
MRR ↑	70.23 ± 16.08	**75.74 ± 16.29**	**76.56 ± 71.01**	71.01 ± 41.57
Factual Consistency (UniEval) ↑	82.68 ± 16.38	**84.58± 15.72**	**82.08 ± 16.90**	78.57± 19.79
Fluency (UniEval) ↑	**94.02± 4.54**	94.01± 5.26	**92.90± 8.21**	92.65 ±9.35
Coherence (UniEval)↑	**91.82± 12.97**	89.34± 15.63	**90.88± 15.19**	87.95± 19.57
Pairwise Rep. (LLM) ↑	**66.67**	28.52	-	-
Pairwise Factualness (LLM) ↑	**60.16**	33.07	-	-
Pairwise Rep. (Human) ↑	**48.57**	21.43	-	-
Pairwise Factualness (Human) ↑	**39.52**	22.62	-	-
IAA - ROUGE-L ↑	24.92 ± 7.82	**28.98 ± 9.29**	**32.67 ± 12.56**	28.62 ± 12.52
IAA-BARTScore ↑	**-2.71 ± 0.53**	-2.55 ± 0.55	**-2.72 ± 0.61**	-2.93 ± 0.73
Pairwise BARTScore ↑	-2.81±0.51	**-2.56± 0.52**	**-2.85± 0.63**	-2.88± 0.70

TABLE 4.3: RQ3 Evaluation: Speech-based versus Transcript-based Summaries for Expert and Non-expert annotators.↑ signifies that a higher value of the metric is desirable, while ↓ signifies that a lower value of the metric is desirable.

Mechanical Turk, however, such conclusions may not be definitive since such models can generate human-like speech. This makes non-expert summaries less reliable for analysis of speech-based versus transcript-based summaries, and hence, we decided against running the LLM and human evaluation for these non-expert summaries.

For **RQ3**, we find that *expert annotations are more informative, coherent, fluent, and less information variant across speech-based and transcript-based summaries*.

4.7 Chapter Conclusion

In this work, we take the first steps towards understanding human annotation for speech summarization by proposing and assessing different approaches. Annotators can either read a transcript of spoken content or listen to the audio recording to produce an abstractive speech summary. We formulate three research questions to assess differences in summaries resulting from variations in source modality (listening to audio versus reading a transcript of spoken content), the nature of transcription (manual versus ASR-based), and the expertise of annotators. Extensive analysis using automatic and human evaluation reveals that speech-based summaries are more information-selective, factually consistent, and abstractive compared to transcript-based summaries. Analysis of the impact of ASR on transcript-based summarization shows that errors in transcript generation decrease the informativeness and coherence of transcript-based summaries. Finally, analysis of expert and non-expert annotations demonstrates higher informativeness, coherence, and fluency across expert annotations.

Through this process, we develop and release a new human-annotated dataset to enable reproduction and encourage further research in the area. We hope that this study galvanizes further research into human summarization, and powers informed model design for automatic speech summarization.

Part II

Our End-to-End Approach

Chapter 5

The End-to-End Approach to Speech Summarization

Speech summarization is typically performed using a cascade approach. The input speech is first converted into its text transcript using an ASR model. The resulting transcript is used as input to an ATS model to produce abstractive summaries. The challenge with such a setup is error propagation across the cascade, i.e., where transcription errors amplify errors in the summarization module.

To minimize the impact of error propagation and construct simpler models, in this chapter, we present an end-to-end sequence model that is optimized for speech summarization. The resulting end-to-end model comprises a speech encoder that transforms the input speech into a latent sequence representation, and a summary decoder that uses the latent representation to auto-regressively generate the summary.

To realize the end-to-end model, we address two challenges: *abstract learning*, and *global acoustic context*. Abstract learning, characterized by an indirect relationship between the input and output is mitigated using a two-stage training strategy, where we first generate grounded acoustic representations by pre-training for ASR. We use these grounded acoustic representations to train the summarization model.

To consider *global acoustic context*, we need to be able to train with the entire speech sequence, which is very long. Transformers are among the state of the art for many tasks in speech, vision, and natural language processing, among others. Self-attentions, which are crucial contributors to this performance have quadratic computational complexity, which makes training on longer input sequences challenging. Therefore, we introduce optimizations to self-attention to make them accurate and efficient over long input sequences.

5.1 The Problem with Cascade Models

As discussed in Chapter 3, the cascade comprises speech recognition and text summarization modules in most cases. The main challenge with cascade models is error propagation across modules of the cascade. Errors made within the speech recognition module create noisy transcripts that are input to the subsequent text summarization module. Noisy inputs impact the performance of the text summarization module, leading to noisy outputs. In essence, errors in a module cause errors in subsequent modules within a cascade, which is termed error propagation. This phenomenon can have a significantly detrimental impact on the performance of cascade systems. The greater the number of modules within the cascade, the greater the impact of such error propagation. Errors within initial modules are often more inimical than errors in later modules.

To overcome this fundamental disadvantage, there are a few strategies that can be adopted. First, modules after the first can be trained to be robust to noisy inputs. However, any lost information from the input cannot be recovered, leading to suboptimal performance. Second, error correction modules can be inserted to attempt to recover missing information. However, these incur a significant computational cost and may not be able to recover the lost information completely. These considerations motivate us to optimize speech summarization models in an end-to-end manner, which encourages intermediate outputs that are likely to lead to accurate summaries.

Another challenge within cascade models is the necessity for a larger number of model parameters. The total number of parameters within a cascade is the sum of model parameters within individual modules. This increases the memory and computational cost for such models, making such approaches unsuitable for constrained deployments, like on-device. This can be addressed by making an end-to-end formulation relatively lightweight, comprising a single encoder and decoder as opposed to multiple encoders and decoders.

In the sections that follow, we describe our end-to-end framework for the training and inference of speech summarization models and experimentally demonstrate its benefits.

5.2 Our End-to-End Formulation

Given a recording with input speech features $X = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots \mathbf{x}_M]$, where $\mathbf{x}_i$ represent N dimensional speech features; the objective of speech summarization is to estimate the model parameters θ_{SSUM} that maximize the probability of predicting the correct abstractive summary token sequence $Y = [y_1, y_2, y_3, \cdots y_L]$ of length L given the input speech feature sequence X.

We realize this by modeling $P(Y|X;\theta)$ using a sequence model comprising a speech encoder and a summary decoder. The encoder takes in the speech features X and learns to produce a latent representation H based on Equation 5.1.

$$H = \text{Encoder}(X) \tag{5.1}$$

The decoder consumes this latent representation H and auto-regressively predicts tokens of the hypothesis summary $\hat{Y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots \hat{y}_L]$ based on equation 5.2.

$$\hat{y}_i = \text{Decoder}(y_1, y_2, \cdots, y_{i-1}, X) \tag{5.2}$$

Each token in the decoder predictions is one of V elements within the decoder's token vocabulary. The token vocabulary is constructed from the text of the training data. Additional tokens like $< unk >, < sos >, < pad >$ are used to indicate an unknown token, the start/end of the prediction token, and the pad token respectively.

The model is optimized using the cross-entropy criterion, defined as follows:

$$L_{ce} = -\sum_{i=1}^{L}\sum_{j=1}^{V} M(i) y_i \log(\hat{y}_i) \tag{5.3}$$

Here $M(i)$ is an indicator mask function that has the value of 1 if the current token is not a pad token in the reference y and 0 otherwise.

Speech encoders can adopt transformer-based [Vaswani et al., 2017b] architectures or recurrent [Graves, 2012] architectures. Throughout this thesis, we leverage conformer [Gulati et al., 2020a] architecture with learnable convolutional feature downsampling for our speech encoder. The summary decoder is transformer-based.

Inference for a trained end-to-end model with parameters θ_{SSUM} aims to find the most likely summary $\hat{Y}$ given the input speech feature sequence X as shown in Equation 5.4.

$$\hat{Y} = \text{argmax}_Y P(Y|X; \theta_{SSUM}) \tag{5.4}$$

The end-to-end model auto-regressively predicts tokens in the summary $\hat{Y}$ based on previously decoded tokens.

5.3 Challenges in End-to-End Modeling

There exist multiple challenges in realizing end-to-end models for real-world applications.

First, end-to-end models consume long input sequences directly at once to make decisions on what is important enough to summarize. However, such models cannot be realized in real-world settings owing to compute restrictions. In this thesis, we address this challenge of global acoustic context by proposing multiple approaches within Chapters 5 and 7 to enable speech summarization of long recordings.

Second, speech summarization is an example of an abstract learning task, where the relationship between the input speech and output summary is complex and indirect. This makes optimization from scratch challenging. To address this, in this thesis, we propose a multi-stage training strategy to facilitate performant end-to-end models.

Third, there is a lack of large-scale labeled data across multiple domains of application for speech summarization. To address this challenge, through this thesis, we release two additional corpora for speech summarization of TED talks and radio interviews later in the next chapter.

Finally, summarization models are trained to produce summaries based on unitary references, and evaluated against the same. However, given an input recording, there may be multiple equally valid summaries. Further, summaries ought to be human-like, which implies a diversity in phrasing. However, this is limited by the diversity and type of dataset(s) used in training. To make model outputs more diverse, in this paper, we introduce AugSumm, a method to generate and use additional summary references during training and evaluation.

5.4 Abstract Learning of Complex Mappings

5.4.1 The Challenge of Abstract Learning

Abstractive speech summarization involves selecting important portions of the input, combining the information within these important portions, and then paraphrasing to rewrite as human-like summaries. Because all of these operations are complex, this results in the mapping between speech and the resulting summary being complex and indirect. Due to importance selection being somewhat subjective due to human annotation, it is challenging to teach models to select these important portions of a given recording. Due to the combination of important semantic concepts into a coherent logical progression, and the fact that multiple coherent progressions can result, the alignment between the input speech and these important semantic concepts is complex and non-monotonic. Many words within the output text may not be present verbatim in the input speech as a result of paraphrasing, leading to cases when some tokens in the output may not be lexically aligned with any part of the input.

5.4.2 Our Solution

Training end-to-end models from scratch that can summarize speech from scratch is challenging owing to the complex unconstrained mapping between input speech and output summary. Therefore, we derive inspiration from the fundamental tenet of curriculum learning, to solve easier problems first and then progress to increasingly challenging problems.

An abstractive summary may be composed of extractive elements and abstractive elements. The extractive elements, such as entity mentions, are present within the input speech, while the abstractive elements do not have the same lexical form as content within the input due to paraphrasing. Models need to be able to produce both these types of elements to do abstractive speech summarization.

The portions of the summary that are extractive, i.e., present in the input are easier to predict compared to the paraphrased portions since there are frames in the input speech that directly represent all of these words and/or phrases. Further, these extractive portions are present within the reference transcript of speech. Teaching the model to transcribe speech will in turn teach the model to be able to predict these extractive elements. Therefore, we contend that for tasks with extractive elements within the abstractive summary, it is useful to pre-train the end-to-end model for speech recognition first.

Speech recognition can be performed over long-form inputs, but it is harder to recognize long-form inputs in comparison to shorter utterances. Therefore, we consider the simpler form of speech recognition, i.e., to recognize individual utterances in speech as the first step of pre-training. To obtain utterance-level transcripts as labels for this, we used forced alignment based on HMM-GMM models in Kaldi.

Once the end-to-end model can do utterance-level speech recognition, then we take on the more complex task of producing the abstractive summary. The model that is pre-trained is then fine-tuned to do abstractive speech summarization.

5.5 Global Acoustic Context

5.5.1 The Challenge

End-to-end models consume the entire input at once, meaning that end-to-end speech summarization models take in the entire speech sequence at once, in contrast to cascade models where speech recognition can be done on isolated utterances as identified by acoustic segmentation. Since speech summarization models need to look over the entire input to identify what is important enough to summarize, end-to-end models need to be able to process the entire input sequence.

Different from other speech tasks like speech recognition, the input sequences for summarization are much longer. Table 5.1 shows the average and maximum lengths of the input and output sequences for speech summarization on the How2 dataset for video summarization. We can see from the table that there appears to be a 36x increase in input sequence length from speech recognition to summarization. This poses a significant challenge in modeling that we term *the global acoustic context problem*. Within this thesis, we examine this challenge in greater detail and develop multiple solutions to this challenge, one in this chapter, and more in Chapter 7.

Transformers are among the state of the art for many tasks in speech, natural language processing, and vision, among other fields involving ordered sequences of data. They achieve their exceptional generalization in part due to “multi-head self-attention” — processing blocks that derive (a set

TABLE 5.1: 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) are shown.

Set	**Max N**	**Mean N**	**Mean L**	**Max L**
Train ASR	4057	281.58	10.12	24
Train SSUM	145,082	9,806.58	60.54	173
Test SSUM	39,537	9,866.55	60.29	152

of) updated representations for each input in a sequence as a weighted sum of values derived from all the inputs in the sequence. This computation, however, has a computational and space complexity that is quadratic in the length of the input sequence. As a result, it becomes very challenging to operate on longer input sequences on modern GPUs and TPUs, and training such models from scratch is time and labor-intensive.

The computational challenges arise from the formulation of self-attention. For an input sequence of length N each head of the multi-head self-attention block derives three N-row matrices: a *value* matrix V representing the latent representation vectors for the N inputs in the sequence, a *query* matrix Q representing the probes with which each input derives the "self-attention" weights required to update itself, and a *key* matrix K, representing the key contribution of each input to the computation of its own weight in updating any input. The actual updated representations for the input computed by the head have the form $softmax(QK^\top)V$. The bottleneck arises from $softmax(QK^\top)$: both Q and K have N rows, $QK^\top$ requires $O(N^2)$ computation. This cannot be factored, since the softmax operates on the *product*, which must necessarily be computed before the softmax is applied.

To address the challenge of long input sequences, [Dai et al., 2019] uses segment-wise recurrence within transformer self-attention to provide longer context, and [Rae et al., 2020] compresses the segment-level contexts and provides them as additional input to enable a longer context. Reformer [Kitaev et al., 2020] uses Locality Sensitive Hashing to compute localized self-attention in O(n.logn), ETC [Ainslie et al., 2020] uses efficient global-local attention to scale to longer sequences. To reduce the complexity of self-attention to O(n), Linformer [Wang et al., 2020] uses a low-rank factorization of the self-attention matrix, and Big Bird [Zaheer et al., 2020] uses a combination of sliding window, global and random attention. Longformer [Beltagy et al., 2020] uses different attention patterns for each layer and restricted dilated self-attention with task-specific global attention. These long sequence techniques have been evaluated on text inputs, where the input sequence lengths are often several hundred times smaller than sequence lengths of video-level speech.

5.5.2 Our Solution: Restricted Self-Attention for Long Sequence Modeling

Consider the input speech sequence X of length N, which results in a summary token sequence Y with length L($N >> L$).

It is known that encoder self-attention has a computational complexity of $O(N^2)$, decoder self-attention has a complexity of $O(L^2)$, and encoder-decoder source-target attention has a complexity of $O(NL)$. Since $N >> L$, the encoder self-attentions are the largest contributors to computational complexity.

To make end-to-end training practical for summarization, the computational complexity of the encoder self-attention needs to be reduced since it forms the most significant contributor to computational cost. Inspired by [Beltagy et al., 2020, Moritz et al., 2021], we break down the self-attention computation into fixed-sized context windows of size W. For each sequence element, a surrounding context of width $W/2$ on each side is considered while computing the self-attention result. The number of such windows required will be $P = N/W$, and the cost of the encoder-self attention is now reduced to $O(PW^2) = O(NW)$, which is smaller than $O(N^2)$. To further reduce the computational complexity, we can drop one element in D elements, i.e., use dilation. Dilation further reduces the complexity to $O(P(NW/D))$.

Due to constraints of memory arising from computing, we consider only the first 100s of audio input, which is reasonable as it is the mean input duration for the How2 data (From Table 5.1, 10,000 input frames corresponds to 100s of input).

5.6 Experimental Setup and Model Hyperparameters

5.6.1 Dataset and Evaluation

The How-2 Dataset [Sanabria et al., 2018] contains 2000h of instructional videos with corresponding text transcripts, video, speech, translations, and summaries. Summarization is evaluated using standard metrics ROUGE [Lin, 2004b], METEOR [Denkowski and Lavie, 2014], and BERTScore [Zhang et al., 2020].

5.6.2 Models compared

In this work, we consider three different classes of approaches - *toplines*,*cascade models*, and our *end-to-end model*.

Toplines: Topline approaches constitute an upper bound on performance within the cascade. Assuming that the first block in the cascade, i.e., speech recognition has no errors, the toplines obtain the best possible summarization performance. We use BART [Lewis, 2020] as the pre-trained text summarization model and also assess the role of model size on summarization performance. Since BART is not pre-trained for video summarization, there exists a domain mismatch between the pre-training and the intended task. Therefore, we fine-tuned the BART models `BART-base` and `BART-large` on pairs of (reference transcript, reference summary) from the How2 dataset and then evaluated the resulting model on the test set. Consistent with training, we use the reference transcript during testing.

Cascade Models: We consider multiple cascade approaches to compare the end-to-end model against. We first train an in-domain speech recognition model on the How2 dataset. This model obtains a Word Error Rate (WER) of 9.3 on the How2 test set. The first component of our cascade models includes this in-domain speech recognizer.

Text summarization models are pre-trained on written text, which does not have ASR errors. This also constitutes a domain shift, and so in this case, rather than fine-tuning on pairs of (reference transcript, reference summary), we instead fine-tune with pairs of (speech recognition transcript, reference summary). This increases the robustness of the text summarization model to errors in speech recognition.

Our cascade models for speech summarization are comprised of the in-domain speech recognition model and the fine-tuned `BART-base` and `BART-large` models. We also compare to cascades used in other work:

1. A combination of recurrent sequence model for speech recognition and attention-based recurrent sequence model for text summarization [Palaskar et al., 2019b]
2. A combination of speech recognizer and text summarization model whose input is obtained by combining multiple speech recognition transcripts [Kano et al., 2021c]
3. A three-module cascade comprising a Kaldi-based [Povey et al., 2011] speech recognizer, a semantic concept predictor and a text summarization model that can transform semantic concepts into an abstractive speech summary.

End-to-end model: Our end-to-end model comprises a conformer-based speech encoder with restricted self-attention and a transformer-based text decoder. It is first pre-trained on utterance-level speech recognition and fine-tuned on abstractive speech summarization.

5.6.3 Model Hyperparameters

ESPNet [et. al., 2018] is used for speech model training. Our conformer encoder uses 2-fold convolutional subsampling followed by 12 encoder layers with feed-forward dimensions 2048 and 8 attention heads. The transformer decoder has 6 layers with feed-forward dimension 512 and 4 attention heads. ASR models are trained with joint Connectionist Temporal Classification (CTC)-Attention [Kim et al., 2017] with the weight for CTC training set to 0.3. The videos are trimmed to 100s for the video-level speech tasks owing to compute constraints. Specaugment [Park et al., 2019a] is used during model training and fine-tuning. We use 40-dimensional filter-bank and 3-dimensional pitch features for training all models. The Huggingface transformers library [Wolf, 2020] is used to fine-tune text-only cascade models. `BART-large` and `BART-base` [Lewis, 2020] are fine-tuned on How2 transcript and summaries.

TABLE 5.2: Word Error Rate (WER) (%) for Test and Held Test sets of the 2000h How-to Corpus. Window Size of 20 is used for Restricted Self-Attention

Encoder	**Decoder**	**Test WER (%)**
Transformer	Transformer	10.2
Conformer	Transformer	**9.1**
+ Restricted Self-Attention	Transformer	9.3

5.7 Experimental Results

5.7.1 Speech Recognition

Table 5.2 compares the performance of different architectures for speech recognition on the How2 dataset. We see that a conformer-based speech recognizer obtains a better performance compared to a transformer-based speech recognizer. We also note that using restricted self-attention as opposed to the standard multi-head self-attention proposed in [Vaswani et al., 2017b] results in comparable performance.

