Enabling Structured Navigation of Longform Spoken Dialog with Automatic Summarization

Daniel Li

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

Abstract

Enabling Structured Navigation of Longform Spoken Dialog with Automatic Summarization

Daniel Li

Longform spoken dialog is a rich source of information that is present in all facets of everyday life, taking the form of podcasts, debates, and interviews; these mediums contain important topics ranging from healthcare and diversity to current events, economics and politics. Individuals need to digest informative content to know how to vote, decide how to stay safe from COVID-19, and how to increase diversity in the workplace.

Unfortunately compared to text, spoken dialog can be challenging to consume as it is slower than reading and difficult to skim or navigate. Although an individual may be interested in a given topic, they may be unwilling to commit the required time necessary to consume long form auditory media given the uncertainty as to whether such content will live up to their expectations. Clearly, there exists a need to provide access to the information spoken dialog provides in a manner through which individuals can quickly and intuitively access areas of interest without investing large amounts of time.

From Human Computer Interaction, we apply the idea of information foraging, which theorizes how people browse and navigate to satisfy an information need, to the longform spoken dialog domain. Information foraging states that people do not browse linearly. Rather people “forage” for information similar to how animals sniff around for food, scanning from area to area, constantly deciding whether to keep investigating their current area or to move on to greener pastures. This is an instance of the classic breadth vs. depth dilemma. People rely on perceived structure and

information cues to make these decisions. Unfortunately speech, either spoken or transcribed, is unstructured and lacks information cues, making it difficult for users to browse and navigate.

We create a longform spoken dialog browsing system that utilizes automatic summarization and speech modeling to structure longform dialog to present information in a manner that is both intuitive and flexible towards different user browsing needs. Leveraging summarization models to automatically and hierarchically structure spoken dialog, the system is able to distill information into increasingly salient and abstract summaries, allowing for a tiered representation that, if interested, users can progressively explore. Additionally, we address spoken dialog's own set of technical challenges to speech modeling that are not present in written text, such as disfluencies, improper punctuation, lack of annotated speech data, and inherent lack of structure.

We create a longform spoken dialog browsing system that utilizes automatic summarization and speech modeling to structure longform dialog to present information in a manner that is both intuitive and flexible towards different user browsing needs. Leveraging summarization models to automatically and hierarchically structure spoken dialog, the system is able to distill information into increasingly salient and abstract summaries, allowing for a tiered representation that, if interested, users can progressively explore. Additionally, we address spoken dialog's own set of technical challenges to speech modeling that are not present in written text, such as disfluencies, improper punctuation, lack of annotated speech data, and inherent lack of structure. Since summarization is a lossy compression of information, the system provides users with information cues to signal how much additional information is contained on a topic.

This thesis makes the following contributions:

1. We applied the HCI concept of information foraging to longform speech, enabling people to browse and navigate information in podcasts, interviews, panels, and meetings.

2. We created a system that structures longform dialog into hierarchical summaries which help users to 1) skim (browse) audio and 2) navigate and drill down into interesting sections to read full details.

3. We created a human annotated hierarchical dataset to quantitatively evaluate the effectiveness of our system's hierarchical text generation performance.

4. Lastly, we developed a suite of dialog oriented processing optimizations to improve the user experience of summaries: enhanced readability and fluency of short summaries through better topic chunking and pronoun imputation, and reliable indication of semantic coverage within short summaries to help direct navigation towards interesting information.

We discuss future research in extending the browsing and navigating system to more challenging domains such as lectures, which contain many external references, or workplace conversations, which contain uncontextualized background information and are far less structured than podcasts and interviews.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Thanks to Prof. Lydia Chilton for being an awesome advisor for being patient with me (especially when I was not motivated to do research) and providing research direction; I could not ask for a better advisor. Next, I would like to thank Prof. Satish Rao from UC Berkeley for believing in me and that I could still do computer science (despite my *non-trivially* below average CS70 midterm scores).

Thanks to everyone I worked with in industry (in order of recency) and mentored me on proper research direction:

1. Google Cloud AI Translation: Junpei Zhou and Music Li, for their direction on how research interfaces with client facing products and needs.

2. Google Translate: Te I, Naveen Arivazhagan, Colin Cherry, and Dirk Padfield, for their direction on proper experimentation in research.

3. NEC Labs: Asim Kadav, for taking a chance on me when I knew nothing about deep learning and being an excellent role model for a young intern. Working with Asim gave me my first in depth research experience, who supported and guided every step of the way.

Lastly, I could not have done all of this without the help (though help may be an understatement) of my friends: Thomas Chen, Albert Tung, Alec Zadikian, Andrew Liu, Alan Tang, Matthew Deng, and Marc WuDunn.

Dedication

This one goes to the lads; all of you were instrumental and helped me every step of the way. Without you guys this wouldn't have been possible. You have all taken more than your fair share of undue abuse and then some. I hope you all know that I would be willing to do the same for each of you on a dime's notice and without any hesitation.