5.7.2 Cascade versus End-to-end models

TABLE 5.3: Summarization Performance of Topline, Cascade, and E2E Models using automatic (ROUGE and METEOR) and semantic evaluation metrics (BERTScore).

	Model	**Parameters**	**ROUGE-L**	**METEOR**	**BERTScore**
Topline	Groundtruth Text				
	+ `BART-large` Summarization	400M	55.5	30.0	**91.0**
	+ `BART-base` Summarization	140M	53.7	27.7	90.7
Cascade	Conformer ASR	107M			
	+ `BART-large` Summarization	400M	52.3	27.8	90.6
	+ `BART-base` Summarization	140M	50.3	25.6	90.3
	S2S- PredText2Summary [Palaskar et al., 2019b]	-	46.1	22.9	-
	ASR + BERTSum [Kano et al., 2021d]	-	48.2	-	-
	Kaldi ASR + Concept2Summary [Palaskar et al., 2021a]	-	51.4	**30.4**	-
E2E	Conformer Encoder				
	+ Transformer Decoder	**104M**	**56.10**	29.3	**91.53**

Table 5.3 highlights summarization results on three types of models: topline models, Cascade models, and end-to-end (E2E) models.

Size of Text Summarization Model across Cascade and Toplines: Of the text summarization models `BART-large` and `BART-base`, `BART-large` outperforms `BART-base` in ROUGE, METEOR, and BERT Scores, showing that larger models can perform better. Similar trends are seen across these model sizes within our cascade models.

Our Cascades versus prior work: Our in-domain conformer-based speech recognizer coupled with BART leads to strong cascade models that outperform previous work.

E2E versus Cascades: The E2E model outperforms the best cascade model on all metrics with 4x fewer parameters, indicating that the end-to-end model can produce more fluent, semantically relevant summaries. The difference in METEOR between our models is correlated with the difference in ROUGE-L scores. METEOR scores are content-based, and missing out on key noun phrases lowers the METEOR scores. From Table 5.3, it is clear that the cascade model and E2E models have lower METEOR Scores than the cascade concept Model and the Ground-truth models as the latter are better at retaining these noun phrases.

5.7.3 Ablation: Window Size and Dilation

TABLE 5.4: Effect of Window Size and Dilation in Self-Attention of the Speech Encoder on E2E Summarization Model Training. W is the Window Size, and D is the dilation factor

W	D	ROUGE-L	METEOR	BERTScore
20	✗	52.0	26.5	90.5
40	✗	**53.1**	**27.3**	**90.6**
60	✗	52.5	27.1	90.5
100	5	51.9	26.3	90.5

To understand the impact of context window size on summarization performance, we train models with different window sizes using a subset of the training data. From Table 5.4, a window size of $W = 40$ seems to yield the best ROUGE-L scores, while a smaller window of $W = 20$ yields a lower ROUGE-L score. An optimal window size is neither too short nor too long. Short windows lose important context, while longer windows incorporate less relevant context.

We also consider whether dilation can be used to improve the computational complexity while retaining comparable performance. Comparing the first and last rows, the first row uses a window size of 20 and no dilation, while the last row uses a window size of 100 with a dilation factor of 5. Both of these entries have comparable performance even though the row with the larger window size can learn more long-term correlations efficiently. This shows that dilation can enable the use of longer contexts with a lower computational complexity.

5.7.4 Ablation: Qualitative Examples

Apart from obtaining quantitative improvements as seen in Table 5.3, we manually inspect summaries produced by our best cascade and end-to-end models on the How2 test set.

Table 5.5 demonstrates two kinds of errors that we attribute to the cascade effect — missing content words(in blue), and mistranscribed words(in red). The proposed E2E approach mitigates the impact of these two types of errors, improving ROUGE and METEOR scores.

TABLE 5.5: Qualitative analysis of errors in Cascade and E2E Approaches. Text in red shows examples of mistranscribed words/phrases while text in blue shows examples of missing words/ phrases

E2E	DEFENDING AGAINST A SELF-DEFENSE TECHNIQUE IS THE PRINCIPLE OF THE ATTACKER 'S ARM . LEARN HOW TO STRIKE AGAINST A SELF-DEFENSE IN THIS FREE VIDEO FROM AN INDUCTEE IN THE US MARTIAL ARTS HALL OF FAME.
Cascade	DEF OR DEFANGING THE SNAKE IS A SELF-DEFENSE TECHNIQUE THAT TAKES THE ATTACKER'S STRIKE OUT OF PLAY. DEFANG THE SNAKE WITH TIPS FROM A MARTIAL ARTS INSTRUCTOR IN THIS FREE VIDEO ON SELF DEFENSE.
Ground Truth	SELF DEFENSE TECHNIQUES MADE EASY ! LEARN HOW TO STRIKE AGAINST A PUNCH IN THIS FREE VIDEO FROM AN INDUCTEE IN THE US MARTIAL ARTS HALL OF FAME .

5.8 Chapter Conclusion

In this chapter, we begin by formulating an end-to-end model for abstractive speech summarization to address the shortcomings of the classical cascade approach. End-to-end modeling is challenging with many challenges being unique to the task of abstractive summarization. We focus on the challenges that inhibit the practical realization of end-to-end models, namely, abstract learning and global acoustic context.

The abstract learning problem results from the complex and indirect mapping between input speech and output summary and can be addressed by solving simpler problems first and progressing to more complex problems. In this spirit, the abstract learning challenge can be mitigated by first pre-training the end-to-end model on utterance-level speech recognition, and then fine-tuning the pre-trained model for abstractive speech summarization.

The global acoustic context problem is a product of very long input sequences that need to be processed to perform abstractive speech summarization. One solution to this problem is to replace the encoder self-attention with window-restricted self-attention which limits the amount of context considered in the self-attention computation.

We compare end-to-end modeling against the classical cascade approach and find that our approach demonstrates strong performance compared to previous approaches to speech summarization (cascaded pipeline models). We also demonstrate the effects of various window sizes and dilations on summarization, concluding that optimal window sizes are neither too long nor too short, and that dilations can be used to look over larger contexts efficiently. Qualitative evaluation of summaries reveals that end-to-end models outperform cascades due to two types of errors - mistranscriptions and missing transcriptions.

All the observations and experiments in this chapter focused on an application of summarization, namely, video summarization, and concluded that end-to-end models outperform cascade approaches. In the next chapter, we assess the generalizability of this conclusion by benchmarking end-to-end and cascade models on multiple datasets. To do this, we introduce two novel datasets for speech summarization, and analyse when end-to-end models can outperform cascade models.

By restricting the scope of self-attention, every speech frame is not connected to every other frame within the encoder self-attention. Therefore, the question of whether we are truly learning global context remains. To be able to use the entire sequence to compute the self-attention without restriction, and truly learn long-term dependencies, we would need to examine other alternatives to restricted self-attention. In the next chapter and beyond, we examine other alternatives to restricted self-attention.

Chapter 6

Introducing the Multi-domain Benchmark for Speech Summarization

6.1 The Challenge of Data for Speech Summarization

Spoken language understanding (SLU) tasks involve inferring the linguistic structure or semantic meaning of a speech signal beyond its text transcript. We use this term broadly to include any natural language processing (NLP) task applied to speech, and tasks that involve linguistic understanding but also localization in the signal of relevant segments or producing speech as output. SLU has been an active area throughout the history of speech research [Hemphill et al., 1990, Calhoun et al., 2010, Busso et al., 2008, Zadeh et al., 2018, Chen et al., 2020a, Cohn et al., 2019, Yadav et al., 2020, Martinez-Lucas et al., 2020]. However, compared to "lower-level" tasks like automatic speech recognition (ASR) and speaker identification, SLU has received much less attention and resources, and specifically there are much fewer benchmarks with freely available data.

In particular, the SLU task of speech summarization finds application in many fields ranging from call centers to clinical visits. Automatic methods for speech summarization have many applications — generating keywords and descriptions from videos, creating talk titles and abstracts, producing concise summaries of medical visits [Espejel, 2019], and notes from a lecture, to name a few. There is a need for labeled data to build and benchmark automatic speech summarizers for all these applications.

First, we consider speech summarization for monologue summarization. Most prior work in speech summarization [Sharma et al., 2022a, Palaskar et al., 2019b, Shon et al., 2023, Kano et al., 2021b] focuses on summarizing monologue-style speech in how-to videos from the How2 [Sanabria et al., 2018] dataset. However, abstractive summarization on this dataset is relatively simple due to shorter videos and formatted reference summaries. Further, all of the raw audio for

TABLE 6.1: Statistics of the publicly available corpora for speech summarization. The nature of input speech, number of recordings, number of hours of audio, average recording length in minutes, the average number of words in the summary (transcript), input source, percentage of non-stop novel words in the summary that do not appear in the transcript are shown for each corpus, and word compression ratio. Speech is Monologue (Mono), Conversational (Conv), or Synthetic (Synth). Word compression ratio is computed as the ratio of the number of words in the transcript to summary, and averaged across the dataset.

Corpus	Speech	#Recs	#Hrs	Avg. Mins	Avg. #Words	Input Source	N. Words (%)	Comp. Ratio
How2	Mono.	72,981	2,000	1.64	32.79 (229.52)	How-to videos	47.55	7.53
Our SLUE - TED	Mono.	4,233	830.9	11.77	67.27 (1,749.44)	TED talks	55.27	28.95
AMI	Conv.	137	100	29.41	291.16 (5,841.73)	Meetings	47.9	18.48
ICSI	Conv.	59	70	60	534 (10,189)	Meetings	-	19.08
Spoken Gigaword	Synth.	328,880	-	-	31.4 (8.3)	Article Headlines	55.3	3.78
Ours - Interview	Conv.	49,411	4,354	5.28	40.12 (886.37)	Radio Interviews	47.79	22.9

this corpus is not publicly available due to users deleting content from YouTube[1]. Recent work introduced the TEDSummary corpus [Kano et al., 2021d] for speech summarization to address these challenges, however, the lack of information about the talks used in their corpus makes it difficult to reproduce their data selection. To advance the state of monologue summarization, we introduce *SLUE-TED*, a corpus containing around 824 hours of real speech from TED talks. The abstractive summarization task here is to predict the summary and title of the TED talk.

Different from monologue summarization, we consider other domains of application. A recent work [Huang et al., 2022] introduces the Spoken Gigaword corpus that can be used to predict article titles based on the first sentence. However, this dataset contains no real speech and is simpler due to the shorter and synthetic input recordings. There are two public corpora (see Table 1) that can be used for multi-party meeting summarization, such as AMI [McCowan et al., 2005] and ICSI [Janin et al., 2003]. However, these corpora are very small and are confined to certain topics and domains. To improve the breadth of corpora for speech summarization and advance multi-party meeting summarization, we introduce *Interview*, the largest-to-date open-domain corpus comprising 4,354h of multi-party radio interviews. *Interview* provides speech, transcripts, and short abstractive summaries of radio interviews with spontaneous speech that can be used for academic research.

In the following sections, we introduce the two new corpora, *SLUE-TED* and *Interview*.

6.2 Introducing SLUE-TED

Recent work introduced the TEDSummary corpus [Kano et al., 2021d] for speech summarization to address these challenges, and based on the crawler, and more recent talks released on the TED website[2], we introduce SLUE-TED, a re-designed corpus of summaries for TED Talks spanning the years until 2022.

[1] https://youtube.com

[2] CC BY–NC–ND 4.0 license

Statistics of our SLUE-TED can be found in Table 6.1. Recordings have an average length of 11.77 minutes and abstractive summaries are 67 words long on average. We find that, on average, nearly 66% of words in the title and 57.4% of words in the abstract are present in the transcript, suggesting that ASR pre-training can be useful in improving speech summarization performance.

For benchmarking and evaluation, we randomly split this corpus into 80% finetune, 10% validation, and 10% test set as shown in Table 6.2.

TABLE 6.2: SLUE-TED data split

	utterances	duration (h)
finetune	3384	664
dev	425	81
test	424	84

6.3 Introducing the Interview Corpus

To address the dearth of labeled speech summarization corpora for multiparty open-domain dialogue, we introduce the Interview corpus. National Public Radio (NPR) hosts shows that are generally 5-10 minutes long and interviews between hosts and guests. The Interview corpus for language modeling was introduced in [Majumder et al., 2020], where transcripts with speaker attribution were published for 105k different two-party and multi-party interactions, however, this data did not contain any summary annotations or speech. The Mediasum corpus [Zhu et al., 2021] for abstractive text summarization contains interviews from NPR and CNN along with corresponding transcript and summary. Summaries are basically the description text provided for interview recordings. However, this corpus does not contain any audio or links to download audio. We used the NPR API to search for paired audio and textual summaries and curated the Interview corpus for speech summarization.

Interview is the largest to-date open-domain public corpus for speech summarization with a total duration of 4,354h. Table 6.1 compares the statistics of the Interview data to other existing datasets. Speech recordings contain spontaneous speech along with real audio noises that represent music, and events inside and outside the studio corresponding to discussion topics. The dataset has 49.4k recordings that span multiple topics and domains, with an average recording length of 5.28 minutes. The transcription on average contains 886.4 words and the summary contains 40.1 words on average, representing a compression ratio of 22.09 between the transcript and summary. Around 48% of words in the summary are not present in the transcription, making this a relatively challenging task. On average, each recording has 24 turns and 4 speakers. Audio download links are available through the repository[3] for personal non-commercial and research use as per NPR's license terms[4] along with transcriptions and abstractive summaries. The corpus can be used for speech recognition and speech summarization.

[3] https://github.com/roshansh-cmu/Interview-corpus-SSUM

[4] https://www.npr.org/about-npr/179881519/rights-and-permissions-information

6.4 Review: Fourier Transform based self-attentions

FNet [Lee-Thorp et al., 2022a] is a recently proposed approach to reduce the computation of the attention module. It has achieved performance approaching that of models such as Bidirectional Encoder Representations from Transformers [Devlin et al., 2019b] in the language modeling tasks. We proposed introducing it to replace the attention in the speech encoder of our dual speech summarization system.

FNet interprets the Q, K, and V's dot product as a convolution representation, and the FNet changes the Transformer's self-attention function as $\text{Att}(Q) = F_{\text{seq}}(F_{\text{dim}}(Q))$. Here, F_{dim} and F_{seq} denote the Fourier transform of the hidden dimension direction and time direction. The computation complexity of FNet is $\mathcal{O}(DL \log L) + \mathcal{O}(DL \log D)$. In general, we have $L \gg D$; thus, in this work, we omit the complexity of $\mathcal{O}(DL \log D)$ for simplicity.

The advantage of this replacement is that, in addition to reducing the computational complexity to the log order of the sequence length, it saves many resources compared to other methods since the conventional query, key, and value projection matrices are no longer needed. In addition, it continues to be able to view the entire series. However, since the Fourier transform has no learnable parameters, it may have weaker modeling capabilities than other attention modules.

6.5 Benchmarking E2E and Cascade Models across multiple domains

Speech summarization can be addressed using cascade or end-to-end approaches, and depending on the task and the application domain, different approaches may perform better. In this chapter, we investigate the strengths and weaknesses of these approaches across a variety of application domains to conclude their effectiveness. Cascade approaches [Kano et al., 2021b, Palaskar et al., 2021a, Yu et al., 2021, Palaskar et al., 2019b] summarize speech in the textual domain, by first transcribing speech, and then using text summarization models on the transcript. While such cascades can effectively leverage strong pre-trained ASR [Radford et al., 2022, Chan et al., 2021, Chen et al., 2022] and text summarization [Raffel et al., 2020b, Lewis, 2020] models, they are prone to error propagation and have slower inference. Furthermore, cascade systems cannot effectively leverage speech properties like prosody [Chen et al., 2020b, Jurafsky et al., 1998, Tran et al., 2018] that are not represented in the transcript. To address this, end-to-end models [Sharma et al., 2022a, Kano et al., 2023a, Matsuura et al., 2023b, Matsuura et al., 2023a, Sharma et al., 2023b] learn to directly map between speech features and a textual summary, and have outperformed cascade models on abstractive video summarization. Due to the difference in corpus size, recording length, nature of speech, and application domain, these observations may not hold true for all corpora. In this chapter, we conduct a comprehensive evaluation of cascade and end-to-end models on abstractive summarization tasks involving 4 different corpora and find that cascade models outperform end-to-end models when the input recording is very long and when the corpus is smaller. Through this evaluation, we also study the impact of various design choices on the performance of end-to-end summarization models – including the ASR pre-training, long-term context modeling, and the use of large-scale pre-trained speech models like Whisper [Radford et al., 2022] as a backbone. Experiments demonstrate the need for strong pre-training, the scope for better long-term context modeling, and the limited utility of Whisper-style pre-training. Prior work [Shon et al., 2023, Palaskar et al., 2021a, Sharma et al., 2022a] has focused on building speech summarization models for specific application domains. Since there exist multiple datasets for the task, the question arises – Can we use other datasets to improve low-resource summarization? Experiments show the benefits of multi-style training over fine-tuning for transferring knowledge between Interview and SLUE-TED.

6.5.1 Approach and Models

6.5.1.1 Cascade Models

Model cascades have historically been the *de facto* approach to speech summarization [Zhu et al., 2020, Rezazadegan et al., 2020, Manakul et al., 2020, Kano et al., 2021b]. The cascade is generally composed of an ASR model that first maps the input speech to a textual transcription. This is followed by an Abstractive Text Summarization (ATS) that generates a summary from the transcribed speech. The challenge of global context can be addressed by using either utterance-level or long-form speech transcription (e.g. Whisper), followed by transcript summarization. In the former case, audio segmentation is used to obtain shorter utterances that can be independently transcribed using an utterance-level ASR model and then concatenated to obtain the entire transcript. To best leverage the strengths of the cascade approach, we assemble our cascade with two powerful pre-trained models: Whisper [Radford et al., 2022] for ASR and T5 [Raffel et al., 2020b] for ATS. We note that Whisper ASR is used in a zero-shot manner while T5 is fine-tuned on the dataset in question. To evaluate the effect of ASR transcription errors on downstream performance, we also perform experiments where the ground truth ASR text is used as the input to T5.

Whisper: Whisper is a Transformer-based [Vaswani et al., 2017b] sequence model trained on 680k hours of crawled web audio. Model training was performed on 30-second chunks, allowing the transcription of long-form audio using predictions of prior chunks as additional context.

T5: The Text-to-Text Transfer Transformer (T5) is a Transformer-based sequence model that was pre-trained using span corruption task, where regions of the input are randomly replaced with a single span token, on the C4 dataset [Raffel et al., 2020b]. The model is thus pre-trained to use the unmasked regions to generate the masked content. T5 is then fine-tuned on a variety of tasks, including text summarization, by pre-pending a task token as part of the label sequence.