I'm sure you will support me the same when just the same in the next stage of my life: jumping out of planes, kicking in doors, breaching compounds, and eating the general suck. Whatever is next, this document's existence would not be possible without you all, and is a living testament to the love and support I received through my Ph.D. I will never forget that.

Chapter 1: Introduction

Spoken dialog is a rich source of information that is present in all facets of our everyday lives. Speech is a medium that is native and relatable - we communicate and share ideas, debate topics, and express ourselves through speech. Take for example a televised political debate where two opponents engage in heated discussion over policy regarding immigration, or in a different setting where a company is discussing its quarterly earnings and its future direction through a panel call; they both contain rich information with varying degrees of individualized relevance towards different listeners. Similar speech also extends to many spoken settings ranging from city council meetings, project discussions, doctor's appointments, seminar based commentary, and dictated instructions. Undoubtedly, conversation and dialog is content rich and diverse. Creating such language applications for navigating and modeling speech for more efficient understanding has many direct and immediate user benefits.

One of the largest challenges with longform audio is the difficulty in navigating and exploring the medium [1, 2, 3]. Speech is intrinsically disorganized and lacks traditional structure such as paragraphs, topic sentences, and proper punctuation that is present in traditional prose and written text. Such information cues that normally can be used to assist users to more quickly identify relevant information are non-existent. Furthermore, speech can often be non-informative as is in the case of greetings and banter, resulting in unnecessary details that are not necessary or pertinent towards understanding the underlying content.

To demonstrate the problem with navigation in a real world setting, consider an example where you're in a teaching position at a learning institution. A colleague emails you a 30 minute YouTube video containing a recent interview on diversity and inclusion. This is an area that you're not an expert on, but as an educator, is something you're interested in. This is something that you might find interesting, but not to the extent that you'd be willing to invest 25 minutes into. What's more is that there is no easy way to skim through and navigate through the interview's content to find parts that you find interesting. Some strategies that you might be left with include:

1. Randomly skipping throughout the video.
2. Listening for a few seconds periodically and making an educated guess on what you think each segment discusses.
3. The obvious method of simply listening through the entire interview (potentially in a sped up manner).
4. Obtaining a transcript (presented without any punctuation, capitalization, or paragraph formatting) and reading through.

For options 1 and 2, this could lead to frustration due to missed information and the lack of user awareness in navigating the interview content - consider the user's navigation choice in option 3, this somewhat guarantees no missed content and provides an indicator as to how far the user has progressed into the interview's content. The tradeoff in option 3, however, is too much of a time investment from the user and a cognitive overload if listened to on a sped up timeframe.[1] Lastly, in option 4 there is no structure nor punctuation as to how information is presented, which in turn does not offer a clear navigation procedure. Clearly, there exists a need to provide a tool or system that helps users more easily skim audio content *within an audio file* and quickly identify parts that they are interested in.

From the brief example, it becomes immediately apparent what criteria must be met to have a successful speech content navigation system:

1. **Importance and Relevance:** Content needs to be succinctly presented in a manner that preserves the most important and salient information. Irrelevant information (such as filler sentences in speech) should be summarized out.

[1]For the purposes of this work we do not consider this option, we use automatic speech recognition to distill audio into a text transcript and then operate on said transcript for the entirety of this dissertation. References to speech, longform spoken dialog, audio, and etc. are to their transcribed (transcript) form.

2. **User Tailored:** Information that is interesting to one user may not necessarily be interesting to another user. The proposed system should enable present information in a manner that supports user browsing freedom.

3. **User Needs:** Different users have different browsing objectives, ranging from a quick look-through to an in depth structured exploration. The proposed system should be able to support different styles of browsing.

With these criteria in mind, this thesis addresses the application need by developing a system[2] that utilizes automatic summarization and speech modeling to structure longform dialog to present information in a manner that is both intuitive and flexible towards different user browsing needs. Leveraging summarization models to automatically and hierarchically structure spoken dialog, the system is able to distill information into increasingly salient and abstract summaries (which can be intuitively thought of as a *long summary*, *medium summary*, and *short summary*), allowing for a tiered representation that, if interested, users can progressively explore.

Additionally, when designing such a system, we need to contemplate how users intuitively browse and search for information to satisfy an information need by examining different patches of information. This is described by a concept in human computer interaction known as *information foraging* [4], which we use to guide our system design. Consequently, our system's summaries of longform spoken dialog mimic semantically cohesive "patches" of information that can be individually browsed. With separate patches, the system is now able to impose a hierarchical structure on the underlying longform speech content.

The hierarchically presented nature of the summaries affords flexibility by allowing users to be in control of what information they read at a high level and what information they choose to consume in greater detail. We aim to structure and format spoken dialogue in a way that allows users to fit the cognitive tasks of information foraging with the following concepts:

1. The system begins with providing an **overview** of the different topics discussed in a long-

[2]For reader clarity, any time this dissertation mentions something along the lines of "(our) system", it is referring to the overall longform speech browsing system that we developed.

form audio file. By reducing the total amount of information and abstracting it into distinct concepts, users can quickly scan for pertinent high level information.