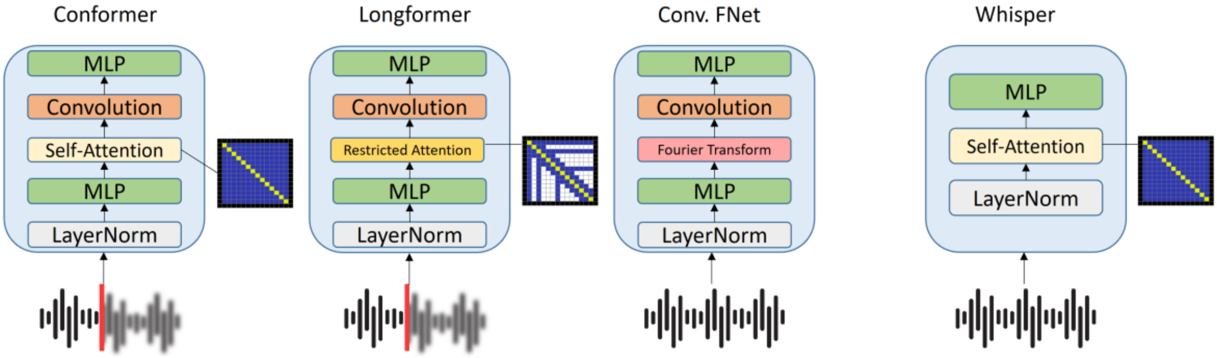


FIGURE 6.1: Comparison of end-to-end architectures. The Conformer and Longformer require the input speech to be truncated to 100 seconds. Whisper uses only 30s of the input context, while FNet can utilize the full acoustic context, due to the latter's higher computational efficiency. The squares next to the model architectures in the figure represent the typical attention maps computed by each model, with the white squares denoting points where the attention is not computed to save memory.

6.5.1.2 End-to-End Models

To address the weaknesses of the cascade approach, end-to-end formulations of speech summarization have been popularized in recent years [Kano et al., 2023b, Matsuura et al., 2023a, Matsuura et al., 2023c, Sharma and Raj, 2022, Sharma et al., 2022a, Shon et al., 2022]. These models generate summaries directly from the input speech, allowing them to leverage para-linguistic information and avoid propagating ASR errors.

However, the non-monotonic and complex mapping between the input speech and output summary makes training end-to-end models difficult since there are likely words in the output summary that are not directly present in the input speech. [Sharma et al., 2022a] addresses this by using two-stage training – first the sequence model is pre-trained for utterance-level ASR, and then, the trained ASR model is fine-tuned for speech summarization over longer sequences. [Kano et al., 2023a] uses an auxiliary encoder with transcripts to better learn this complex mapping while [Matsuura et al., 2023a] initializes the decoder with a strong generative pre-trained language model.

Modeling global input context is challenging for many end-to-end approaches due to the quadratic memory complexity of attention in Transformer-based [Vaswani et al., 2017b] models. For example, a prior study on speech summarization [Kano et al., 2023b] found that a standard Conformer [Gulati et al., 2020b] architecture could only fit up to 100 seconds of input speech on an 80 GB A-100, a far cry from the typical input lengths of 10-20 minutes. As such, recent work in speech summarization has placed a strong focus on developing methods that can more efficiently scale to longer inputs [Kano et al., 2023b, Sharma et al., 2022a, Sharma and Raj, 2022].

We benchmark several end-to-end architectures that have been proposed for speech summarization on each dataset. The Conformer model is among the state-of-the-art for speech recognition and can be used for speech summarization with truncated inputs. We then experiment with architectures that address the memory issues of the Conformer, such as by using restricted attention or no attention. Finally, we evaluate the efficacy of fine-tuning large-scale pre-trained ASR models like Whisper for speech summarization. An overview of each architecture is presented in Figure 6.1.

Conformer: The Conformer [Gulati et al., 2020b] is a convolution-augmented Transformer [Vaswani et al., 2017b]. It consists of a vanilla multi-head attention block followed by a convolution module, with both sandwiched between two macaron-like [Lu* et al., 2019] feed-forward networks. Due to the ($O(n^2)$) quadratic memory complexity of the attention block in terms of the sequence length n, the input speech needs to be truncated to fit within GPU memory. In our experiments, we truncate to a maximum length of 100 seconds.

Restricted Attention - Longformer Longformer [Beltagy et al., 2020] restricts attention computation to a fixed window around every point, with an option to dilate the window w by a factor of d, allowing for longer context without increased computation. Its memory complexity with dilation is thus $O(n \times w/d^2)$. We use the setup proposed in [Sharma et al., 2022a], with a window size of 60.

Alternatives to Attention - FNet: Rather than attempting to approximate the attention mechanism, some studies in long-form sequence modeling have opted to forgo it entirely [Kano et al., 2023b, Lee-Thorp et al., 2022a, Khalitov et al., 2023]. We experiment with the FNet [Lee-Thorp et al., 2022a] architecture, which has been shown to work well in speech summarization [Kano et al., 2023b]. FNet replaces attention with a parameter-less Fourier transform, which makes the computational complexity linear in sequence length: $O(n \log n)$. Similar to our Longformer experiments, we use a combination of Conformer and FNet that was proposed for speech summarization [Kano et al., 2023a]: FNet's Fourier transform replaces the attention block.

Whisper Fine-tuning: Whisper is fine-tuned for speech summarization. For this purpose, a new "⟨|SUMMARIZE|⟩" task tag is appended to the vocabulary of multilingual `whisper-base`. The model then predicts summaries based on 30 seconds of input context in the format
“⟨|en|⟩⟨|SUMMARIZE|⟩⟨|notimestamps|⟩“

6.5.2 Benchmark Results

Comparing end-to-end and cascade models

How2: Table 6.3 compares various end-to-end and cascade models for speech summarization. [Kano et al., 2023a] uses a speech and transcript encoder to effectively and accurately model long-term context, while [Matsuura et al., 2023a] uses transfer fine-tuning from a pre-trained language model and speech summarization model. [Matsuura et al., 2023b] uses synthetic data augmentation, and hence is not directly comparable to our FNet baseline. Whisper fine-tuning performs worse than the Conformer with the same amount of context(30s), showing that Whisper pre-training is likely not very helpful for speech summarization. The end-to-end FNet(full) model which utilizes the entire input context outperforms Whisper+T5 cascade for video summarization.

TABLE 6.3: Summarization Performance of different end-to-end summarization models using approximate attention mechanisms and truncated inputs on the How2 Corpus

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Longformer [Sharma et al., 2022a]	60.7	44.9	56.1	29.3	91.53
FNet (100s)	61.20	41.60	58.00	28.30	92.80
Conformer (100s)	64.70	46.70	62.30	30.40	93.00
FNet (full)	66.90	48.70	64.00	32.70	93.50
Dual Encoder [Kano et al., 2023a]	64.9	46.5	61.9	32.0	-
Pre-trained LM [Matsuura et al., 2023a]	67.0	52.1	63.2	34.4	93.98
Synthetic Data [Matsuura et al., 2023b]	68.40	54.10	65.00	34.90	93.80
Whisper FT (30s)	57.67	43.45	54.40	28.03	88.49
Conformer (30s)	63.30	47.58	59.16	31.76	92.08
Whisper+ T5-Base	63.16	44.55	57.45	30.98	91.53
GT + T5-Base	62.14	43.45	56.41	30.49	91.41

TABLE 6.4: Summarization Performance of different end-to-end summarization models using FNet for different input lengths on the How2 Corpus

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTSc
FNet (30s)	60.00	40.80	57.00	27.50	92.60
FNet (60s)	61.40	42.50	58.50	28.50	92.90
FNet (100s)	61.20	41.60	58.00	28.30	92.80
FNet (FULL)	66.90	48.70	64.00	32.70	93.50

TABLE 6.5: Summarization Performance of different end-to-end summarization models using approximate attention mechanisms and truncated inputs on the SLUE-TED Corpus

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Whisper FT (30s)	23.96	5.43	16.80	11.99	85.11
Conformer (100s)	22.00	5.20	15.60	7.10	83.50
Conformer (600s) [Shon et al., 2023]	23.8	5.1	16.3	11.7	84.0
FNet (full)	23.60	5.60	16.70	8.00	83.10
Whisper+ T5-Base	28.11	6.68	18.22	12.62	87.09
GT + T5-Base	30.02	8.08	19.64	13.41	87.35

TABLE 6.6: Summarization Performance of different end-to-end summarization models using approximate attention mechanisms and truncated inputs on the INTERVIEW Corpus

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Conformer(100s)	33.65	13.75	25.05	15.73	87.39
FNet(full)	34.81	23.05	31.54	13.63	91.54
Whisper+T5-Base	38.13	17.70	28.22	17.80	89.70
GT+T5-Base	39.46	19.20	29.51	18.66	90.13

TABLE 6.7: Summarization Performance of different end-to-end summarization models using approximate attention mechanisms and truncated inputs on the AMI Corpus

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
FNet (full)	17.14	6.55	11.68	6.19	70.31
Whisper+ T5-Base	35.03	10.519	18.957	8.64	76.97
GT + T5-Base [Manakul et al., 2020]	36.83	11.98	23.77	-	-

TABLE 6.8: Summarization Performance of different FNet-based end-to-end models using different ASR initializations on How2

Initialization	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Fully Trained ASR	66.90	48.70	64.00	32.70	93.50
ASR Epoch 10	42.60	22.60	39.50	38.20	88.70
Scratch	57.10	40.50	53.70	30.40	91.00

TABLE 6.9: Example model outputs when using varying amounts of ASR pre-training.

Source	Output
Reference	LOOKING FOR NEW LIP COLOR IDEAS ? GET TIPS FOR APPLYING LIP COLOR TO DRAMATIC MAKEUP IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST
Fully Trained ASR	LOOKING FOR DRAMATIC EYE MAKEUP ? GET TIPS FOR APPLYING SHEER LIP GLOSS IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL COSMETOLOGIST
ASR Epoch 10	HOW TO APPLY EYE CREASE MAKEUP FOR CREATING AN 80S MAKEUP LOOK ; GET EXPERT TIPS AND ADVICE ON 1980S MAKEUP TRENDS AND HOW TO APPLY THEM IN THIS FREE INSTRUCTIONAL VIDEO
Scratch	LINING THE INSIDE OF YOUR EYE IS A DRAMATIC WAY TO ENHANCE YOUR EYE MAKEUP LEARN HOW TO LINE THE INSIDE OF YOUR EYE IN THIS FREE VIDEO CLIP FROM A PROFESSIONAL MAKEUP ARTIST

SLUE-TED: Table 6.5 reports the results of various end-to-end and cascade models. It is noted that strong pre-training using Whisper is competitive, likely because there is a lot of useful information in the first 30 seconds of a TED talk. Cascade Whisper+T5 outperforms the end-to-end Fnet (full) in this case. We hypothesize that this is because large-scale text summarization pre-training likely helps with the more challenging task [5].

Interview: Table 6.6 highlights the performance of various end-to-end and cascade models. The FNet end-to-end model outperforms the cascade on all metrics except METEOR.

AMI: From Table 6.7, due to the small dataset size used for fine-tuning the end-to-end and cascade models, the pre-trained text summarization model outperforms the end-to-end FNet model.

In conclusion, for smaller datasets (SLUE-TED and AMI), cascade models outperform the end-to-end model, while for the larger datasets (Interview and How2), end-to-end models outperform the cascade. For smaller datasets, large-scale pre-training of the text summarization model is likely more helpful, making the cascade model more powerful than an end-to-end model that does not use such pre-training. Further, the smaller datasets (AMI, SLUE-TED) have longer inputs than the larger datasets, and the performance gap between standard self-attention and approximations is significant as mentioned above. Methods that can better handle long-term context and use strong language pre-training [Matsuura et al., 2023a] may enable end-to-end models to be competitive with cascade models for meeting summarization and TED talk abstract generation.

Long Context Modeling: Table 6.3 reports performance on the How2 corpus for different end-to-end models. First, we compare strategies to handle long-term context. Comparing the Longformer, FNet model, and Conformer with context 100s, we observe a significant gap in performance between standard attention and the other two approximate attention mechanisms. Table 6.4 compares the performance of speech summarization models trained on How2 with different input lengths. Performance generally increases with the length of input context, demonstrating the necessity of long input context modeling.

[5] This is based on more novel words in the summary compared to How2 or Interview (55.27 % versus $\tilde{4}$7.5,47.8 %)

Reliance on ASR Pre-training: Prior work [Sharma et al., 2022a, Kano et al., 2023a, Matsuura et al., 2023b] has used a two-stage training strategy that involves ASR pre-training followed by summarization fine-tuning. To verify the necessity of strong initialization for end-to-end summarization, Table 6.8 compares the performance of speech summarization models with different initialization. A model initialized from a fully trained ASR outperforms models that are initialized with partially trained ASR and models that use no pre-training. This demonstrates the benefit of two-stage training for speech summarization. Interestingly, we observe that models with ASR pre-training tend to deviate more in topic and style from the reference summary (Table 6.9).

6.5.3 Multi-style and Transfer Learning

The presence of multiple datasets for a task provides means to utilize information from other tasks to improve performance on all or some of the tasks. Multi-style training, which involves augmenting the training set with data from other corpora has been used to confer noise robustness and better generalizability in speech recognition [Chen et al., 2022, Lippmann et al., 1987]. However, it is unclear whether multi-style training can benefit speech summarization.

Another method to utilize multiple datasets and models involves transfer learning. Prior work [Matsuura et al., 2023a] has found that initializing the encoder from a previous summarization model, and decoder from a pre-trained language model is beneficial. With pre-trained summarization models that are trained on other corpora, fine-tuning can be used to transfer knowledge across multiple speech summarization tasks. We compare fine-tuning-based transfer learning with multi-style training for speech summarization.

TABLE 6.10: Transferring knowledge across datasets for speech summarization. INT and TED represent the Interview and SLUE-TED respectively. X+Y represents multi-style training while X→Y represents fine-tuning.

Training Data	Test	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
TED*	TED	20.00	2.84	13.33	7.54	80.62
INT + TED	TED	23.53	4.22	15.81	10.09	84.83
INT → TED	TED	22.81	3.70	15.17	10.58	83.13

Experimental Results: Given multiple datasets, knowledge may be transferred across summarization tasks to improve model performance under low-resource settings. We compare a SLUE-TED baseline[6] using the Interview model and tokenization against multi-style training and fine-tuning using the SLUE-TED and Interview datasets. From Table 6.10, we note that multi-style training with Interview outperforms fine-tuning SLUE-TED a pre-trained Interview model on SLUE-TED.

[6] SLUE-TED baseline here uses the same architecture as the E2E model on Interview

6.6 Chapter Conclusion

In this chapter, we address the paucity of labeled data and standard benchmarks for the evaluation of speech summarization models.

First, we address the lack of labeled data for speech summarization by introducing two new open-domain public corpora - *SLUE-TED* and *Interview* corpus with 824h and 4,354h of real speech for abstractive speech summarization. The two corpora address the tasks of abstract and title prediction from TED talks and abstractive summarization of multi-party meetings respectively.

Then, we compare and contrast end-to-end and cascade approaches on a benchmark comprising 4 different abstractive speech summarization tasks. We find that on larger datasets with relatively short recordings, end-to-end models outperform cascades, while on smaller datasets, with much longer recordings, cascade models perform better. There exists a significant gap in performance between standard self-attention and approximations for end-to-end summarization, and it is important to address this challenge in future work. Further, two-stage training with ASR pre-training appears to be important to obtain relevant summaries. When using *Interview* data to improve speech summarization on *SLUE-TED*, we find that multi-style training slightly outperforms fine-tuning.

Based on the observations made in this chapter, developing methods for better long-context modeling is an important research direction to unlock the full potential of end-to-end models.

In the next part of this thesis, we examine ways to improve end-to-end models for speech summarization. Chapter 7 revisits the challenge of global acoustic context to make models stronger at capturing long-term context.

Part III

Improving Models: Long-term context and generalization

Chapter 7

Improving Modeling for Global Acoustic Context

In this chapter, we examine the challenge of imbuing end-to-end models for speech summarization with the ability to capture global acoustic context.

In Chapter 5, we introduced the use of restricted self-attentions for speech summarization, however, we questioned whether limiting the scope of attention allows models to truly learn global acoustic context. As a solution to this challenge, we examined Fourier transform-based FNet self-attentions in Chapter 6. Though FNet self-attentions obtain good performance and can model long-range context, we note that it has a computational complexity of $O(NlogN)$ where N is the length of the input sequence. The natural question is whether this can be improved upon. A class of self-attention approximations called linear transformers [Katharopoulos et al., 2020] use the kernel trick in conjunction with associativity of matrix products to make the self-attention scale linearly with the length N of the input sequence. However, existing linear transformers sacrifice performance at the altar of efficiency, and in a bid to bridge the gap, we introduce the XNORformer, a linear transformer with XNOR-based kernels. Experiments demonstrate that the proposed XNORformer outperforms the state-of-the-art linear transformer, i.e., the Cosformer [Qin et al., 2022].

In general, optimized self-attention can help models learn patterns over longer input contexts, but all optimized self-attentions, like standard self-attention, have capacity limits. This capacity limit depends on the model architecture and computing memory available, which we will discuss in greater detail within this chapter. What this means though for any input with a length N that is beyond the capacity C is that such an input is truncated to comprise only the first C frames. However, in practice, considering only the first C frames can be inimical to performance. Any frame(s) after the first C are not considered due to truncation, but, may contain important information that needs to be in the summary. However, there is no way for such information to be represented in the summary.

Therefore, an important challenge to address to enable the practical deployment of such end-to-end models is the ability to deal with 'arbitrarily long' sequences. In this chapter, we propose "BASS: Block-wise Adaptation for Speech Summarization" to facilitate the processing of arbitrarily long recordings. BASS reframes speech summarization as a streaming process where summaries are predicted based on blocks of input, and updated within subsequent blocks. Experiments show that BASS can process arbitrarily long inputs in a performant manner, however, the use of standard self-attentions within degrades the computational efficiency.

To improve the efficiency of BASS, we propose "R-BASS: Relevance-Aware Blockwise Adaptation for Speech Summarization" that considers only relevant blocks to update context and produce summaries. Experiments demonstrate that R-BASS can improve efficiency by over 86 % without significant degradation in performance.

7.1 Linear Self-Attention with Linear Transformer and Cosformer

The restricted self-attention approach introduced in Chapter 5 limits the effective acoustic context that is used by the speech encoder. Since such approaches limit the scope of self-attention, we examined the FNet-based self-attention in Chapter 6. However, the computational complexity of FNet [Lee-Thorp et al., 2022a] is $O(Nlog(N))$ for a sequence with input length N.

In this section, we consider an alternative, more efficient framework - linear self-attention and linear transformers whose complexity varies linearly with the input sequence length N.

The $O(N^2)$ complexity of softmax-based attention results from inner-product term $Q_i K_j^\top$ that occurs within the similarity term $S(Q_i, K_j) \sim \exp(Q_i K_j^\top)$ for the softmax and prevents factorization of the computation.

Linear transformers replace the exponent over inner products by a factored "product over Kernels" similarity measure $S(\mathcal{Q}, \mathcal{K})$ defined in Equation 7.1, where $\phi()$ is a "Kernel" function. Using the associative law of matrix multiplication permits the computation to be of the following form, which can be computed in linear time.

$$S(\mathcal{Q}, \mathcal{K}) = \phi(\mathcal{Q})\phi(\mathcal{K})^T \tag{7.1}$$

$$\mathcal{O}_i = \frac{\phi(Q_i)(\sum_{j=1}^{N} \phi(K_j)V_j)}{\sum_j \phi(Q_i)\phi(K_j)}. \tag{7.2}$$

Prior works have used Gaussian Kernels [Choromanski et al., 2021], ELU based kernels [Katharopoulos et al., 2020], and ReLU kernels [Qin et al., 2022]. Of these approaches, the Cosformer was shown to be the state-of-the-art approach.

Cosine self-attention [Qin et al., 2022] was introduced to derive a linear cost replacement of quadratic self-attention without approximating the softmax function. Using a ReLU kernel as a replacement for the softmax ensures the non-negativity of the attention scores. The authors further propose a cosine-based re-weighting scheme that stabilizes attention weights by amplifying local correlations. The proposed cosine re-weighting results in the similarity metric $S(\mathcal{Q}_i, \mathcal{K}_j) =$ ReLU$(\mathcal{Q}_i)$ReLU$(\mathcal{K}_j)$cos$(\frac{i-j}{N})$. Let $K_j^{sin} =$ ReLU$(K_j)sin(\frac{j\pi}{2N})$, K_j^{cos} = ReLU$(K_j)cos(\frac{j\pi}{2N})$, Q_i^{sin} = ReLU$(Q_i)sin(\frac{i\pi}{2N})$, and Q_i^{cos} = ReLU$(Q_i)sin(\frac{i\pi}{2N})$. The resulting output is then:

$$\mathcal{O}_i = \frac{\sum_{j=1}^{N} Q_i^{cos}((K_j^{cos})^T V_j) + \sum_{j=1}^{N} Q_i^{sin}((K_j^{sin})^T V_j)}{\sum_{j=1}^{N} Q_i^{cos}(K_j^{cos})^T + \sum_{j=1}^{N} Q_i^{sin}(K_j^{sin})^T} \tag{7.3}$$

The cosine re-weighting also functions as a relative positional embedding for the attention computation, and has been shown to improve performance on bidirectional fine-tuning tasks [Qin et al., 2022].

7.2 Our Proposal: the XNORformer

Though the cosformer achieves incredible performance on language modeling and the Long Range Arena benchmark, there is a considerable gap in performance between this self-attention and the standard multi-head self-attention when used for speech recognition on Librispeech-100. Therefore, it is desirable to develop a different solution with linear complexity and improved performance.

To do this, we develop a new kernel function which when integrated into the linear transformers framework yields improved performance. The closest kernel to our proposed approach is the softmax Kernel, which uses $\phi() = Sm()$. Though these succeed in reducing the computational overhead, they often result in undesirable performance reductions, compared to the full softmax-based attention. Further, we develop positional embeddings that can incorporate relative positional biases within the self-attention.

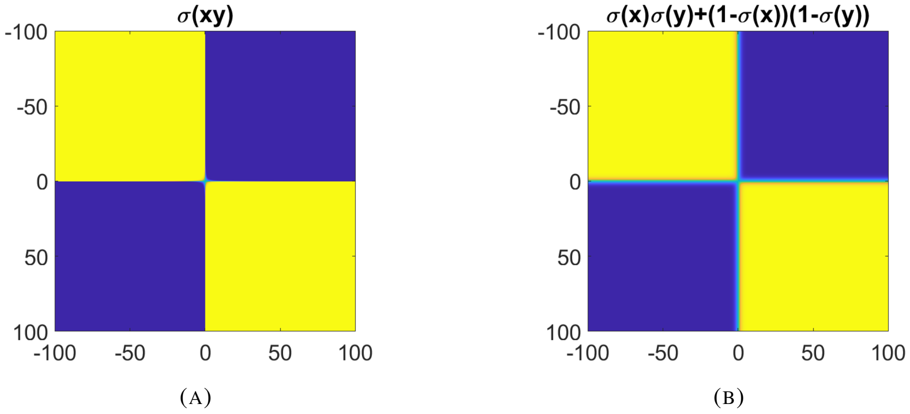
7.2.1 Examining the Sigmoid of the Product

Our objective is to close the gap between linear transformers and the full transformer formulation without losing the computational advantage of linear time. As mentioned earlier, the main impediment to linear-time factorization is the softmax over the product in the conventional transformer.

Our proposal builds on the following insight: consider the simplest version of a softmax, which is the sigmoid. $\sigma(xy)$, the sigmoid of the product of two variables x and y, has a distinct XNOR-like ($op\bar{l}us$-like) behavior as shown in Figure 7.1 a. This cannot be factored because the XNOR cannot be modeled by a linear boundary. However, analogously to the decomposition of the XNOR, $\sigma(xy)$ *can* be expressed as the sum of *two* bilinear terms:

$$\sigma(xy) \approx \sigma(x)\bar{\oplus}\sigma(y) = \sigma(x)\sigma(y) + (1 - \sigma(x))(1 - \sigma(y)) \tag{7.4}$$

Figure 7.1b shows the approximation $\sigma(x)\bar{\oplus}\sigma(y)$. Except for a narrow region near the axes, the error is minimal.



(A)

(B)

FIGURE 7.1: Approximating the Sigmoid of a Product using the XNOR of the Sigmoids(A) Left: $\sigma(xy)$ for $x, y \in (-100, 100)$. (B) Right: The approximation $\sigma(x)\bar{\oplus}\sigma(y)$.

This leads us to our formulation for the XNOR-former.

7.2.2 X-NOR Self-Attention

Drawing from the X-NOR approximation of the sigmoid, we now extend this to approximate the softmax product in Equation **??** as

$$S(Q_i, K_j) = w_1 Sm(Q_i)Sm(K_j) + w_2 Sm'(Q_i)Sm'(K_j), \tag{7.5}$$

where, as before, Sm represents the softmax operator and $Sm' = 1 - Sm$ represents its complement. The weights w_1 and w_2 account for the fact that the variable is of length N as opposed to binary.

This gives us the following XNOR factorization of self-attention, which allows it to be computed in linear time and space complexity.

7.2.3 Experimental Results

On End-to-End Speech Summarization, Table 7.1 highlights the results. We observe that the proposed W-XNOR self-attention-based model outperforms the Cosformer on a very long input sequence (10k) task without extensive hyperparameter tuning. The XNOR model has a better

TABLE 7.1: Results of our models on the speech summarization task for the How2-2000h data. Abstractive Summarization is evaluated using Rouge scores (ROUGE-1,ROUGE-2,ROUGE-L), METEOR scores for content, and BERTScore for semantic relevance. Higher numbers are better.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
Cosformer baseline	47.91	27.85	42.49	21.26	89.36
W-XNOR	**56.50**	**38.05**	**51.02**	**27.21**	**90.65**

performance in terms of the ROUGE-L, METEOR, and BERTScore metrics, which demonstrates that the proposed W-XNOR model produces more coherent, semantically relevant, content-rich summaries in comparison to the baseline.

7.3 The Trade-off between memory and performance

From our investigations in this chapter, and previously in Chapters 5, and 6, it is patently clear there exists a tradeoff between the computational complexity(and memory usage) and summarization performance obtained using different approaches to capture global acoustic context. The ideal solution would have the smallest memory usage with the highest performance.

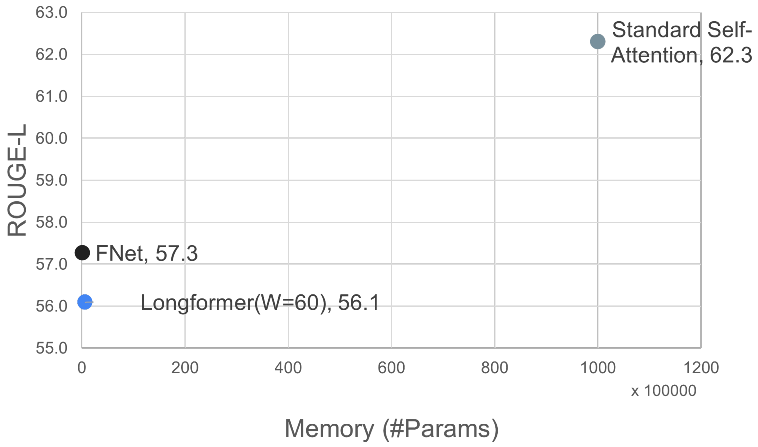


FIGURE 7.2: Plot of summarization performance in ROUGE-L over the How2 dataset for models that use the first 100 seconds of input context as a function

Figure 7.2 plots the performance of FNet-based, Longformer-based (restricted self-attention), and standard self-attention as a function of the amount of memory required in the number of float params to perform the self-attention operation. We note that the FNet performs better than the Longformer, but has higher complexity. The standard self-attention outperforms the optimized self-attention by a significant margin but has a much higher computational cost.

Next, we look at the challenge of dealing with arbitrarily long inputs for speech summarization.

7.4 The Challenge of Arbitrarily Long Inputs

End-to-end speech summarization models [Sharma et al., 2022b, Matsuura et al., 2023b] have been shown to outperform competitive cascade models that comprise speech recognition and text summarization modules. Such end-to-end models use very long speech sequences as input, and standard transformer models cannot handle very long inputs owing to the quadratic computational complexity of self-attention. Prior work has proposed restricting the scope of attention using the Longformer [Sharma et al., 2022b, Beltagy et al., 2020] or linear self-attentions like the XNOR-former [Sharma and Raj, 2022]. However, even with such optimizations, there remains an upper limit on the number of input frames that a given end-to-end model can consume with available computing infrastructure.

End-to-end models that consume the entire input at once have limitations on how much input context can be considered in making predictions. Not being able to consider the entire input context severely impedes summarization performance, as seen in Chapter 6. This underscores the importance of developing approaches to summarize arbitrarily long recordings.

One solution to this problem, which we have discussed thus far is the use of optimized self-attentions. Optimized self-attention can help models learn patterns over longer input contexts. But all self-attention mechanisms, optimized self-attentions have limits on the amount of input context that can be considered. For example, with a 6-layer conformer [Gulati et al., 2020b] encoder, a 32G V-100 GPU can take sequences of length 25,000; and with our XNOR-based encoder, the same GPU can take 45,000 frames. Any input speech sequence of length greater than this upper limit is truncated to be able to train and infer, and truncating inputs makes summarization less accurate since information is effectively removed from the input.

Another solution to this problem could involve using truncated inputs in training and using entire inputs during inference. However, attention-based sequence models, which are used to realize end-to-end speech summarization do not generalize well to input lengths that are different from those used in training [Deng et al., 2022], which makes it important to develop methods that work similarly during training and inference.

A third potential solution involves expanding the available capacity, i.e., shrinking model sizes and/or increasing the amount of computing resources. However, modifying the model architecture could have negative implications on model performance, and increasing the amount of computing resources is challenging and not sustainable in practice.

A fourth solution could be to use as model input only relevant portions of the speech input to the end-to-end model. In practice, this may not work since defining relevance and extracting relevant portions of a recording are incredibly challenging research problems. Furthermore, the relevant portions of the input recording may exceed the amount of input context that can be processed given capacity limits. This brings us back to square one and perhaps makes this solution infeasible.

In the remainder of this chapter, we develop two solutions to summarize arbitrarily long recordings.

7.5 BASS: Blockwise Adaptation for Speech Summarization

One way to circumvent the restrictions on arbitrarily long inputs is to develop methods that use the maximum permissible input context *at a time*.

To address the challenge of arbitrarily long input, a viable solution is to build models that can process a small set of input frames, i.e., a block of input at a time rather than using the entire input sequence.

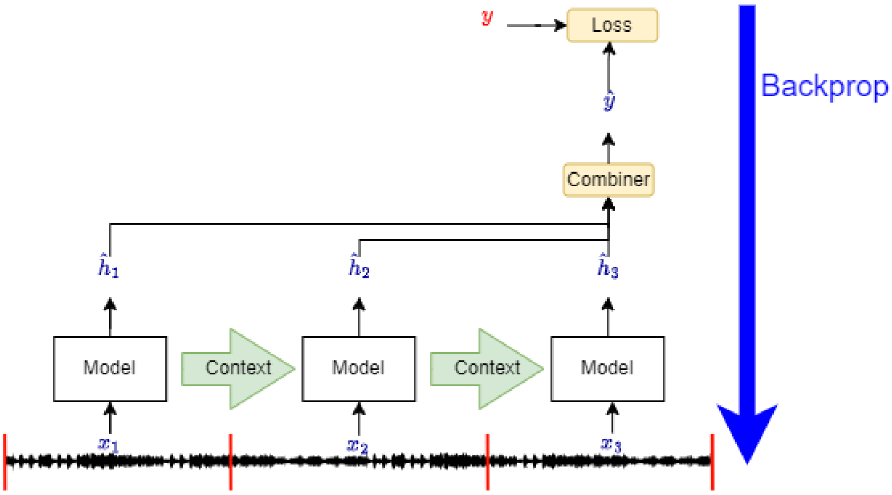


FIGURE 7.3: Blockwise modeling for streaming ASR/SLU

Such "block-wise" models can be trained to predict an output after seeing multiple blocks of input, which is shown in Figure 7.3. Most prior work has been focused on the former to enable streaming inference for tasks like speech recognition [Rao et al., 2017, Moritz et al., 2020, Narayanan et al., 2019, Tsunoo et al., 2021, Shi et al., 2021], speech translation [Ma et al., 2021], spoken intent detection [Deng et al., 2022], and wake word detection [Wang et al., 2021]. Block-wise training for streaming applications uses one block of input at a time to generate a block-level encoding. These block-level encodings from all blocks are then combined to make an utterance-level prediction. During training, all the inputs from all blocks, intermediate outputs, and the final output are retained in the computational graph for backpropagation. This requires significant computing and memory so these models still can't scale to very long audio sequences.

The alternative, shown in Figure 7.4 is to design block-wise models that produce outputs after every block and can be optimized at the block level without requiring the entire input to be present in the computational graph. Prior work in context-aware speech recognition for long conversations [Kim and Metze, 2018, Hori et al., 2021, Hori et al., 2020] can be considered

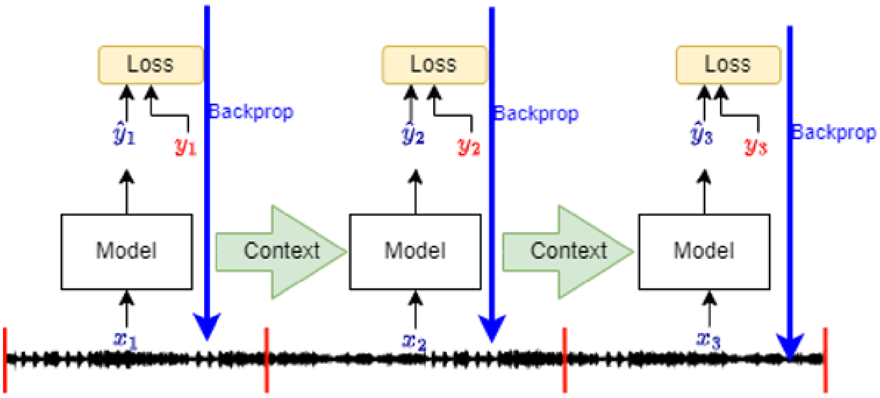


FIGURE 7.4: Block-wise Modeling with Block-level Optimization used in Contextual Conversational ASR and Proposed for Speech Summarization

instances of such block-wise models, in that they produce an utterance transcription for every new block of input, which is used to compute a loss and perform block-wise optimization.

Block-level targets like the utterance transcription in this case are relatively easy to derive for tasks like speech recognition, where there exists a monotonic alignment between the input frames and output tokens. However, for abstractive speech summarization, the relationship between input frames and output tokens is non-monotonic and indirect, and it is consequently challenging to obtain block-level targets.

In this section, we first mathematically formulate the process of block-wise training and introduce *Block-Wise Adaptation for Speech Summarization (BASS)*, an online model that can be trained with the full reference summary as the block-level target. This means that our model attempts to produce the output summary given only the first block, and then subsequently refines its prediction with every additional block of speech input. While streaming mechanisms assume that new acoustic inputs may incrementally modify the output, we permit the model to modify the entire summary if necessary based on the information present in the new input block. When using such block-wise inputs during training or inference, blocks should have access to the information encoded by previous blocks. We propose to achieve this by passing the latent representation across blocks since it is likely where the semantic information is encoded. While it is also possible to carry forward input acoustics or output summaries, these may not be as useful because input acoustics may not be as informative, and output summaries could be erroneous or change entirely with new blocks.

7.5.1 Formulating Block-wise Training

Given a long audio instance with N frames of D-dimensional speech features $X = (\mathbf{x}_i \in \mathbb{R}^D | i = 1, 2, \cdots, N)$, the goal of summarization is to produce a summary token sequence $Y = [y_1, y_2, \cdots y_L]$ of length L, which is shorter than the original sequence but still contains the relevant semantic information.

The input sequence X can also be represented as a sequence of T abutting blocks with block size B such that $X = \{X^1, X^2, ...X^T\}$. The i-th input block X^i produces a block-level output $\hat{Y}^i$, which is the model hypothesis for the full reference Y. We use the notation $X^{1:T}$ to represent $X^1, \cdots, X^T$ and $Y^{1:T}$ to represent $Y^1, \cdots, Y^T$.

The goal of a block-wise model is to generate the best possible summary $\hat{Y}^T$ after seeing the T blocks of input. Equation 7.6 expresses the probability of observing the final output sequence Y^T given the input blocks $X^{1:T}$ based on the joint conditional density $\mathbb{P}(Y^T, Y^{1:T-1}|X^{1:T})$.

$$\mathbb{P}(Y^T|X^{1:T}) = \sum_{Y^1} \cdots \sum_{Y^{T-1}} \mathbb{P}(Y^T, Y^{1:T-1}|X^{1:T}) \tag{7.6}$$

Using the chain rule of probability, we can represent the inner term $\mathbb{P}(Y^T, Y^{1:T-1}|X^{1:T})$ as shown in Equation 7.7.

$$\mathbb{P}(Y^{1:T}|X^{1:T}) = \mathbb{P}(Y^T|X^{1:T}, Y^{1:T-1})\mathbb{P}(Y^{1:T-1}|X^{1:T}) \tag{7.7}$$

Based on the fact that the model is causal (streaming), the present output cannot depend on future outputs or inputs. This implies $\mathbb{P}(Y^{1:T-1}|X^{1:T}) = \mathbb{P}(Y^{1:T-1}|X^{1:T-1})$. Combining this with Equation 7.7 results in Equation 7.8.

$$\mathbb{P}(Y^{1:T}|X^{1:T}) = \mathbb{P}(Y^T|X^{1:T}, Y^{1:T-1})\mathbb{P}(Y^{1:T-1}|X^{1:T-1}) \tag{7.8}$$

This leads to the following general decomposition based on the chain rule and the streaming assumption, shown in Equation 7.9.

$$\mathbb{P}(Y^{1:T}|X^{1:T}) = \mathbb{P}(Y^T|X^{1:T}, Y^{1:T-1}) \cdots \mathbb{P}(Y^1|X^1) \tag{7.9}$$

Consider Equation 7.6 which involves marginalizing over the output variables $Y^{1:T-1}$, and is challenging to compute. Rather than evaluating all possible values for past context $Y^{1:T-1}$, we can greedily perform this optimization, i.e., by choosing the block-level output sequence with the highest probability as context for future predictions. Combining this Viterbi assumption with Equations 7.6 and 7.9 leads us to the final formulation in Equation 7.10.

$$\mathbb{P}(Y^T|X^{1:T}) \approx \max_{Y^T} \mathbb{P}(Y^T|X^{1:T}, Y^{1:T-1}) \cdots \max_{Y^1} \mathbb{P}(Y^1|X^1) \tag{7.10}$$

In summary, Equation 7.10 demonstrates a setup wherein after receiving a new block of input, we maximize the probability of the block level output being as close as possible to the ground-truth summary given past and current block inputs and past block level outputs. In practice, we take in a block of input and any context from the past, then we compute a divergence between the block-level output and the ground-truth summary. We perform backpropagation with this criterion to update the neural network parameters after every block.