2. When a user finds interesting information and wishes to learn more, the system needs to **support user freedom** by allowing the user to drill down into the identified topic for a deeper dive into the corresponding finer details.

3. However, if at any point, if the desired information is not found in the user's particular navigational path, the system also needs to support the user in **popping back up**. Users must be able to reorient themselves to find the relevant information that satisfies their information need.

4. Creating information cues through information estimation heuristics to provide users with means to inform and guide content browsing.

By supporting these actions, we are able to present longform dialog content to users in a manner that not only enables navigation, but also reflects tailored user information goals. Simultaneously, we address technical speech domain characteristics (such as disfluencies, incorrect sentence boundaries, and word recognition errors) and the unique challenges they pose towards speech and language processing.

1.1 Thesis and Dissertation Contributions

Concisely stated, this dissertation proposes using HCI principles of information foraging to design a system that leverages automatic summarization to structure longform audio transcripts and create information cues to enable navigating and browsing of longform spoken dialog. This contributions of this dissertation are as follows:

First, *we apply the HCI concept of information foraging to longform speech, enabling people to browse and navigate information in podcasts, interviews, panels, and meetings*. Before even designing a system for navigating and browsing of longform audio, it is important to understand how people browse for information. People do not browse linearly, rather they “forage” for information similar to animals sniffing around for food, scanning from area to area. The underlying hypothesis is that users have some innate foraging mechanisms that guide them in a rational and goal-driven manner through different levels of specificity [5]. When searching for relevant information, users rely on the concept of “information scent” which they use to estimate how much useful information is contained on any given path, and how to adjust and reorient themselves as necessary to retrieve relevant information. A successful system that allows users to effectively browse longform speech content must consider and incorporate user browsing habits.

Second, *we accordingly introduce an approach that recursively applies automatic summarization to create hierarchical summaries, thereby condensing longform dialog to help users to 1) skim (browse) audio and 2) navigate and drill down into interesting sections to read full details.* The system employs abstractive summarization language models to recursively process and structure longform dialog into summaries containing multiple levels of detail, with the most high level summary (or *short summary*) containing only the most important key concepts and the most detailed level summary (or *long summary*) containing additional information, such as supporting facts to a central point. The system's user interface presents users the summaries in a hierarchical fashion, leading the user's attention to the shortest summaries, presented in a column. These can then used quickly skim all the high level pieces of information discussed. Because the short summary retains the segment's most salient information, it gives the best indication to the user on whether or not this segment contains the information that the user wants; if the user decides the information is interesting, they can proceed to read subsequent levels of detail until their "information need" [4] (in other words, how content is the user with their current understanding of the presented content) is met. The user interface also provides additional information cues [6], such as visual indicators estimating the amount of content contained and captured in summaries to better assist user browsing and navigation decisions.

Third, *we created the first human annotated hierarchical summarization dataset, providing gold standard intermediate text level summary annotations. This is then used to quantitatively eval-*

uate the effectiveness of our system's performance against these human written gold summaries. To the best of our knowledge, no abstractive summarization datasets exist for longform speech, much less annotated with multiple levels of summarization detail. To provide additional levels of annotations, we create a hierarchical abstractive summarization dataset and protocol (complete with an live open-sourced user interface annotation tool[3]), where longform text is recursively summarized to create progressively broader summaries in a bottom up manner. In addition to user studies and qualitative evaluation, this heirarchically annotated dataset[4] facilitates quantitative evaluation our system's summary quality performance for various summary detail levels with conventional metrics such as ROUGE [7].

Fourth *and lastly, we introduce a suite of dialog oriented processing optimizations to improve the user experience of summaries: enhanced readability and fluency of short summaries through better topic chunking and pronoun imputation, and reliable indication of semantic coverage within short summaries to help direct navigation towards interesting information*. Unlike written text, transcribed speech is far more noisy and difficult to process and model; speech contains informal language and disfluencies such as hesitation and vocal fillers, and discussions frequently jump from topic to topic. These speech specific attributes makes it challenging for summarization language models to semantically comprehend and generate high quality summaries, leading to user frustration when extra time and energy is spent trying to comprehend the AI generated summary. By specifically modeling speech noise effects and correcting input transcript text before summarization, the system is able to minimize detrimental speech effects.

1.2 Dissertation Overview

This section presents an outline of the dissertation by chapters.

[3]`https://resubstrate.github.io/lex-client/index.html#/login`

[4]The full scope of this dataset can be of wide interest to the overall NLP community and not immediately relevant to this thesis. We plan on additional development for a this dataset.

1.2.1 Related Work

The Related Work (Chapter 2) is used to properly contextualize and situate this dissertation and its contributions in the cross section of Human Computer Interaction (HCI) and Natural Language Processing (NLP). Specifically, we examine principles from user browsing habits (information foraging) and how to apply them when designing a system to help users browse and navigate longform dialog.