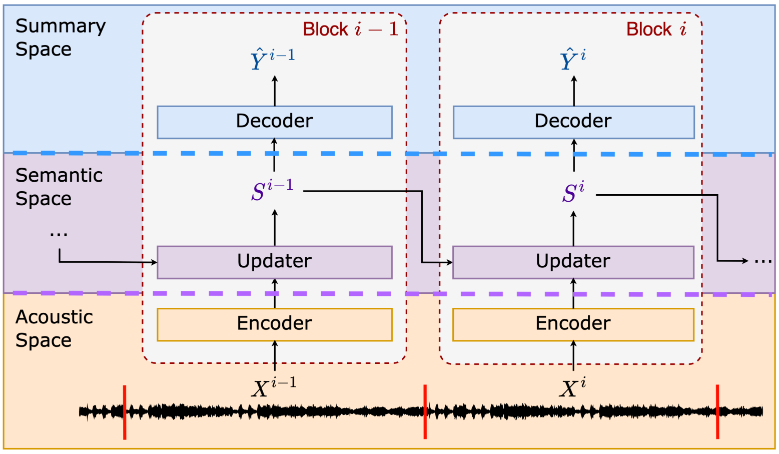


FIGURE 7.5: The framework for our proposed BASS: Blockwise Adaptation for Speech Summarization

Apart from the aforementioned assumptions, we can also make the Markov assumption while modeling contextual dependence. To minimize the impact of context from further away blocks on the current block, we can rewrite $\mathbb{P}(Y^{1:i}|X^{1:i}) = \mathbb{P}(Y^{i-M:i}|X^{i-M:i})$.

7.5.2 Modeling Strategy and Architecture

The proposed BASS model is shown in Figure 7.5. Different from past work in summarization, we explicitly introduce a semantic representation variable $S = (\mathbf{s}_i \in \mathbb{R}^F | i = 1, 2, 3, \cdots, M)$, which comprises M F-dimensional vectors. S contains the semantic information encoded in the speech X, and separates the acoustics from the summary. Modifying the input language or ambient environment changes the acoustics, but not the semantics. Summaries are generated by sampling from this rich semantic representation, and modifying S leads to a different summary Y.

The process of speech summarization occurs at the intersection of three distinctive spaces- the acoustic space, the semantic space, and the summary space. The acoustic space comprises the acoustic input X, which it transforms into semantic representations. The summary Y can be produced in the summary space by sampling based on the semantic representations S. We can reasonably assume that the acoustics and summary are conditionally independent given the semantics, and thus disentangle the acoustics and the summary.

Consider the task of estimating the most likely summary Y given the input speech X under this setting.

$$\begin{aligned}\hat{Y} &= \arg\max_{Y} \mathbb{P}(Y|X) \\ &= \arg\max_{Y} \sum_{S} \mathbb{P}(Y, S|X) \\ &\approx \arg\max_{Y} \max_{S} \mathbb{P}(Y, S|X)\end{aligned} \tag{7.11}$$

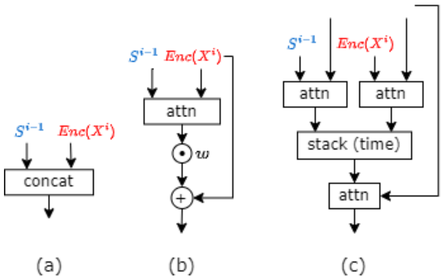


FIGURE 7.6: Updater mechanisms (a) concatenation, (b) gated attention, and (c) hierarchical attention use previous embedding S^{i-1} and encoding Enc(X^i) to produce current embedding

Equation 7.11 describes the process of identifying the most likely hypothesis summary $\hat{Y}$ given the input X and semantic representation S. From Figure 7.5, based on the conditional independence between the summary Y and acoustics X given semantic representation S, we can write $\mathbb{P}(Y, S|X) = \mathbb{P}(Y|X)\mathbb{P}(S|X)$. Thus, we can obtain the solution for Equation 7.11 using the coordinate descent update shown in Equation 7.12.

$$\begin{aligned}\hat{S} &= \arg\max_S \mathbb{P}(S|X) \\ \hat{Y} &\approx \arg\max_Y \mathbb{P}(Y|\hat{S}) \max_S \mathbb{P}(S|X)\end{aligned} \tag{7.12}$$

From Equations 7.12 and 7.10, BASS estimates $\mathbb{P}(Y^i|X^{1:i}, Y^{1:i-1})$ using Equation 7.13.

$$\mathbb{P}(Y^i|X^{1:i}) = \mathbb{P}(Y^i|S^i)\mathbb{P}(S^i|S^{1:i-1}, X^i) \tag{7.13}$$

The encoder and decoder model the probabilities $\mathbb{P}(S|X)$ and $\mathbb{P}(Y|S)$ respectively. The updater uses the past semantic embeddings and the current encoder output to produce the current semantic embedding. Figure 7.6 shows three alternate structures for our updater to aggregate semantic context from the prior and the current block:

1. **Concatenation**: $S^i = \text{Concat}(S^{i-1}, \text{Encoder}(X^i))$
   The current semantic embedding is obtained by concatenating the embeddings from the previous and current blocks.

2. **Gated Attention**:
   $S^i = \text{Encoder}(X^i) + w \cdot \text{Attn}(S^{i-1}, \text{Encoder}(X^i))$
   The current and previous semantic embeddings are combined using an attention mechanism and incorporated into the final embedding as a weighted sum.

3. **Hierarchical Attention**:
   $S_i = \text{Attn}([\text{Attn}(S^{i-1}, D^i), \text{Attn}(\text{Encoder}(X^i)), D^i]), D^i)$
   This method performs the context passing within each decoder block, based on hierarchical

attention [Libovický and Helcl, 2017]. We compute attention for the current decoder state D^i with the previous and current semantic embeddings independently. Then, we stack the two attention outputs and perform a second level of attention between this result and the decoder state.

7.5.3 Experimental Setup

Models: Our models use ESPNet2[1] [et. al., 2018] and 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 blocks [Gulati et al., 2020b] with 8 attention heads and a hidden size of 2048. The decoder has 6 transformer blocks, with 4 attention heads and a hidden size of 2048. The total number of model parameters is 103M. 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.

ASR: ASR models are trained with Connectionist Temporal Classification (CTC) and Cross-Entropy loss with CTC weight of 0.3. We use the Adam optimizer with peak lr=0.001, and a warmup scheduler for ASR pre-training. This takes 2 days on 8 V-100 32G GPUs

SUMM: Our summarization models are trained with cross-entropy loss and label smoothing of 0.15. During inference, we use a beam size of 8. Model averaging was not performed as it was found to hurt summarization performance. Fine-tuning is run for a day on one A40 GPU.

BASS: For BASS models, we use a block size of 1,000 input frames, corresponding to 10s of audio. We only use the semantic embedding from the previous block as context for the current block unless otherwise specified.

Evaluation: We evaluate our models with ROUGE [Lin, 2004a], METEOR [Banerjee and Lavie, 2005b], and BERTScore [Zhang* et al., 2020], which are the most common automatic metrics for evaluating summarization models.

BASS-Scratch: BASS-Scratch models are trained using flat start, without pre-training on the first block.

BASS-Adapt: BASS-Adapt models are trained by adapting models trained on the first block to longer input contexts.

7.5.4 Experimental Results

7.5.4.1 Truncated Input Baselines

First, we train end-to-end summarization baseline models on truncated inputs (`Trunc`) that are 10 seconds long and 30 seconds long. Table 7.2 reports the results of training on truncated inputs

[1] Code will be released in https://github.com/espnet/espnet

TABLE 7.2: Performance of Block-wise Adaptation and Training Approaches compared to Truncated Baselines with different inference strategies using ROUGE, METEOR, and BERTScore metrics- higher scores indicate better performance

Training Method	Inf.	Pre-train	Context	ROUGE-L↑	METEOR↑	BERTScore↑
Trunc, Restricted SA [Sharma et al., 2022b]	Std.	X	100s	56.10	29.30	91.53
Trunc, Full SA [Matsuura et al., 2023d]	Std.	X	100s	62.10	32.50	93.00
+ TTS Aug. [Matsuura et al., 2023d]	Std.	X	100s	65.00	34.90	93.80
Trunc-Baseline	Std.	X	10s	56.79	30.00	91.6
Trunc-Baseline	Std.	X	30s	59.16	31.76	92.08
Trunc-Baseline	Std.	X	60s	60.49	32.47	92.38
BASS-Adapt	Block	10s	30s	**60.17**	**32.17**	**92.51**
BASS-Train	Block	X	30s	54.79	29.12	89.40

and evaluating recordings that are 10 seconds and 30 seconds long, compared to different state-of-the-art approaches referenced in prior work. We note that using the standard full multi-head attention provides significant gains over restricted self-attention, and therefore use the standard multi-head self-attention for our experiments.

7.5.4.2 Block-wise Training versus Truncated Training

The proposed BASS method can be used to help models trained on shorter recordings adapt to longer inputs (BASS-Adapt), or to train models from scratch in a block-wise manner (BASS-Train). Inference for blockwise models can be performed in the standard manner, i.e., where the entire input is fed in at once to predict the final output. Alternatively, Block inference can be performed, where the input is fed as abutting blocks of input as described in Section 7.5.1.

We train BASS-Adapt initialized from the 10-second truncated baseline to handle 30-second recordings and infer using standard and block mechanisms. BASS-Adapt is compared against BASS-Train by training a model on 30-second recordings from scratch using our BASS algorithm. The latter performs worse - for training from scratch, the challenge is relatively poor initial context. Initially, the learned context is not very helpful which leads to slower convergence and poorer performance.

We compare both model adaptation and training strategies in Table 7.2. We see that the proposed BASS-Adapt approach outperforms BASS-Train on all metrics. We also observe that our proposed BASS algorithm improves over the truncated 10-second baseline and the truncated 30-second baseline. BASS-Adapt with block inference results in a nearly 4-point improvement in ROUGE-L over the 10-second truncated baseline, and a 1-point improvement in ROUGE-L over the truncated 30-second baseline. This result is comparable to that obtained by a truncated input baseline that takes in 60 seconds of audio, showing that our BASS model trained with 30-second recordings comprising 10-second chunks can do as well as a model trained on 60-second recordings. The proposed approach is more computationally efficient than the baseline by a factor of 3 since the proposed approach uses 3x smaller inputs 3 times for quadratic self-attentions.

TABLE 7.3: Part-of-speech coverage between the Predicted Summary and the Reference for Truncated 10s baseline and BASS-ADAPT 30s model

Model	Noun	Verb	Adj	Adv	Prop. Noun
Baseline 10-sec	0.85	0.76	0.84	0.65	0.78
BASS-ADAPT	**0.87**	**0.79**	0.84	**0.67**	**0.81**

Finally, Table 7.3 sheds light on the improvement in the prediction of different parts of speech in the reference summary using the best BASS-ADAPT model. We observe that the proposed model generally improves the prediction of all parts of speech. Future work may benefit from exploring named entity prediction for summaries.

7.5.4.3 Comparison of Block-wise Adaptation Strategies

TABLE 7.4: Performance of BASS models with block level inference across different implementations of the semantic updater model. Models are pre-trained on 10s and fine-tuned on 30s. R-1, R-2, R-3 represent the ROUGE-1, ROUGE-2, and ROUGE-L metrics respectively

Updater	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	METEOR↑	BERTScore↑
Concat	**63.99**	**49.00**	**60.17**	**32.17**	**92.51**
Gated Attn	63.94	48.91	60.16	32.12	92.12
Hier. Attn	59.71	43.99	55.74	29.32	91.27

Table 7.4 compares the various modeling strategies for the semantic updater. We observe that simply concatenating the semantic embedding of the previous (one) block with the current block yields significant improvements in summarization performance. Of the three updater mechanisms described in Figure 7.6, gated attention and concatenation appear to yield similar gains in performance, with hierarchical attention performing significantly worse. Gated attention is able to achieve similar performance while reducing having a very small memory footprint compared to concatenation.

7.5.5 Comparing BASS to Optimized Self-Attentions

Table 7.5 compares the two dominant approaches for end-to-end speech summarization with approaches that utilize truncated baselines. Approximation errors with FNet result in a 3.2% absolute difference in ROUGE-L between the performance of BASS and FNet, for both 30s and 100s long inputs. BASS improves over the standard self-attention-based conformer baselines.

TABLE 7.5: Performance of BASS models with block-level inference across different implementations of the semantic updater model. Models are pre-trained on 10s and fine-tuned on 30s. R-1, R-2, R-3 represent the ROUGE-1, ROUGE-2, and ROUGE-L metrics respectively

Model	R-1↑	R-2↑	R-L↑	METEOR↑	BERTScore↑
FNet (30s)[2]	60	40.8	57	27.5	92.6
BASS (30s,10s)	63.99	49	60.17	32.17	92.51
Conformer (30s)	63.3	47.58	59.16	31.76	92.08
FNet (100s)	61.42	45.37	57.27	29.77	91.62
BASS (100s,25s)	**64.57**	**49.11**	**60.49**	**32.47**	**92.38**
Conformer (10s)	60.86	45.13	56.78	29.98	91.41

7.6 Relevance-Aware Blockwise Adaptation for Speech Summarization (R-BASS)

Block-wise adaptation for Speech Summarization (BASS) [Sharma et al., 2023b] 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 is well suited for streaming inference, where summaries are expected to be updated based on new acoustic information, processing of all relevant and irrelevant blocks is computationally inefficient. In this section, 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 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.

Then, we examine automatic methods to identify the relevance of blocks based on lexical and semantic similarity. Lexical similarity calculates 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 [Devlin et al., 2019b] 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.

7.6.1 Introducing R-BASS

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 based on it, otherwise not. 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.

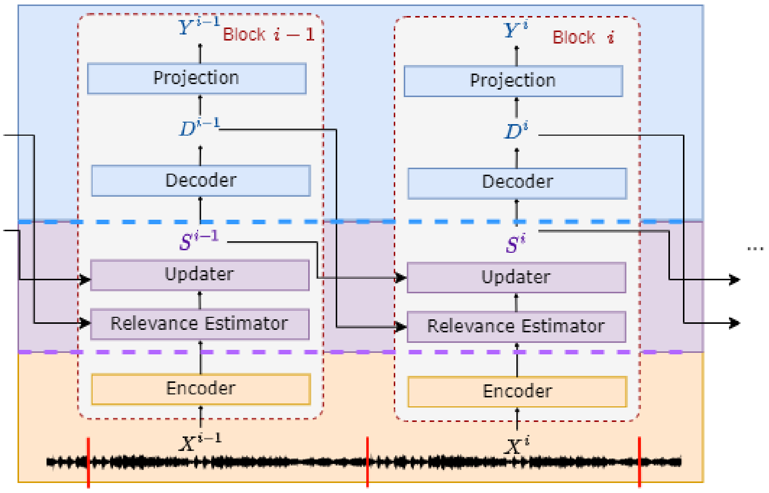


FIGURE 7.7: Proposed Relevance-Aware Block-wise Adaptation for Speech Summarization

Figure 7.7 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 7.14 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.

$$R^i = \text{Sim.}(Y, X^i) \approx \text{Sim.}(\hat{Y}^{i-1}, Enc(X^i)) \tag{7.14}$$

To compute similarity, there is a need to bring $\hat{Y}^{i-1}$ and $Enc(X^i)$ into a common space. 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 help the model learn to accurately predict relevance.

7.6.2 Identifying 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. 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 keywords that are present in the summary, then the block may be considered relevant - we refer to this as *lexical similarity*. If the block-level transcript is related in intent or meaning to the summary, then the block is relevant, and we refer to this as *semantic similarity*. We can use these relevance pseudo-labels directly during inference (R-BASS-Inf) to decode using only relevant blocks. We compare this to a random baseline which randomly selects $x\%$ blocks without relying on any relevance metric, so as to validate whether the notions of relevance capture useful information. We remove stop words using NLTK [Bird and Loper, 2004] before computing similarity metrics. Apart from using labels during inference, we can also train R-BASS models to predict relevance on a given block, optimized using a relevance loss that is described in the next section.

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 block transcript. 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 follows: $LS(T^i, Y) = Count(y \in T^i | y \in Y)/Count(y)$

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 τ to convert $LS(T^i, Y)$ to binary.

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 as follows: $SS(T^i, Y) = \text{cos-sim}(\mathcal{B}(T^i), \mathcal{B}(Y))$ 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

7.6.3 Introducing Relevance Loss for Gated Attention

Now that we devised mechanisms to model relevance with R-BASS, and methods to obtain automatic annotations, we describe the *Relevance loss*. To estimate the true relevance R^i, for the i-th block, we compute a 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 of the transcript and the reference summary. In practice, we find this to work better than the original approach. The BCE loss can be written as shown in Equation 7.15.

$$\mathcal{L}_{bce} = -\mathbb{E}(y \log(p) + (1-y)\log(1-p)) \tag{7.15}$$

7.6.4 Experimental Setup

Experiments are performed on the How2 dataset for video summarization. Our conformer encoder [Gulati et al., 2020b] - transformer [Vaswani et al., 2017b] decoder models use ESPNet2 [et. al., 2018], and computational cost and hyperparameters. 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., 2020b] blocks with 8 attention heads and hidden size 2048. The decoder has 6 transformer [Vaswani et al., 2017b] blocks, with 4 attention heads and hidden size 2048. Models have 103M paraeters. 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., 2022a].

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.

7.6.5 Experimental Results

First, we show that the lexical and semantic relevance, as described earlier, are related to each other. We consider blocks of 10s to have more fine-grained control blocks and calculate the lexical and semantic relevance per block, creating a vector of binary relevance measures for each recording. We then take the dot product of the two vectors, and average over all the examples. The averaged dot product is 0.7, demonstrating that semantic and lexical relevance capture similar information in the data.

Figure 7.8 shows the binary relevance of the lexical and semantic similarity scores, averaged over all the examples in the training data. 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. Comparing this to a random baseline,

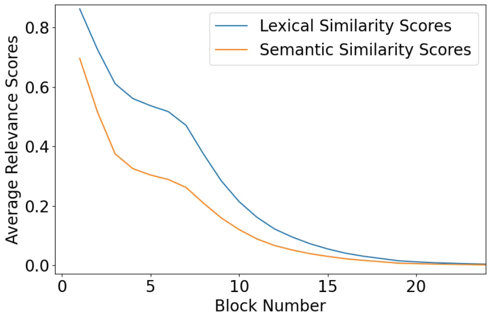


FIGURE 7.8: Binary relevance scores averaged over all training samples as a function of block index in audio recordings

the performance drop is 64.9%, which is quite significant demonstrating the importance of choosing relevant blocks.

TABLE 7.6: Performance of R-BASS-Inf and R-BASS w/ Loss using Lexical and Semantic Similarity. ROUGE-L, METEOR, and BERTScore are reported with the % of dropped blocks (efficiency gain

Updater	ROUGE-L↑	METEOR↑	BERTScore↑	% Dropped↑
Baseline-FNet	57.27	29.77	91.62	-
Baseline- BASS	57.98	31.67	91.48	-
Random Inf. Baseline	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	69.2
Sem. R-BASS w/ Loss	57.82	31.15	91.42	**86.31**

R-BASS with Relevance Loss: Table 7.6 summarizes the results of using automatic annotations for training and inference. When the reference relevance annotations are used to guide semantic context updation during inference (R-BASS Inf.), over 64 % of blocks can be dropped for longer sequences using semantic relevance, while lexical relevance can reduce around 40 % of the blocks. When using the proposed training and inference method, we find that the number of dropped blocks can be further increased with a small drop in performance. Training with lexical relevance can enable dropping 69.2 % of blocks and training with semantic relevance enables dropping up to 86 % of blocks. *R-BASS* with semantic relevance is 2.2x faster than the BASS for inference.

Qualitative Analysis: We perform qualitative analysis by human inspection to evaluate the impact of the slight degradations in the performance of R-BASS on summary quality. We observe from Table 7.7, that the quality of summaries is not significantly degraded when using R-BASS. We also compute UniEval [Zhong et al., 2022b] scores that attempt to mimic human preference and find from Table 7.8 that R-BASS does not significantly degrade summary quality on consistency, coherence, relevance, and factualness.

TABLE 7.7: 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 .

7.7 Revisiting the Memory-Performance Tradeoff

In this section, we revisit the memory performance tradeoff introduced in Section 7.3.