1.2.2 Hierarchical Summarization for Longform Spoken Dialog

Chapter 3 presents the first proof of concept longform audio browsing system, herein referred to as "System 1"[5] and proposes the foundational framework of using automatic summarization to decompose and presenting longform spoken audio in a hierarchical nature. Specifically, Chapter 3 discusses initial speech and language modeling challenges in developing the proof of concept audio browsing application.

1.2.3 Creating a Hierarchical Summarization Dataset for Longform Spoken Dialog

Chapter 4 discusses the development of a summary annotation tool and annotation protocol for the creation of a hierarchically annotated abstractive summarization dataset; this is required for the quantitative evaluation in Chapter 4. Chronologically, this data was obtained after work and development in Chapter 3 had finished, meaning longform hierarchical summaries were not yet available for use at the time.

1.2.4 Improving and Evaluating the User Browsing Experience in Summarization Systems

Chapter 5 covers an iterated and improved longform audio browsing system, herein referred to as "System 2", addressing and improving upon technical shortcomings in System 1 (Chapter 3) that affect usability. System 2 also introduces additional features to improve information cues

[5]System 1 exclusively refers to Chapter 3's instance of longform audio browsing system, while "system" refers to the overall concept of a longform audio browsing system.

and better assist user browsing. In particular, Chapter 5 emphasizes evaluation: quantitatively evaluating System 2's technical improvements on generated summary quality and qualitatively studying user interactions on browsing and skimming efficiency with System 2's.

1.2.5 Conclusion and Future Work

Chapter 6 summarizes this dissertation's contributions, discusses limitations of the longform audio browsing system, and suggests future work directions.

Chapter 2: Related Work

This section discusses the relevant subsets of Human Computer Interaction (HCI) and Natural Language Processing (NLP) fields this dissertation builds on top of through the lens of the spoken dialog (speech) domain. We begin by providing the necessary background theory on *information foraging* and related work describing how individuals browse for information. We then cover current works on media browsing systems and how automatic summarization is currently used in HCI applications.

On the NLP portion, we provide a brief overview on automatic speech recognition (ASR) systems and provide a discussion on their downstream effects towards language modeling. Next, we provide background on existing abstractive summarization architectures (with an emphasis on hierarchical and recursive abstractive summarization), summarization datasets, and summarization quality evaluation.

The vast majority of existing work in NLP summarization field is focused on developing and improving summarization language model quality and performance, neglecting research in downstream user applications and additional language model use cases beyond condensing text. However, by applying information foraging principles from HCI to longform speech, we gain valuable insights as to how automatic summarization can be leveraged to create a novel longform speech content browsing system.

It is important to note that our longform audio browsing system simply uses an abstractive summarization LM as its backbone in the overall system. Technical LM performance improvements (i.e. through new architectures or training regimens) are not the primary focus of this work and outside the scope of this dissertation, though improved LM performance directly translates to a better longform audio browsing experience to the user due to improved summarization quality.

2.1 Notation and Terminology

Throughout the technical portions of the dissertation we will symbolically refer to text at different levels (sentences, words, etc.) and use this section to set the standard notation and define terminology used throughout this dissertation.

- *Transcript*: Using ASR speech-to-text to transcribe an audio file (i.e. a .mp3 or .flac) outputs a text transcript that can then be then used as in input to downstream applications (such as summarization).

- *Gold Standard* and *Gold Label*: A reference to a summary that is human authored and assumed to be the perfect benchmark. Colloquially this can also be referred to as the *reference summary*. In the case of ASR transcripts, a gold standard or reference transcript is free of word transcription errors.

- *Language Model*: Though a language model (abbreviated LM or sometimes referred to as just "model" given sufficient context) typically refers to a probability distribution over words, we refer to them as exclusively large scale transformer architectures [8] that have been pre-trained on a large corpus and can be further used for additional NLP tasks. For the scope of this dissertation, we typically use language models in the context of *summarization*; for example a summarization language model is a LM that has been adapted for summarization use. Language modeling refers broadly to any sort of NLP task (i.e. machine translation, summarization, classification, entailment, etc.).

- Indexing: given example variable s, indexing will be usually expressed using subscript s_i to denote an element in a set. If another index is required, superscript s^k will be used. Superscripts s^k can also be used to signify membership to particular class or group (e.g. s^k belongs to class K and s^o belongs to class O). Variable n will be used to denote the last element in a set, such as $(s_1, s_2, ..., s_i, ..., s_n)$. Indexing will be used when specific references are needed, otherwise $s \in S$ is used.

- Word w is a singular token and is treated as the lowest level unit for the scope of text processing in this thesis. For simplicity, though words can be split into multiple tokens depending on their tokenization schema [9], we always assume one word is **exactly** one token.

- Sentence s is a sequence of words, where $w \in s$.

- Capital variables (e.g. S) refer to a list of sentences s, where $s \in S$. The grouped sentences are considered a structure. There are several ways S can be interpreted and referred to, all of which are a list of sentences:

  1. Source text: this is the set of input sentences that is passed into a summarization model. When referring to training and test data, all the inputs can be referred to as "source" and will usually be denoted with an S. In addition to source text, the input (to be summarized) can also be referred to as a "document" and input sequence.

  2. Target text: this is the set of text sentences generally on the output side. *Reference text*, text that usually consist of gold standard human authored summaries, is an example of a target sequence. This is colloquially referred to as the "targets" in text generation and will usually be denoted with a T. Output text (below) is also an instance of target text.

  3. Output text: also known as the *hypothesis*, this is the set of sentences outputted by a text generation model. In other words, this is the summarization model produced text summary, usually denoted with a H.

- Bold faced capital variables (e.g. $\mathbf{T}$ indicating a collection of a list of **segmented** (defined further down) sentences, such as a segmented audio transcript) indicate the the highest rank of items unless otherwise specified. This will refers to all segments S, where $S \in \mathbf{T}$.

- Ordinals and hierarchy: $w \in s \in S \in \mathbf{T}$ shows the complete progression of a word in a sentence in a segment in a transcript. Note that we can omit levels when referring to this hierarchy. For example, $w \in \mathbf{T}$ refers to all words in the transcript, $s \in \mathbf{T}$ refers to all sentence in the transcript, and $w \in S$ refers to all words in the segment (list of sentences).

- *Text Segmentation*: where a text is broken into semantically topical segments (or other information units), typically preserving and using existing sentence boundaries (.!?). The following example uses a toy illustration and to cover notation used in a combined setting.

  Consider a list of 5 sentences $S = (s_1, s_2, s_3, s_4, s_5)$, where each sentence has a word count of 30 words ($|w^{s_i}| = 30 \ \forall s_i \in S$) with the goal of segmenting S into segments, each containing no more than a total 65 words. Concretely stated, S now becomes $\mathbf{S} = \texttt{SEGMENT}(S)$[1] as S now becomes a collection of segments (each is a list of sentences) and $|w^{S_i}| \leq 65 \ \forall S_i \in \mathbf{S}$.

  1. Topic segmentation [10]. Sentences are typically grouped according to their semantic similarity and can be thought of as a clustering problem [11]; a segment can also be referred to as a cluster. If s_1, s_2 discussed dogs, s_3 discussed pet food, and s_4, s_5 discussed pet care, a resulting topic segmentation of these sentences would be: $S_1 = (s_1, s_2)$, $S_2 = (s_3)$, $S_3 = (s_4, s_5)$ and $\mathbf{S} = (S_1, S_2, S_3)$.

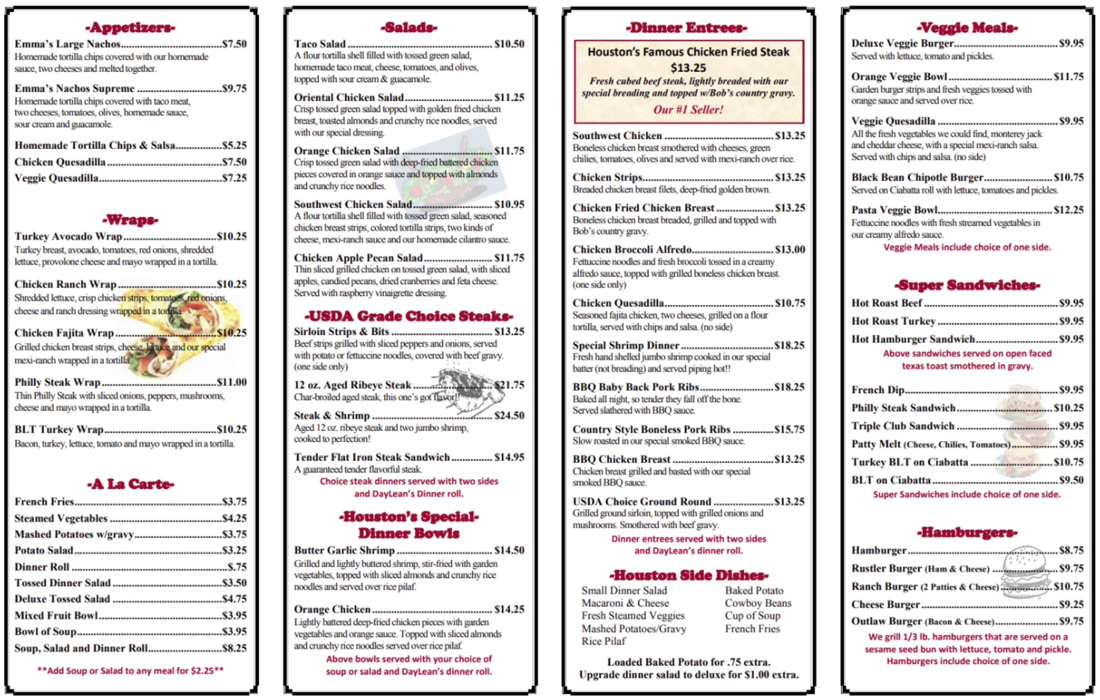
  2. Preserving sentence boundaries. When creating a segment of sentences, we do not split sentences apart (unless otherwise noted). If s_3 had also discussed dogs and is now semantically related to s_1, s_2 would result in $S_1 = (s_1, s_2, s_3)$ and would not a valid segment ($|w^{S_1}| = 90$). Thus s_3 can either (i) be added to S_1, removing either s_1 or s_2 and placing either in a different valid segment $S_i : i \neq 1$, preserving sentence boundaries or (ii) split between sentence boundaries and be grouped where the first 5 words are now contained in $S_1 = (w_1^{s_1}, ...w_{15}^{s_1}, w_1^{s_2}, ..., w_{15}^{s_2}, w_1^{s_3}, ..., w_5^{s_3})$, partially truncating s_3.