Figure 7.9 plots the performance in ROUGE-L on the How2 dataset as a function of number of float numbers needed to compute self-attentions for an input of size 100s. The figure was updated to include the proposed block-wise methods for speech summarization. We observe that though BASS significantly improves performance, it also comes with a 5x increase in computational complexity over FNet due to the use of standard self-attentions within each block. However, using our notion of semantic relevance and the proposed R-BASS approach, we see that the model maintains performance while discarding the 5x increase in computational complexity.

UniEval Measure	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

TABLE 7.8: UniEval scores of R-BASS-Inf, BASS and R-BASS w/ Loss models

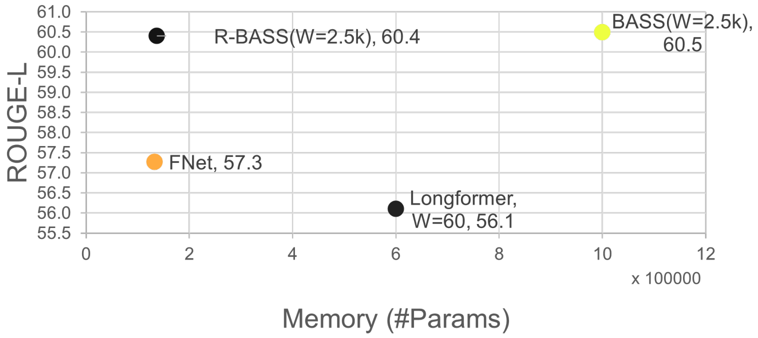


FIGURE 7.9: The Memory Performance Trade-off Plot with all Methods

R-BASS achieves the best of both worlds – highest performance in ROUGE-L terms and low computational complexity of the self-attention, and the point representing R-BASS is in our ideal spot - the top left.

7.8 Chapter Conclusion

In this chapter, we addressed the challenge of the global acoustic context by proposing three solutions.

To reduce the computational complexity of FNet self-attentions while bridging the performance gap between the state-of-the-art linear transformer and standard self-attention, we introduce the XNORformer. Experiments show that the proposed XNORformer outperforms the state-of-the-art linear transformer, i.e., the Cosformer on speech summarization for the How2 data.

We shed light on the trade-off between summarization performance and computational complexity of the self-attention for all approaches. We note that though FNet and linear transformers improve complexity over the standard self-attention, performance suffers a significant degradation.

To address this performance gap and enable speech summarization models to process arbitrarily long inputs, we introduce Block-wise Adaptive for Speech Sequences (BASS) - an algorithm that consumes the input in blocks and passes semantic context across blocks to encourage better learning. The BASS algorithm can be used to adapt pre-trained truncated input models to longer sequences, or train models over long sequences from scratch. We show that the proposed model outperforms truncated baselines and enables the training of speech summarization models with very long inputs.

Experiments show that BASS significantly improves performance over the optimized self-attentions, but has a greater computational cost. To improve the efficiency of BASS, R-BASS is introduced. We propose a novel mechanism to obtain binary relevance decisions during model inference and introduce automatic methods to produce annotations for relevance based on lexical and semantic similarity between the reference summary and ground-truth block-level transcript. Experiments demonstrate that these two methods of automatically producing relevance annotations are correlated and that there exist multiple irrelevant blocks by this measure within long recordings. Finally, we discuss a relevance loss formulation to help the model predict relevance based on lexical and semantic similarity. Experiments show that training with the proposed semantic similarity loss enables end-to-end processing by a single model with 86% gain in efficiency and 2.2x faster inference than BASS while obtaining relatively small performance degradations.

Revisiting the performance-computational cost tradeoff shows that R-BASS obtains the best performance at the least cost.

Chapter 8

Improving Diversity of Summaries with AugSumm

Automatic summarization models are heavily influenced by the structure of the summaries used to train them. This can be problematic because these reference summaries themselves can be shaped by factors beyond the content itself.

For example, the How2 corpus contains short, uppercase summaries, often starting with "in this free video." Models trained on this data tend to generate similar summaries, regardless of the input recording, mimicking the structure of the training data rather than creating optimal summaries for all recordings.

This raises concerns about comprehension and potential bias. Different individuals may have varying preferences for summary structure and phrasing – what works for one reader might not be ideal for another [Fisk and Hurst, 2003].

Current speech summarization models typically rely on supervised learning, where they are trained on labeled data consisting of audio recordings and a single corresponding reference summary for each recording. However, for any given audio recording, there can be multiple valid summaries that convey the same information using different sentence structures and phrasing. Therefore, a more effective approach for training speech summarization models within the supervised learning paradigm would be to train them to generate *any or all of the valid summaries* with different structures, rather than forcing them to adhere to a single structure. This multi-reference training framework offers several advantages:

1. Reducing annotator bias: By incorporating diverse perspectives, the model wouldn't be solely influenced by individual preferences in the training data.
2. Enhancing diversity: Utilizing multiple references could lead to summaries with varying structures and presentations, potentially catering to a wider range of reader preferences.

3. Improving generalizability: Utilizing multiple references during training may improve the ability of the model to generalize to new and unseen application domains.

However, creating corpora with multiple reference summaries presents a challenge. Traditionally, human annotation serves as the primary method, but it is a costly and time-consuming process, making large-scale annotation for multiple references impractical.

In this chapter, we address this limitation by using large language models (LLMs) as a proxy for human annotators. We use them as generative models to sample additional references for training and evaluation since they are exceptional at generating human-like text [Aher et al., 2023, Sorensen et al., 2022, Argyle et al., 2023].

We introduce *AugSumm* (Augmenting Summaries), a method to generate additional synthetic summaries based on available data — the reference (ground-truth) transcript of audio files or the reference summary of audio recordings where it exists. Additional reference summaries can be obtained based on the reference transcript of the input audio by the direct process of summarization, which we term *AugSumm-direct*. Additional references can be generated by paraphrasing existing reference summaries, which we term *AugSumm-paraphrase*. We note that *AugSumm-direct* can be applied to any corpus that contains audio recordings since the transcript can be obtained using speech recognition, while *AugSumm-paraphrase* requires a corpus with abstractive reference summaries. In this chapter, we demonstrate the benefits of AugSumm using the existing How2 [Sanabria et al., 2018] corpus for speech summarization. Analysis shows that LLM-generated AugSumm summaries are useful and valid by extensively examining the generated summaries with lexical-, semantic-, and human-based metrics. Our human evaluation finds that AugSumm summaries are more valid than the reference summaries with a 95% confidence interval.

Once additional summaries are generated, multiple methods can be used to incorporate them within training. We explore three primary paradigms — multi-style training, two-stage training, and a combination of the two. For evaluation, we construct a new synthetic evaluation set with AugSumm summaries to evaluate model generation for structures different from those in the existing reference summaries. Experiments show that using AugSumm summaries in training can play a critical role in altering its behavior during inference. Pre-training on both AugSumm and existing reference summaries, and fine-tuning on the existing reference summaries improved all 10 metrics across both the original and synthetic test sets.

8.1 The challenge of unitary references

The goal of Automatic Speech Summarization [Murray et al., 2010, Palaskar et al., 2019a, Li et al., 2019, Shang et al., 2018] is to condense essential information from long recordings and generate human-like summaries. Figure 8.1 shows the different steps in the process of generating abstractive summaries from input speech.

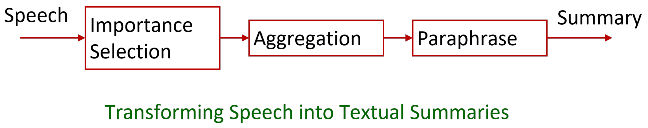


FIGURE 8.1: The Process of Abstractive Summarization includes selecting important portions of speech (semantic concepts), aggregating these, and then paraphrasing to produce the final summary

Humans summarize speech in two steps: (i) select important semantic concepts and (ii) combine these concepts to form an abstractive summary. Humans differ in what they perceive as important [Rath et al., 1961], and how they combine these important concepts to form a summary. Therefore, numerous *valid* summaries exist given input speech due to differences in concept selection and semantic concept combination, and these summaries can be reasonably modeled as belonging to a *distribution*.

However, current approaches for automatic speech summarization use a single subjective ground-truth human-annotated summary from one annotator to train and evaluate summarization models. We believe that sampling a single ground-truth summary per sample does not sufficiently represent the entire distribution, and hence both the training and evaluation of SSUM models with a single reference can be sub-optimal [Cohan et al., 2022].

In the following section, we introduce AugSumm, our method to address the challenge of unitary references for speech summarization.

8.2 Our Approach: AugSumm

To generate summaries with greater fluency, coherence, and lexical diversity, we contend that it is necessary to expose models to multiple references during training.

Obtaining multiple human-annotated summaries for each recording would be ideal for a more accurate approximation of the underlying distribution. However, the high cost of human annotation makes obtaining multiple references through human annotation infeasible.

Recently, large language models have been shown to be exceptional at generating text in a manner humans Large Language Models (LLMs) [Ouyang et al., 2022, Touvron et al., 2023a, Touvron et al., 2023b] to generate textual data for summarization. LLMs have been used in prior work to generate synthetic data for text classification using approaches such as Self-Instruct [Wang et al., 2023], AttrPrompt [Yu et al., 2023a], ZeroGen [Ye et al., 2022], and more recently use in-context learning with seed samples [Yu et al., 2023b]. However, generating synthetic data for abstractive speech summarization is more complex, and has not been addressed in prior work to the best of our knowledge.

8.2.1 Formulation

As we saw in Chapter 2, speech summarization models attempt to represent the probability distribution $P(Y|X;\theta)$, i.e., the likelihood of predicting the summary sequence Y given the input speech X.

To train such models in a supervised manner, we sample pairs (X, Y) from the true joint distribution of X and Y, $\mathbb{P}(X, Y)$ and create a training corpus D. The greater the number of such pairs sampled, the better the training corpus is at representing the true joint distribution $\mathbb{P}(X, Y)$. Sampling in this case corresponds to obtaining more human annotations Y for many different speech inputs X. There are two main limitations of this sampling method.

First, speech summarization models are trained to estimate $P(Y|X;\theta)$. Random sampling from the joint distribution yields a set of valid (X, Y) pairs, however, sampling from $P(Y|X)$ is likely better. The latter means that X is selected first, and for every X, we sample multiple values from $P(Y|X)$, effectively improving distribution coverage and reflecting the fact that there are multiple valid summaries for every input. Second, additional samples require human annotations, and this is challenging to obtain in practice. The proposed AugSumm aims to sample additional training examples for every X from the conditional distribution $\mathbb{P}(Y|X)$.

Since the true probability density, $\mathbb{P}(Y|X)$ is unknown, we resort to using a model with parameters θ that represents $P(Y|X;\theta)$ to obtain such samples. It is also challenging to obtain models that represent $P(Y|X)$, so we factor the conditional distribution to include the transcript T.

$$P(Y|X) \approx \text{max}_T P(Y|T)P(T|X). \tag{8.1}$$

From Equation 8.1, the conditional distribution $P(Y|X)$ can be factored into the product of the likelihood of producing a summary given the transcript $P(Y|T)$, and the probability of obtaining the transcript T from the input speech X. The second term $P(T|X)$ is 1 for ground-truth transcripts (from ASR data or human transcription). Therefore, the problem of modeling $P(Y|X)$ reduces to modeling $P(Y|T)$. Any generative text summarization model, including an LLM, represents precisely this probability.

Based on this formulation, *AugSumm* obtains synthetic summaries from the LLM by sampling as shown in Equation 8.2, where $P(Y|T, Prompt)$ represent the LLM operation with a static Prompt and Y_{synth} represents the generated summary.

$$Y_{synth} \sim P(Y|T, Prompt). \tag{8.2}$$

8.2.2 AugSumm-direct

The first approach, *Augsumm-direct*, utilizes Equation 8.2 to generate synthetic summaries Y_{synth} from transcripts by using $direct\_prompt$ in place of $Prompt$ as shown in Equation 8.3

$$Y_{synth} \sim P(Y|T, direct\_prompt) \tag{8.3}$$

However, because *AugSumm* adopts LLMs in place of humans to obtain summaries, the output summary can be erroneous. To address this, we prompt the LLM to generate extractive summaries by selecting key phrases from the input transcript and combining them. This helps produce additional references using words that already exist in the transcript rather than generating unconstrained abstractive summaries that are factually inconsistent with the transcript.

> *You are here to create an extractive summary from the transcript. An extractive summary uses words from the input to convey the important portions of the video. Please make sure that the summary has between 40 and 60 words. Respond with only the extractive summary for:* {*transcription*}.
> *transcription:*

FIGURE 8.2: Prompt for extractive AugSumm in word-level, directly generated using the transcript only without GT summary.

> *You are here to create an extractive summary from the transcript. An extractive summary uses words from the input to convey the important portions of the video. Please make sure that the summary has between 40 and 60 words. Respond with only the extractive summary for:* {*transcription*}. ***transcription****:*

FIGURE 8.3: Direct AugSumm Prompt that produces word-level extractive summaries generated from the transcript of the audio input.*GT_len* refers to the number of words of the GT summary.

Figure 8.3 displays the prompt used. *AugSumm-direct* can be used with any speech data with available transcriptions and hence it has the potential to train speech summarization models with existing thousands of hours of ASR corpora.

8.2.3 AugSumm-paraphrase

AugSumm-direct presents a straightforward approach to sample additional summaries. However, we found empirically that current LLMs cannot generate valid abstractive summaries that focus on the same semantic concepts as the available reference.

As reported in prior work, prompting affects the generated outputs of LLM significantly [White et al., 2023]. We therefore propose another method, *AugSumm-paraphrase* to sample additional summaries by simply paraphrasing existing reference summaries.

In *AugSumm-paraphrase*, we devise prompts to query LLMs so that they paraphrase ground-truth reference summaries. An example prompt is shown in Figure 8.4.

You are here to paraphrase a given summary in the same style as the provided input. Please make sure that the summary has between min(GT_len-10, 20) to max(GT_len+5, 20) words. ***given summary****:*

FIGURE 8.4: Paraphrase AugSumm Prompt (without concept words) that uses the reference (ground-truth) summary as input to produce paraphrases.*GT_len* refers to the number of words of the GT summary.

$$\begin{aligned} P(Y|T) = \sum_{Y_{GT}} P(Y, Y_{GT}|T) &= \sum_{Y_{GT}} P(Y|T, Y_{GT}) P(Y_{GT}|T) \\ &\approx \sum_{Y_{GT}} P(Y|Y_{GT}) P(Y_{GT}|T). \end{aligned} \tag{8.4}$$

Mathematically, *AugSumm-paraphrase* represents the conditional probability $P(Y|T)$ based on the conditional density of $P(Y, Y_{GT}|T)$ as shown in Equation 8.4 where $P(Y_{GT}|T)$ represents the process of human annotation and is a constant.

Using the fact that Y and Y_{GT} are conditionally independent given the transcript T, we note that $P(Y|T)$ can be expressed in terms of $P(Y|Y_{GT})$. This probability can be modeled by an LLM that produces a Y based only on the ground-truth reference summary.

$$Y_{synth} \sim P(Y|Y_{GT}, paraphrase\_prompt) \tag{8.5}$$

Using a different static prompt $paraphrase\_prompt$, sampling occurs as shown in Equation 8.5. Compared to *Augsumm-direct*, *Augsumm-paraphrase* uses Y_{GT} instead of T and a different prompt.

You are here to paraphrase a given summary in the same style as the provided input. Please make sure that the summary has between min(GT_len-10, 20) to max(GT_len+5, 20) words. Also please include these words in the summary: {important_keys}. ***given summary****:*

FIGURE 8.5: *Augsumm-paraphrase* Prompt (with concept words) that uses the reference (ground-truth) summary as input to produce paraphrases that contain the provided concept words.*GT_len* refers to the number of words of the GT summary.

To restrict the focus of paraphrasing from modifying important entities within the augmented summaries, we propose to modify *Augsumm-paraphrase* by providing a specific set of semantic concepts within the prompt.

For this purpose, we first extract semantic concepts as noun/verb phrases from the summary following [Palaskar et al., 2021b, Sharma et al., 2022b]. Then, we modify the prompt as shown in Figure 8.5 to produce the extracted semantic concepts within the paraphrase AugSumm.

8.3 Experimental Setup

8.3.1 Models and Data

Experiments were performed on the How2 dataset for speech summarization. Our E2E speech summarization models are implemented as attention-based sequence models with a 12-layer conformer [Gulati et al., 2020b,Guo et al., 2021] encoder, and a 6-layer transformer decoder. The encoder and decoder use a feedforward dimension of 2,048 and 512, along with 8 and 4 attention heads respectively. To efficiently process longer sequences, we replace standard self-attention with a parameter-less Fourier transform from FNet [Lee-Thorp et al., 2022b], which has been shown to work well for speech summarization [Kano et al., 2023b]. 43-dimensional filter-bank pitch features are extracted using Kaldi [Povey et al., 2011] and used as input to the model. The model is first pre-trained on ASR using the hybrid CTC/attention loss [Watanabe et al., 2017] with a CTC weight of 0.3 before being trained for speech summarization using cross-entropy. SpecAugment [Park et al., 2019b] is used for ASR pre-training, while no augmentation is used for speech summarization training.

8.3.2 Metrics

We employ a diverse set of metrics that assess lexical and semantic similarity between hypothesis and reference summaries. Lexical metrics in this paper include ROUGE-1 [Lin, 2004b] and ROUGE-L [Lin, 2004b]. These metrics measure in a word sense how close the model-generated summaries are compared to the reference summary. Semantic metrics complement lexical metrics by directly comparing semantic embeddings to assess similarity. In this work, we employ BERTscore [Zhang et al., 2019] and UniEval [Zhong et al., 2022a] as model-based metrics. We selected these two model-based metrics because BERTscore is a widely adopted metric in the research community and UniEval, although relatively new, provides scores across various dimensions that are also used in human evaluation – coherence, consistency, fluency, and relevance. Coherence and consistency compare the model-generated output with the transcript. Fluency is measured using the model-generated output only. Relevance compares the model-generated output with a reference.

8.4 Experimental Results

8.4.1 Quality of AugSumm summaries.

Tables 8.1 and 8.2 report multiple metrics that reflect the quality of LLM-generated summaries. For paraphrase AugSumm, Table 8.1, all three UniEval metrics except relevance were higher than the ground truth. For relevance, which measures how relevant a given summary is compared to the ground truth, is 98.32. Surprisingly, even BERTscore computed with ASR transcript

TABLE 8.1: Quality comparison of paraphrase AugSumm. Results between 2 to 7 columns are model-based where the first four are from UniEval [Zhong et al., 2022a] and the two last are from BERTscore [Zhang et al., 2019]. ROUGE-1 is based on lexical similarity. (Pa: paraphrase, Coh: coherence, Con: consistency, Flu: fluency, Rel: relevance, RG1: ROUGE-1 with transcript, BT-T: BERTscore with transcript, BT-G: BERTscore with ground-truth summary, HME: human evaluation (validity, %)

	Coh	Con	Flu	Rel	BT-T	BT-G	RG1	HME
GT	81.11	80.10	91.30	-	82.65	-	12.48	51.67
Pa	86.04	83.76	94.89	98.32	82.65	93.77	12.49	65.33

TABLE 8.2: Evaluating the quality of synthetic data generated using Direct AugSumm. (Di: direct AugSumm, Coh: coherence, Con: consistency, Flu: fluency, Rel: relevance, ROUGE-1: ROUGE-1 with transcript,RG-L: ROUGE-L with transcript)

	Coh	Con	Flu	BERTScore-Trans	ROUGE-1	RG-L
GT	81.11	80.10	91.30	82.65	12.48	8.30
Di	97.00	93.04	92.57	85.09	28.65	23.15

as reference were almost identical, indicating that semantics are preserved. ROUGE-1 scores are also similar, showing a similar degree of word overlap in synthetic summaries. For direct AugSumm, Table 8.2, three UniEval metrics were higher than both the paraphrase AugSumm and ground-truth .[1] ROUGE-1 and -L is higher than ground truth. High ROUGE values occur because we constrain the model to use the words within the transcript to generate direct summaries.

We further conducted a *human evaluation*, with 20 annotators and 15 questions each to choose among the four options: (i) summary 1 is valid while summary 2 is not, (ii) summary 2 is valid while summary 1 is not, (iii) Both summaries are valid, and (iv) Both summaries are invalid. In total, we received 300 responses, with 51, 92, 104, and 53 responses for each respective option. Here, summary 1 is the ground-truth How2 summary and 2 is the AugSumm summary. Therefore, from human ratings, ground-truth, and AugSumm summaries were considered valid 51.67 ± 0.0565% and 65.33 ± 0.0538% of the time respectively with a 95% confidence interval. We partly attribute this to the fact that How2 summaries were constructed by crawling user descriptions from YouTube and inserting them into predefined sentence structures. In summary, AugSumm summaries are valid based on lexical, semantic, and human evaluation.

TABLE 8.3: Baseline results trained with and without AugSumm labels. Results better than the Baseline are depicted in boldface and the best performance is underlined. ROUGE metrics with $*$ are measured using AugSumm labels. (BT-A: BERTscore with AugSumm test set)

	Coh	Con	Flu	Rel	BT-G	RG1	RGL	RG1*	RGL*	BT-A
Sharma et al.	-	-	-	-	91.53	60.73	56.10	-	-	-
Baseline	68.58	72.66	82.92	77.57	**92.16**	**61.95**	**57.52**	**47.69**	**36.77**	**88.40**
Paraphrase	**77.18**	**81.09**	**86.81**	**81.93**	88.88	51.44	35.20	43.98	30.13	87.16
+ w/ concept	**78.26**	**81.29**	**85.54**	**82.65**	90.21	54.95	35.69	47.80	31.78	**88.40**
Direct	**81.73**	**86.37**	**83.15**	62.93	81.36	23.04	16.10	17.99	12.84	80.69

[1] As summaries being extractive, it's not meaningful to measure relevance.

8.4.2 Baselines and AugSumm-only.

Table 8.3 compares the baseline and speech summarization models trained only with AugSumm summaries. Our baseline based on FNet outperforms the reported performance with restricted self-attention [Sharma et al., 2022b], and thus, we use the former. All models trained with AugSumm demonstrated improved performance on UniEval; however, the baseline has the best result on other metrics. Overall, we conclude that models trained with AugSumm have the potential for improvement when jointly trained with an existing ground-truth summary using various techniques.

TABLE 8.4: Comparing approaches to utilize AugSumm for training. All experiments have been conducted using "Paraphrase, w/ concept" in Table 8.3. Results better than the Baseline is depicted in boldface and the best performance is underlined. (Pt: pre-train, Ft: fine-tune)

	Coh	Con	Flu	Rel	BT-G	RG1	RGL	RG1*	RGL*	BT-A
	1 stage training									
GT-only (=Baseline)	68.58	72.66	82.92	77.57	92.16	61.95	57.52	47.69	36.77	88.40
AugSumm-only	**78.26**	**81.29**	**85.54**	**82.65**	90.21	54.95	35.69	**47.80**	31.78	88.40
GT half + AugSumm half	**73.34**	**76.77**	**83.91**	**80.10**	91.32	58.89	47.76	47.94	34.89	**88.49**
Enlarge (GT+AugSumm)	**75.73**	**78.38**	**84.81**	**82.18**	91.52	60.30	47.30	**49.14**	34.54	**88.66**
	2 stage training									
Pt Augsumm-only → Ft GT	**69.74**	**73.52**	**83.48**	**77.11**	**92.44**	**63.17**	**58.96**	**48.50**	**37.65**	**88.59**
Pt Enlarge → Ft GT	**69.70**	**73.42**	**83.17**	**78.79**	**92.48**	**63.64**	**59.37**	**48.52**	**37.61**	**88.53**
Pt GT → Ft AugSumm	67.18	72.33	82.01	76.18	**92.20**	**63.00**	**58.76**	**47.74**	**36.79**	88.27

8.4.3 Using AugSumm in Training

Table 8.4 shows the result of our various techniques to leverage AugSumm along with the existing ground-truth summary. "ground-truth half + AugSumm half" and "Enlarge" in Rows 3 and 4 show results when using both AugSumm and ground-truth labels by mixing the two or using them all. These two experiments improve all four UniEval metrics, but ROUGE or BERTscore with the ground-truth summary does not improve. We contend that this is because, in this setup, models are optimized to produce both the ground-truth or AugSumm summary during training, both of which contain different semantic concepts and phrase structures. ROUGE and BERTScore on the AugSumm test set improve when using both ground-truth and AugSumm data in training.

Rows 5 to 7 demonstrate the impact of adopting 2-stage training. 2-stage training is effective regardless of what kind of label is used in the pre-training and fine-tuning stages. Among them, pre-training with both labels and fine-tuning with the ground-truth summary (Row 6) performed the best. All 10 metrics showed improvement and ROUGE scores improved on both ground-truth and AugSumm test sets. Based on these results, we conclude that AugSumm can be used to build speech summarization models that generalize better.

8.5 Chapter Conclusion

In this chapter, we proposed *AugSumm*, which uses LLMs to augment summary labels for SSUM. We compared two methods to generate synthetic reference summaries - directly summarizing speech transcripts and paraphrasing existing GT summaries. Experiments on How2 showed that both were beneficial. Multiple methods to utilize augmented and GT summaries were discussed, with the 2-stage training framework yielding improvements across lexical and semantic metrics and across original and augmented test sets. We can obtain a 1-point absolute improvement in ROUGE-L by using AugSumm.

The success of this work, using LLMs as a source of data augmentation, inspires multiple directions for future work. First, advanced training techniques such as contrastive learning and mix-up can be now applied to better leverage multiple summary labels per audio file. Second, this method can be extended to several tasks such as speech translation where the label is a sentence.

Part IV

Conclusion

Chapter 9

Conclusion and Future Directions

9.1 Thesis Conclusions

This thesis has delved deep into the problem of speech summarization and made several contributions to automatic methods for speech summarization.

First, we identify the limitations of the classical cascade approach and propose a novel end-to-end approach for speech summarization. To realize end-to-end models on available computing resources, we address two imperative challenges – abstract learning to optimize models to learn the complex and indirect relationship between speech and summary, and global acoustic context to enable models to utilize long input contexts for speech summarization. To address the concern of abstract learning, we draw inspiration from curriculum learning to devise a two-stage training approach that involves pre-training the speech summarization model for utterance-level transcription and then fine-tuning the model for abstractive speech summarization. We recommend the use of restricted self-attentions to allow end-to-end models to look over longer input contexts. The proposed end-to-end approach leads to significantly improved performance with fewer parameters and a simpler training method, demonstrating that end-to-end models are not only a viable alternative for video summarization but also that they are the more performant and efficient alternative.

Next, we address the paucity of labeled data for speech summarization by introducing two new datasets – *SLUE-TED*, a dataset of TED talks where the task is to predict the abstract and title given the audio recording, and *Interview*, a dataset of multi-party radio interviews from the National Public Radio (NPR) for abstractive meeting summarization. We then develop a benchmark comparing end-to-end and cascade models on the 4 abstractive speech summarization corpora. Experiments show that end-to-end models can outperform cascade models on larger datasets, which mirrors observations made in speech translation research.

We also explore the task of human annotation for abstractive speech summarization. Annotators can either read a transcript of spoken content or listen to the audio recording to produce an abstractive speech summary, and we question whether differences exist between these approaches. We collect expert and non-expert annotations for 1002 recordings from the Interview test set and perform extensive analysis on the collected summaries. Analysis shows that speech-based summaries are more factually consistent and information selective, and result in higher inter-annotator agreement compared to transcript-based summaries. We also find that errors in the transcript degrade human annotation, and that expert annotators produce more informative and reliable summaries.

Next, we take on the task of improving the ability of end-to-end models to look over longer input contexts. We show that F-Net-based self-attentions can yield the best performance, and develop a new linear transformer based on a new XNOR kernel that improves performance over the state-of-the-art linear transformer, i.e., the Cosformer. To address the challenge of arbitrarily long recordings, we develop BASS: Blockwise Adaptation for Speech Summarization that re-frames speech summarization as a streaming problem. Experiments show that BASS not only improves efficiency and allows the use of arbitrarily long recordings but also helps improve model performance on video summarization. BASS can be in-efficient for non-streaming settings, and hence we introduce a relevance-aware version of BASS, R-BASS, to improve the efficiency of BASS by over 86% while retaining comparable performance.

Finally, we note that speech summarization models are constrained to produce summaries that are similar in structure to data used in model training. Further, though there exist multiple valid summaries for a given recording, current datasets and approaches rely on a single reference summary for training and evaluation. We address this limitation by prompting Large Language Models (LLMs) to generate augmentation summaries and describing methods to use these summaries for training and evaluation using our AugSumm approach. Experiments show that AugSumm can improve the quality of resulting summaries, on coherence, consistency, fluency, relevance, and lexical similarity.

In summary, in this thesis, we have demonstrated that end-to-end modeling is a viable and in many cases more performant alternative to the classical cascade approach. We have addressed various challenges within end-to-end models, and shown how end-to-end models can be improved.

9.2 Future directions for the field

The work done in this thesis has significant implications for the research and product development industries. It is my hope that this thesis motivates and inspires the next generation of researchers and technologists to contribute to the field of speech summarization. In this section, I list a few ideas to ponder for future work in speech summarization.

9.2.1 New Applications of Speech Summarization

Speech summarization has a myriad of applications, and existing corpora do not cover all such applications. The work in this thesis is largely application-agnostic, but using application-specific knowledge can further improve speech summarization. For example, in the medical domain, knowledge of medical terms and entities can significantly improve the prediction of doctor's notes.

In multi-party party settings, knowledge of meeting participants and overlapping information can improve the usefulness and quality of summaries. Within multi-lingual and code-switching environments, making summarization models aware of languages can enable cross-lingual summarization as well. All of these integrations can be performed in an end-to-end manner to enhance impact and retain relatively simple model architectures.

9.2.2 Custom Summarization with Multimodal Foundation Models

The state of artificial intelligence has advanced immensely with the development and deployment of large-scale foundation models that can do multiple tasks based on prompting. Such paradigms enable models to perform new tasks based on simple task descriptions, and that is the need of the hour.

All summaries are not the same, hence being able to customize the type and nature of summaries is essential. For example, for some applications, we may prefer extractive over abstractive summaries, and for others, longer summaries over shorter summaries. There may be cases where specific information is desired to be retained within the summary – for example, it may be desirable for meeting summaries to have information on meeting updates and tasks that need to be completed, while TED talk abstracts may not have this feature.

9.2.3 Explainable Summarization and Summarization Evaluation

An important challenge in summarization and machine learning more broadly is the lack of explainability within end-to-end models. It is important to understand what neural networks learn, and how they make output predictions. Another aspect could include using insights gained from human summarization to build explainable and interpretable end-to-end models for speech summarization.

Finally, there is no consensus on the approach for automatic summary evaluation and that remains an active area of research.

Bibliography

[Aher et al., 2023] Aher, G. V., Arriaga, R. I., and Kalai, A. T. (2023). Using large language models to simulate multiple humans and replicate human subject studies.

[Ainslie et al., 2020] Ainslie, J., Ontanon, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q., and Yang, L. (2020). ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

[Akula and Garibay, 2022] Akula, R. and Garibay, I. (2022). Sentence pair embeddings based evaluation metric for abstractive and extractive summarization. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Odijk, J., and Piperidis, S., editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6009–6017, Marseille, France. European Language Resources Association.

[Anderson, 1992] Anderson, J. D. (1992). Indexing and abstracting in theory and practice.

[Argyle et al., 2023] Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C., and Wingate, D. (2023). Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

[Ba et al., 2016] Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *CoRR*, abs/1607.06450.

[Banerjee and Lavie,] Banerjee, S. and Lavie, A. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. *Proceedings of ACL-WMT*, pages 65–72.

[Banerjee and Lavie, 2005a] Banerjee, S. and Lavie, A. (2005a). 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.

[Banerjee and Lavie, 2005b] Banerjee, S. and Lavie, A. (2005b). 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.

[Beltagy et al., 2020] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

[Bérard et al., 2016] Bérard, A., Pietquin, O., Servan, C., and Besacier, L. (2016). Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

[Bird and Loper, 2004] Bird, S. and Loper, E. (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.

[Borko and Bernier, 1975] Borko, H. and Bernier, C. L. (1975). Abstracting concepts and methods.

[Busso et al., 2008] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. In *Language resources and evaluation*.

[Calhoun et al., 2010] Calhoun, S., Carletta, J., Brenier, J. M., Mayo, N., Jurafsky, D., Steedman, M., and Beaver, D. (2010). The NXT-format Switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language resources and evaluation*.

[Cao et al., 2015] Cao, Z., Wei, F., Dong, L., Li, S., and Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *Twenty-ninth AAAI conference on artificial intelligence*.

[Carbonell and Goldstein, 1998] Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

[Chan et al., 2016] Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4960–4964. IEEE.

[Chan et al., 2021] Chan, W., Park, D., Lee, C., Zhang, Y., Le, Q., and Norouzi, M. (2021). Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*.

[Chen et al., 2020a] Chen, E. Y., Lu, Z., Xu, H., Cao, L., Zhang, Y., and Fan, J. (2020a). A large scale speech sentiment corpus. In *Language resources and evaluation*.

[Chen and Withgott, 1992] Chen, F. and Withgott, M. (1992). The use of emphasis to automatically summarize a spoken discourse. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 229–232 vol.1.

[Chen et al., 2022] Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

[Chen et al., 2015] Chen, X., Tan, T., Liu, X., Lanchantin, P., Wan, M., Gales, M. J., and Woodland, P. C. (2015). Recurrent neural network language model adaptation for multi-genre broadcast speech recognition. In *16th Annual Conference of the International Speech Communication Association*.

[Chen et al., 2020b] Chen, X. L., Ita Levitan, S., Levine, M., Mandic, M., and Hirschberg, J. (2020b). Acoustic-prosodic and lexical cues to deception and trust: deciphering how people detect lies. *Transactions of the Association for Computational Linguistics*, 8:199–214.

[Choromanski et al., 2021] Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., and Weller, A. (2021). Rethinking attention with performers. In *International Conference on Learning Representations*.

[Chorowski et al., 2015] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

[Christensen et al., 2004] Christensen, H., Kolluru, B., Gotoh, Y., and Renals, S. (2004). From text summarisation to style-specific summarisation for broadcast news. In *European Conference on Information Retrieval*, pages 223–237. Springer.

[Chung et al., 2022] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

[Clark et al., 2019] Clark, E., Celikyilmaz, A., and Smith, N. A. (2019). Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760.

[Cohan et al., 2022] Cohan, A., Feigenblat, G., Ghosal, T., and Shmueli-Scheuer, M. (2022). Overview of the first shared task on multi perspective scientific document summarization (mup). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 263–267.

[Cohn et al., 2019] Cohn, I., Laish, I., Beryozkin, G., Li, G., Shafran, I., Szpektor, I., Hartman, T., Hassidim, A., and Matias, Y. (2019). Audio de-identification: A new entity recognition task. In *NAACL*.

[Dai et al., 2019] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

[Deng et al., 2022] Deng, K., Watanabe, S., Shi, J., and Arora, S. (2022). Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 1746–1750.

[Denkowski and Lavie, 2014] Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380. Association for Computational Linguistics.

[Devlin et al., 2019a] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

[Devlin et al., 2019b] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Espejel, 2019] Espejel, J. L. (2019). Automatic summarization of medical conversations, a review. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume III: RECITAL*, pages 487–498.

[et. al., 2018] et. al., S. W. (2018). ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

[Fabbri et al., 2021a] Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021a). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

[Fabbri et al., 2021b] Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021b). SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

[Feng et al., 2020] Feng, H., Ueno, S., and Kawahara, T. (2020). End-to-end speech emotion recognition combined with acoustic-to-word asr model. In *Interspeech*, pages 501–505.

[Fisk and Hurst, 2003] Fisk, C. and Hurst, B. (2003). Paraphrasing for comprehension. *The Reading Teacher*, 57:182–185.

[Galley, 2006] Galley, M. (2006). A skip-chain conditional random field for ranking meeting utterances by importance.

[Gao et al., 2023] Gao, M., Ruan, J., Sun, R., Yin, X., Yang, S., and Wan, X. (2023). Human-like Summarization Evaluation with ChatGPT. *arXiv preprint arXiv:2304.02554*.

[Gao et al., 2020] Gao, Y., Zhao, W., and Eger, S. (2020). Supert: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. *arXiv preprint arXiv:2005.03724*.

[Gillick and Liu, 2010] Gillick, D. and Liu, Y. (2010). Non-expert evaluation of summarization systems is risky. In Callison-Burch, C. and Dredze, M., editors, *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151, Los Angeles. Association for Computational Linguistics.

[Gong and Liu, 2001] Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 19–25, New York, NY, USA. Association for Computing Machinery.

[Graves, 2012] Graves, A. (2012). Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

[Graves and Jaitly, 2014] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR.

[Gu et al., 2019] Gu, R., Wu, J., Zhang, S.-X., Chen, L., Xu, Y., Yu, M., Su, D., Zou, Y., and Yu, D. (2019). End-to-end multi-channel speech separation. *arXiv preprint arXiv:1905.06286*.

[Gulati et al., 2020a] Gulati, A., Chiu, C.-C., Qin, J., Yu, J., Parmar, N., Pang, R., Wang, S., Han, W., Wu, Y., Zhang, Y., and Zhang, Z., editors (2020a). *Conformer: Convolution-augmented Transformer for Speech Recognition*.

[Gulati et al., 2020b] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. (2020b). Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

[Guo et al., 2021] Guo, P., Boyer, F., Chang, X., Hayashi, T., Higuchi, Y., Inaguma, H., Kamo, N., Li, C., Garcia-Romero, D., Shi, J., et al. (2021). Recent developments on espnet toolkit boosted by conformer.

[Hemphill et al., 1990] Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. (1990). The ATIS spoken language systems pilot corpus. In *Speech and Natural Language*.

[Hirohata et al., 2005] Hirohata, M., Shinnaka, Y., Iwano, K., and Furui, S. (2005). Sentence extraction-based presentation summarization techniques and evaluation metrics. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–1065. IEEE.

[Honnibal and Montani, 2017] Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear.*

[Hori et al., 2002] Hori, C., Furui, S., Malkin, R., Yu, H., and Waibel, A. (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.

[Hori et al., 2003] Hori, C., Hori, T., and Furui, S. (2003). Evaluation method for automatic speech summarization. In *INTERSPEECH*.

[Hori et al., 2020] Hori, T., Moritz, N., Hori, C., and Roux, J. L. (2020). Transformer-based long-context end-to-end speech recognition. In Meng, H., Xu, B., and Zheng, T. F., editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5011–5015. ISCA.

[Hori et al., 2021] Hori, T., Moritz, N., Hori, C., and Roux, J. L. (2021). Advanced long-context end-to-end speech recognition using context-expanded transformers.

[Hovy et al., 2006] Hovy, E., Lin, C.-Y., Zhou, L., and Fukumoto, J. (2006). Automated summarization evaluation with basic elements. In Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odijk, J., and Tapias, D., editors, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

[Huang et al., 2022] Huang, Z., Rao, M., Raju, A., Zhang, Z., Bui, B., and Lee, C. (2022). MTL-SLT: Multi-task learning for spoken language tasks. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 120–130, Dublin, Ireland. Association for Computational Linguistics.