2.2 Human Computer Interaction

2.2.1 Information Foraging Background

Information foraging theory [4] is a concept from HCI that describes how users navigate and browse a large dataset to satisfy an information need. When browsing for information, people do

[1]Functions on variables in equations will be abstracted away when possible and referred to just as an operation. For example calling $\texttt{SEGMENT}(S)$ means segmenting the set of sentences S.



-Appetizers-

Emma’s Large Nachos $7.50
Homemade tortilla chips covered with our homemade sauce, two cheeses and melted together.

Emma’s Nachos Supreme $9.75
Homemade tortilla chips covered with taco meat, two cheeses, tomatoes, olives, homemade sauce, sour cream and guacamole.

Homemade Tortilla Chips & Salsa $5.25
Chicken Quesadilla $7.50
Veggie Quesadilla $7.25

-Wraps-

Turkey Avocado Wrap $10.25
Turkey breast, avocado, tomatoes, red onions, shredded lettuce, provolone cheese and mayo wrapped in a tortilla.

Chicken Ranch Wrap $10.25
Shredded lettuce, crisp chicken strips, tomatoes, red onions, cheese and ranch dressing wrapped in a tortilla.

Chicken Fajita Wrap $10.25
Grilled chicken breast strips, cheese, lettuce and our special mexi-ranch wrapped in a tortilla.

Philly Steak Wrap $11.00
Thin Philly Steak with sliced onions, peppers, mushrooms, cheese and mayo wrapped in a tortilla.

BLT Turkey Wrap $10.25
Bacon, turkey, lettuce, tomato and mayo wrapped in a tortilla.

-A La Carte-

French Fries $3.75
Steamed Vegetables $4.25
Mashed Potatoes w/gravy $3.75
Potato Salad $3.25
Dinner Roll .... $.75
Tossed Dinner Salad $3.50
Deluxe Tossed Salad $4.75
Mixed Fruit Bowl $3.95
Bowl of Soup $3.95
Soup, Salad and Dinner Roll $8.25

Add Soup or Salad to any meal for $2.25

-Salads-

Taco Salad $10.50
A flour tortilla shell filled with tossed green salad, homemade taco meat, cheese, tomatoes, and olives, topped with sour cream & guacamole.

Oriental Chicken Salad $11.25
Crisp tossed green salad topped with golden fried chicken breast, toasted almonds and crunchy rice noodles, served with our special dressing.

Orange Chicken Salad $11.75
Crisp tossed green salad with deep-fried battered chicken pieces covered in orange sauce and topped with almonds and crunchy rice noodles.

Southwest Chicken Salad $10.95
A flour tortilla shell filled with tossed green salad, seasoned chicken breast strips, colored tortilla strips, two kinds of cheese, mexi-ranch sauce and our homemade cilantro sauce.

Chicken Apple Pecan Salad $11.75
Thin sliced grilled chicken on tossed green salad, with sliced apples, candied pecans, dried cranberries and feta cheese. Served with raspberry vinaigrette dressing.

-USDA Grade Choice Steaks-

Sirloin Strips & Bits $13.25
Beef strips grilled with sliced peppers and onions, served with potato or fettuccine noodles, covered with beef gravy. (one side only)

12 oz. Aged Ribeye Steak $21.75
Char-broiled aged steak, this one’s got flavor!!

Steak & Shrimp $24.50
Aged 12 oz. ribeye steak and two jumbo shrimp, cooked to perfection!

Tender Flat Iron Steak Sandwich $14.95
A guaranteed tender flavorful steak.

Choice steak dinners served with two sides and DayLean’s Dinner roll.

-Houston’s Special-
Dinner Bowls

Butter Garlic Shrimp $14.50
Grilled and lightly buttered shrimp, stir-fried with garden vegetables, topped with sliced almonds and crunchy rice noodles and served over rice pilaf.

Orange Chicken $14.25
Lightly battered deep-fried chicken pieces with garden vegetables and orange sauce. Topped with sliced almonds and crunchy rice noodles served over rice pilaf.

Above bowls served with your choice of soup or salad and DayLean’s dinner roll.

-Dinner Entrees-

Houston’s Famous Chicken Fried Steak
$13.25
Fresh cubed beef steak, lightly breaded with our special breading and topped w/Bob’s country gravy.
Our #1 Seller!

Southwest Chicken $13.25
Boneless chicken breast smothered with cheeses, green chilies, tomatoes, olives and served with mexi-ranch over rice.

Chicken Strips $13.25
Breaded chicken breast filets, deep-fried golden brown.

Chicken Fried Chicken Breast $13.25
Boneless chicken breast breaded, grilled and topped with Bob’s country gravy.

Chicken Broccoli Alfredo $13.00
Fettuccine noodles and fresh broccoli tossed in a creamy alfredo sauce, topped with grilled boneless chicken breast. (one side only)

Chicken Quesadilla $10.75
Seasoned fajita chicken, two cheeses, grilled on a flour tortilla, served with chips and salsa. (no side)

Special Shrimp Dinner $18.25
Fresh hand shelled jumbo shrimp cooked in our special batter (not breading) and served piping hot!!

BBQ Baby Back Pork Ribs $18.25
Baked all night, so tender they fall off the bone. Served slathered with BBQ sauce.

Country Style Boneless Pork Ribs $15.75
Slow roasted in our special smoked BBQ sauce.

BBQ Chicken Breast $13.25
Chicken breast grilled and basted with our special smoked BBQ sauce.

USDA Choice Ground Round $13.25
Grilled ground sirloin, topped with grilled onions and mushrooms. Smothered with beef gravy.

Dinner entrees served with two sides and DayLean’s dinner roll.

-Houston Side Dishes-

Small Dinner Salad	Baked Potato
Macaroni & Cheese	Cowboy Beans
Fresh Steamed Veggies	Cup of Soup
Mashed Potatoes/Gravy	French Fries
Rice Pilaf	

Loaded Baked Potato for .75 extra.
Upgrade dinner salad to deluxe for $1.00 extra.

-Veggie Meals-

Deluxe Veggie Burger $9.95
Served with lettuce, tomato and pickles.

Orange Veggie Bowl $11.75
Garden burger strips and fresh veggies tossed with orange sauce and served over rice.

Veggie Quesadilla $9.95
All the fresh vegetables we could find, monterey jack and cheddar cheese, with a special mexi-ranch salsa. Served with chips and salsa. (no side)

Black Bean Chipotle Burger $10.75
Served on Ciabatta roll with lettuce, tomatoes and pickles.

Pasta Veggie Bowl $12.25
Fettuccine noodles with fresh streamed vegetables in our creamy alfredo sauce.

Veggie Meals include choice of one side.

-Super Sandwiches-

Hot Roast Beef $9.95
Hot Roast Turkey $9.95
Hot Hamburger Sandwich $9.95

Above sandwiches served on open faced texas toast smothered in gravy.

French Dip $9.95
Philly Steak Sandwich $10.25
Triple Club Sandwich $9.95
Patty Melt (Cheese, Chilies, Tomatoes) $9.95
Turkey BLT on Ciabatta $10.75
BLT on Ciabatta $9.50

Super Sandwiches include choice of one side.

-Hamburgers-

Hamburger $8.75
Rustler Burger (Ham & Cheese) $9.75
Ranch Burger (2 Patties & Cheese) $10.75
Cheese Burger $9.25
Outlaw Burger (Bacon & Cheese) $9.75

We grill 1/3 lb. hamburgers that are served on a sesame seed bun with lettuce, tomato and pickle. Hamburgers include choice of one side.

Figure 2.1: **The typical restaurant menu best demonstrates information foraging and information browsing**. Menus are a good example of how information can be optimized for browsing. When a patron is at a restaurant, they typically do not already know what to order, but have a rough sense of what they like. This puts the patron in a browsing mindset, where information cues and structure such as subheadings are immensely helpful. They can quickly indicate what the different categories of menu items are, without requiring the reader to sift through unnecessary and irrelevant details.

not search linearly. Instead, people "forage" for information, similar to how animals sniff around for food, scanning from area to area (where each area is a possible source of information in the aforementioned hypothetical dataset for this analogy). Clearly, it does not make sense for an animal to stay and explore an area without an obvious indication that it contains food; the animal should move on to greener pastures.

Similarly, when browsing, people mentally consider the trade-off between how much information can be gained from a particular source of information versus the cost (i.e. time spent) of exploring and extracting information from said source. When considering the time cost of browsing [12], they generally fall into two categories:

1. Between-patch: where the user spends time determining what information sources are rel-

evant. For example, when researching a subject you may compile a list of all the websites that may be relevant.

2. Within-patch[2]: where the user spends time investigating each of the patches of information to extract information from them.

People tend to investigate information sources (areas) that minimize both of these time costs and maximize the information gain as a trade off. In other words, people try to obtain as much relevant information in as little time as possible.

An example that demonstrates users in an information foraging context is when one visits a restaurant and is handed a menu (Fig 2.1). Typically, the customer does not know precisely what to order, but has a general idea of what kind of food they like. To explore the restaurant's offerings, customers scan (forage) areas and portions of the menu. When the customer finds a particular section that plausibly contains an item that satisfies what they want (information need), they can dive deeper to find a specific menu item to order.