[Inoue et al., 2004] Inoue, A., Mikami, T., and Yamashita, Y. (2004). Improvement of speech summarization using prosodic information. In *Speech Prosody 2004, International Conference*.

[Janin et al., 2003] Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. (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.

[Jurafsky et al., 1998] Jurafsky, D., Shriberg, E., Fox, B., and Curl, T. (1998). Lexical, prosodic, and syntactic cues for dialog acts. In *Discourse Relations and Discourse Markers*.

[Kano et al., 2023a] Kano, T., Ogawa, A., Delcroix, M., Sharma, R., Matsuura, K., and Watanabe, S. (2023a). 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.

[Kano et al., 2023b] Kano, T., Ogawa, A., Delcroix, M., Sharma, R., Matsuura, K., and Watanabe, S. (2023b). 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.

[Kano et al., 2021a] Kano, T., Ogawa, A., Delcroix, M., and Watanabe, S. (2021a). Attention-based multi-hypothesis fusion for speech summarization. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 487–494.

[Kano et al., 2021b] Kano, T., Ogawa, A., Delcroix, M., and Watanabe, S. (2021b). Attention-based multi-hypothesis fusion for speech summarization. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 487–494.

[Kano et al., 2021c] Kano, T., Ogawa, A., Delcroix, M., and Watanabe, S. (2021c). Attention-based multi-hypothesis fusion for speech summarization. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 487–494.

[Kano et al., 2021d] Kano, T., Ogawa, A., Delcroix, M., and Watanabe, S. (2021d). Attention-based multi-hypothesis fusion for speech summarization.

[Katharopoulos et al., 2020] Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[Kenton and Toutanova, 2019] Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

[Khalitov et al., 2023] Khalitov, R., Yu, T., Cheng, L., and Yang, Z. (2023). Chordmixer: A scalable neural attention model for sequences with different length. In *The Eleventh International Conference on Learning Representations*.

[Khan et al., 2022] Khan, A. A., Nawaz, S., Newn, J., Kelly, R. M., Lodge, J. M., Bailey, J., and Velloso, E. (2022). To type or to speak? the effect of input modality on text understanding during note-taking. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

[Kim et al., 2017] Kim, S., Hori, T., and Watanabe, S. (2017). Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 4835–4839. IEEE.

[Kim and Metze, 2018] Kim, S. and Metze, F. (2018). Dialog-context aware end-to-end speech recognition. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 434–440. IEEE.

[Kitaev et al., 2020] Kitaev, N., Kaiser, L., and Levskaya, A. (2020). Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[Koumpis and Renals, 2005] Koumpis, K. and Renals, S. (2005). Automatic summarization of voicemail messages using lexical and prosodic features. *ACM Transactions on Speech and Language Processing (TSLP)*, 2(1):1–es.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Lee-Thorp et al., 2022a] Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. (2022a). FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

[Lee-Thorp et al., 2022b] Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontanon, S. (2022b). Fnet: Mixing tokens with fourier transforms.

[Lewis et al., 2019] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

[Lewis, 2020] Lewis, M. e. a. (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.

[Li et al., 2019] Li, H., Zhu, J., Ma, C., Zhang, J., and Zong, C. (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.

[Libovický and Helcl, 2017] Libovický, J. and Helcl, J. (2017). Attention strategies for multi-source sequence-to-sequence learning. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.

[Lin, 2004a] Lin, C.-Y. (2004a). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Lin, 2004b] Lin, C.-Y. (2004b). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[Lin and Och, 2004] Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 605–612, Barcelona, Spain.

[Lin et al., 2009a] Lin, H., Bilmes, J., and Xie, S. (2009a). Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386. IEEE.

[Lin et al., 2009b] Lin, S.-H., Lo, Y.-T., Yeh, Y.-M., and Chen, B. (2009b). Hybrids of supervised and unsupervised models for extractive speech summarization. In *Tenth Annual Conference of the International Speech Communication Association*.

[Lippmann et al., 1987] Lippmann, R., Martin, E., and Paul, D. (1987). Multi-style training for robust isolated-word speech recognition. In *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 705–708.

[Liu* et al., 2018] Liu*, P. J., Saleh*, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

[Liu et al., 2020] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2020). Ro{bert}a: A robustly optimized {bert} pretraining approach.

[Lu* et al., 2019] Lu*, Y., Li*, Z., He, D., Sun, Z., Dong, B., Qin, T., Wang, L., and yan Liu, T. (2019). Understanding and improving transformer from a multi-particle dynamic system point of view. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

[Luo et al., 2023] Luo, Z., Xie, Q., and Ananiadou, S. (2023). ChatGPT as a Factual Inconsistency Evaluator for Text Summarization.

[Ma et al., 2021] Ma, X., Wang, Y., Dousti, M. J., Koehn, P., and Pino, J. (2021). Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7523–7527.

[Majumder et al., 2020] Majumder, B. P., Li, S., Ni, J., and McAuley, J. (2020). Interview: Large-scale modeling of media dialog with discourse patterns and knowledge grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8129–8141, Online. Association for Computational Linguistics.

[Manakul et al., 2020] Manakul, P., Gales, M. J., and Wang, L. (2020). Abstractive Spoken Document Summarization Using Hierarchical Model with Multi-Stage Attention Diversity Optimization. In *Proc. Interspeech 2020*, pages 4248–4252.

[Mani, 1999] Mani, I. (1999). *Advances in automatic text summarization*. MIT press.

[Mani, 2001] Mani, I. (2001). *Automatic summarization*, volume 3. John Benjamins Publishing.

[Martinez-Lucas et al., 2020] Martinez-Lucas, L., Abdelwahab, M., and Busso, C. (2020). The MSP-Conversation corpus. In *INTERSPEECH*.

[Maskey and Hirschberg, 2006] Maskey, S. and Hirschberg, J. (2006). Summarizing speech without text using hidden markov models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 89–92.

[Matsuura et al., 2023a] Matsuura, K., Ashihara, T., Moriya, T., Tanaka, T., Kano, T., Ogawa, A., and Delcroix, M. (2023a). Transfer learning from pre-trained language models improves end-to-end speech summarization.

[Matsuura et al., 2023b] Matsuura, K., Ashihara, T., Moriya, T., Tanaka, T., Ogawa, A., Delcroix, M., and Masumura, R. (2023b). 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.

[Matsuura et al., 2023c] Matsuura, K., Ashihara, T., Moriya, T., Tanaka, T., Ogawa, A., Delcroix, M., and Masumura, R. (2023c). 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.

[Matsuura et al., 2023d] Matsuura, K., Ashihara, T., Moriya, T., Tanaka, T., Ogawa, A., Delcroix, M., and Masumura, R. (2023d). Leveraging large text corpora for end-to-end speech summarization.

[McCowan et al., 2005] McCowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D., and Wellner, P. (2005). The ami meeting corpus. In Noldus, L., Grieco, F., Loijens, L., and Zimmerman, P., editors, *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, pages 137–140. Noldus Information Technology.

[Mikolov et al., 2013] Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Sentence embeddings for semantic composition. In *Advances in Neural Information Processing Systems*, pages 3942–3950.

[Moritz et al., 2020] Moritz, N., Hori, T., and Le, J. (2020). Streaming automatic speech recognition with the transformer model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE.

[Moritz et al., 2021] Moritz, N., Hori, T., and Le Roux, J. (2021). Capturing multi-resolution context by dilated self-attention. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5869–5873.

[Murray et al., 2010] Murray, G., Carenini, G., and Ng, R. (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.

[Nallapati et al., 2016a] Nallapati, R., Zhou, B., dos Santos, C., glar Gulçehre, Ç., and Xiang, B. (2016a). Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, page 280.

[Nallapati et al., 2016b] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016b). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

[Narayan et al., 2018] Narayan, S., Cohen, S. B., and Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. *CoRR*.

[Narayanan et al., 2019] Narayanan, A., Prabhavalkar, R., Chiu, C.-C., Rybach, D., Sainath, T. N., and Strohman, T. (2019). Recognizing long-form speech using streaming end-to-end models. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 920–927.

[Nenkova and Passonneau, 2004] Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

[Nenkova et al., 2007] Nenkova, A., Passonneau, R., and McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es.

[Ng and Abrecht, 2015] Ng, J.-P. and Abrecht, V. (2015). Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*.

[Ouyang et al., 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

[Palaskar et al., 2019a] Palaskar, S., Libovický, J., Gella, S., and Metze, F. (2019a). 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.

[Palaskar et al., 2019b] Palaskar, S., Libovický, J., Gella, S., and Metze, F. (2019b). 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.

[Palaskar et al., 2020] Palaskar, S., Salakhutdinov, R., Black, A. W., and Metze, F. (2020). Learning semantic concepts for video understanding. In *Under Review*.

[Palaskar et al., 2021a] Palaskar, S., Salakhutdinov, R., Black, A. W., and Metze, F. (2021a). Multimodal Speech Summarization Through Semantic Concept Learning. In *Proc. Interspeech 2021*, pages 791–795.

[Palaskar et al., 2021b] Palaskar, S., Salakhutdinov, R., Black, A. W., and Metze, F. (2021b). Multimodal speech summarization through semantic concept learning. pages 791–795.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Park et al., 2019a] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019a). Specaugment: A simple augmentation method for automatic speech recognition. In *INTERSPEECH*.

[Park et al., 2019b] Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019b). SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

[Peyrard et al., 2017] Peyrard, M., Botschen, T., and Gurevych, I. (2017). Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84.

[Popović, 2015] Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

[Qin et al., 2022] Qin, Z., Sun, W., Deng, H., Li, D., Wei, Y., Lv, B., Yan, J., Kong, L., and Zhong, Y. (2022). cosformer: Rethinking softmax in attention. In *International Conference on Learning Representations*.

[Radford et al., 2022] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *OpenAI Blog*.

[Rae et al., 2020] Rae, J. W., Potapenko, A., Jayakumar, S. M., Hillier, C., and Lillicrap, T. P. (2020). Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[Raffel et al., 2020a] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020a). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

[Raffel et al., 2020b] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020b). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

[Rao et al., 2017] Rao, K., Sak, H., and Prabhavalkar, R. (2017). Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 193–199. IEEE.

[Rath et al., 1961] Rath, G., Resnick, A., and Savage, T. R. (1961). The formation of abstracts by the selection of sentences. part i. sentence selection by men and machines. *American Documentation*, 12(2):139–141.

[Rezazadegan et al., 2020] Rezazadegan, D., Berkovsky, S., Quiroz, J. C., Kocaballi, A. B., Wang, Y., Laranjo, L., and Coiera, E. (2020). Automatic speech summarisation: A scoping review. *arXiv preprint arXiv:2008.11897*.

[Rush et al., 2015] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

[Sanabria et al., 2018] Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., and Metze, F. (2018). How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*.

[Schuller et al., 2009] Schuller, B., Batliner, A., Steidl, S., and Seppi, D. (2009). Emotion recognition from speech: putting asr in the loop. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4585–4588. IEEE.

[Scialom et al., 2019] Scialom, T., Lamprier, S., Piwowarski, B., and Staiano, J. (2019). Answers unite! unsupervised metrics for reinforced summarization models. *arXiv preprint arXiv:1909.01610*.

[See et al., 2017] See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

[Sellam et al., 2020] Sellam, T., Das, D., and Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

[Serdyuk et al., 2018] Serdyuk, D., Wang, Y., Fuegen, C., Kumar, A., Liu, B., and Bengio, Y. (2018). Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.

[Shang et al., 2018] Shang, G., Ding, W., Zhang, Z., Tixier, A. J.-P., Meladianos, P., Vazirgiannis, M., and Lorré, J.-P. (2018). Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. *arXiv preprint arXiv:1805.05271*.

[Sharma et al., 2023a] Sharma, R., Chen, W., Kano, T., Sharma, R., Arora, S., Watanabe, S., Ogawa, A., Delcroix, M., Singh, R., and Raj, B. (2023a). Espnet-summ: Introducing a novel large dataset, toolkit, and a cross-corpora evaluation of speech summarization systems. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

[Sharma et al., 2022a] Sharma, R., Palaskar, S., Black, A. W., and Metze, F. (2022a). 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.

[Sharma et al., 2022b] Sharma, R., Palaskar, S., Black, A. W., and Metze, F. (2022b). 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. IEEE.

[Sharma and Raj, 2022] Sharma, R. and Raj, B. (2022). Xnor-former: Learning accurate approximations in long speech transformers. *arXiv preprint arXiv:2210.16643*.

[Sharma et al., 2023b] Sharma, R., Zheng, K., Arora, S., Watanabe, S., Singh, R., and Raj, B. (2023b). Bass: Block-wise adaptation for speech summarization.

[Shen et al., 2023] Shen, C., Cheng, L., Nguyen, X.-P., You, Y., and Bing, L. (2023). Large Language Models are Not Yet Human-Level Evaluators for Abstractive Summarization. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[Shi et al., 2021] Shi, Y., Wang, Y., Wu, C., Yeh, C.-F., Chan, J., Zhang, F., Le, D., and Seltzer, M. (2021). Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6783–6787.

[Shivakumar et al., 2019] Shivakumar, P. G., Yang, M., and Georgiou, P. (2019). Spoken language intent detection using confusion2vec. *arXiv preprint arXiv:1904.03576*.

[Shon et al., 2022] Shon, S., Arora, S., Lin, C.-J., Pasad, A., Wu, F., Sharma, R., Wu, W.-L., Lee, H.-Y., Livescu, K., and Watanabe, S. (2022). Slue phase-2: A benchmark suite of diverse spoken language understanding tasks. *arXiv preprint arXiv:2212.10525*.

[Shon et al., 2023] Shon, S., Arora, S., Lin, C.-J., Pasad, A., Wu, F., Sharma, R. S., Wu, W.-L., Lee, H.-y., Livescu, K., and Watanabe, S. (2023). SLUE phase-2: A benchmark suite of diverse spoken language understanding tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8906–8937, Toronto, Canada. Association for Computational Linguistics.

[Sorensen et al., 2022] Sorensen, T., Robinson, J., Rytting, C., Shaw, A., Rogers, K., Delorey, A., Khalil, M., Fulda, N., and Wingate, D. (2022). An information-theoretic approach to prompt engineering without ground truth labels. In *Association for Computational Linguistics*. Association for Computational Linguistics.

[Stollnberger et al., 2013] Stollnberger, G., Weiss, A., and Tscheligi, M. (2013). Input modality and task complexity: Do they relate? In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 233–234. IEEE.

[Touvron et al., 2023a] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023a). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[Touvron et al., 2023b] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

[Tran et al., 2018] Tran, T., Toshniwal, S., Bansal, M., Gimpel, K., Livescu, K., and Ostendorf, M. (2018). Parsing speech: A neural approach to integrating lexical and acoustic-prosodic information. In *Proceedings of NAACL-HLT*.

[Tsunoo et al., 2021] Tsunoo, E., Kashiwagi, Y., and Watanabe, S. (2021). Streaming transformer asr with blockwise synchronous beam search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 22–29.

[Vasilyev et al., 2020] Vasilyev, O., Dharnidharka, V., and Bohannon, J. (2020). Fill in the blanc: Human-free quality estimation of document summaries. *arXiv preprint arXiv:2002.09836*.

[Vaswani et al., 2017a] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017a). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

[Vaswani et al., 2017b] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on*

Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

[Vedantam et al., 2015] Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

[Wang et al., 2020] Wang, S., Li, B., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

[Wang et al., 2023] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. (2023). Self-instruct: Aligning language models with self-generated instructions. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

[Wang et al., 2021] Wang, Y., Lv, H., Povey, D., Xie, L., and Khudanpur, S. (2021). Wake word detection with streaming transformers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5864–5868. IEEE.

[Watanabe et al., 2017] Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi, T. (2017). Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

[White et al., 2023] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

[Wolf, 2020] Wolf, T. e. a. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

[Woodsend and Lapata, 2012] Woodsend, K. and Lapata, M. (2012). Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243.

[Wu et al., 2023] Wu, N., Gong, M., Shou, L., Liang, S., and Jiang, D. (2023). Large Language Models are Diverse Role-Players for Summarization Evaluation. *arXiv preprint arXiv:2303.15078*.

[Xiong et al., 2017] Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M. L., Stolcke, A., Yu, D., and Zweig, G. (2017). Toward human parity in conversational speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2410–2423.

[Yadav et al., 2020] Yadav, H., Ghosh, S., Yu, Y., and Shah, R. R. (2020). End-to-end named entity recognition from English speech. In *INTERSPEECH*.

[Yasunaga et al., 2017] Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., and Radev, D. R. (2017). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462.

[Ye et al., 2022] Ye, J., Gao, J., Li, Q., Xu, H., Feng, J., Wu, Z., Yu, T., and Kong, L. (2022). ZeroGen: Efficient zero-shot learning via dataset generation. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Yu et al., 2021] Yu, T., Dai, W., Liu, Z., and Fung, P. (2021). Vision guided generative pre-trained language models for multimodal abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3995–4007, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Yu et al., 2023a] Yu, Y., Zhuang, Y., Zhang, J., Meng, Y., Ratner, A., Krishna, R., Shen, J., and Zhang, C. (2023a). Large language model as attributed training data generator: A tale of diversity and bias.

[Yu et al., 2023b] Yu, Y., Zhuang, Y., Zhang, R., Meng, Y., Shen, J., and Zhang, C. (2023b). ReGen: Zero-shot text classification via training data generation with progressive dense retrieval. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11782–11805, Toronto, Canada. Association for Computational Linguistics.

[Yuan et al., 2021] Yuan, W., Neubig, G., and Liu, P. (2021). BARTScore: Evaluating generated text as text generation. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

[Zadeh et al., 2018] Zadeh, A., Liang, P. P., Vanbriesen, J., Poria, S., Tong, E., Cambria, E., Chen, M., and Morency, L. P. (2018). Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *ACL*.

[Zaheer et al., 2020] Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020). Big bird: Transformers for longer sequences. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

[Zhang et al., 2007] Zhang, J., Chan, H. Y., Fung, P., and Cao, L. (2007). A comparative study on speech summarization of broadcast news and lecture speech. In *Eighth Annual Conference of the International Speech Communication Association*.

[Zhang et al., 2009] Zhang, J. J., Chan, R. H. Y., and Fung, P. (2009). Extractive speech summarization using shallow rhetorical structure modeling. *IEEE transactions on audio, speech, and language processing*, 18(6):1147–1157.

[Zhang et al., 2019] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[Zhang et al., 2020] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[Zhang* et al., 2020] Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

[Zhao et al., 2019] Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., and Eger, S. (2019). MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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 563–578, Hong Kong, China. Association for Computational Linguistics.

[Zhong et al., 2022a] Zhong, M., Liu, Y., Yin, D., Mao, Y., Jiao, Y., Liu, P., Zhu, C., Ji, H., and Han, J. (2022a). Towards a unified multi-dimensional evaluator for text generation.

[Zhong et al., 2022b] Zhong, M., Liu, Y., Yin, D., Mao, Y., Jiao, Y., Liu, P., Zhu, C., Ji, H., and Han, J. (2022b). Towards a unified multi-dimensional evaluator for text generation. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Zhou et al., 2006] Zhou, L., Lin, C.-Y., Munteanu, D. S., and Hovy, E. (2006). Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the human language technology conference of the NAACL, main conference*, pages 447–454.

[Zhu et al., 2021] Zhu, C., Liu, Y., Mei, J., and Zeng, M. (2021). MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

[Zhu et al., 2020] Zhu, C., Xu, R., Zeng, M., and Huang, X. (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.

[Zhu and Penn, 2006] Zhu, X. and Penn, G. (2006). Comparing the roles of textual, acoustic and spoken-language features on spontaneous-conversation summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 197–200.