To help the reader navigate, the menu has large header sections that signify different item categories, followed by subheading titles and smaller details of each item. Each of these section headings provides clear indicators to a customer on the finer details of the menu's content. Based on these information cues, a customer is able to estimate how relevant the food items in a particular section is towards what they may want to order. This concept is known as *information scent*, [13] which describes what people use to *imperfectly* estimate how much relevant information is contained on their current exploration path, and how to adjust and reorient themselves as necessary to retrieve relevant information.

There are two aspects of information scent that are especially important to understanding user browsing behavior:

1. SNIF-ACT [5] keenly observes how information scent is incorporated into a person's thought process. Users would leave their current exploration path if the information scent diminishes

[2]For the scope of this dissertation, our speech browsing and navigation system primarily focuses on the second category of within-patches where the focus is to help users browse and navigate audio within a particular information source (such as a podcast or interview) as opposed to finding different podcasts and interviews to listen to.

below a certain threshold. This generally categorizes how people browse for information, as individuals browse a source for information and they realize that the information is *now no longer relevant* to their browsing interests, they typically proceed onto the next source or patch of information.

2. Next, Pirolli [13] theorizes that if information scent cues are perfect then the user will make no navigation errors and will proceed directly to the desired information. In other words, when browsing, if each navigational decision is made on perfect information, that is to say each information cue or indicator perfectly aligns with the individual's interpretation of what the subsequent information is and provides, the user will have no difficulty in finding the information they desire.

In most cases, people do not know in advance how much information a particular source of information has and how much time it would take to extract said information. However, people are able to draw upon any information cues the current source of information emits to make their browsing decision. How people make sense of the information cues that are present is known as the concept of *sensemaking*. Sensemaking [14] describes the process of searching and forming useful representations from data. When users give meaning and rationalize information, they draw upon their own collective experiences. As a result, the final understanding an individual arrives at may vary from a different individual's and is relative. It is also important to point out that the concept of information scent is relative to an information need [15]; the same source of information can give off different information scents for different information needs. For example, if a customer is in the mood for sweets and desserts, then the main course section would have a non-existent information scent to this particular customer, as it is very likely that sweets and desserts are not served as main courses. Conversely, the main course section will emit a large information scent to a customer looking to order a filling meal. It follows that different users have different user needs, demonstrating how information scent is also relative to users.

2.2.2 Information Foraging Implications Towards Longform Spoken Dialog

Principles in information foraging describing how users browse information are universal and provide valuable insights that can be applied towards designing a system to help users navigate and browse speech.

Information Patches and Structure. First and foremost, speech content should be structured in a manner that is conducive towards browsing, motivating the system's hierarchical presentation of information and segmenting content into topically distinct segments, or "information patches", which are then presented in a linear sequential fashion. By breaking up and structuring audio and longform dialog transcripts into manageable information patches, users are able to individually assess, patch by patch, what is relevant to their current information need and whether or not to adjust their navigational path.

The system's hierarchical presentation allows users to investigate progressively in more and more detail, giving the user multiple stages to opt out of their current navigational path and to stop browsing when their information need is satisfied. This can be interpreted as an optimization for reducing user's within-patch information extraction time. Encouragingly, foundational work in information foraging [16] supports the notion that structuring information (from a document) into hierarchical clusters is an effective user interface representation that is conducive towards user browsing and navigating. In a different study discussing the effects off text structure on navigation performance [17], researchers found that subjects tend to perform better navigating text that is presented in a linear manner (as opposed to non-linear) whereas a hierarchically structured text experiment fell in between the two. This study emphasizes the importance of maintaining an aspect of linearity to when presenting browsing content.

Information Scent. Next, the system needs to provide a reliable and accurate way for users to estimate an information source's content (information scent), articulating the requirement for information to be represented in a manner that is both concise and indicative of its actual information content [12]. Now, it becomes apparent how automatic summarization can be leveraged.

By extracting salient and representative information of different portions of speech transcripts, the summaries now fit the additional function of being representative indicators of a portions of a speech transcript's underlying content. In other words, *summaries can be used as waypoints to help users navigate longform audio*. This can be interpreted as an optimization for a user's between-patch browsing time. It is important to also observe the opposite situation; when summaries are used as a navigational vehicle and provide no information cues (i.e. the summary is unintelligible and/or confusing), it will cause the user's browsing strategy to devolve into random choices; highlighting the importance of coherent and accurate summary text generation.

2.3 Applications (HCI)

2.3.1 Systems for Navigating and Browsing Media Content

Various tools already exist to assist users with navigating and browsing different forms of media, primarily concerned with videos and the audiovisual domain. In this section we break down prior research in audio (text transcripts) and visual media (video) domains.

Audio and Text Navigation. To the best of our knowledge systems designed to navigate and browse strictly spoken dialog (and text transcripts) have primarily been explored in older work (pre-2000s). Many of the language processing techniques common today were not yet available (primarily due to advances in transformer architectures [8]) which limited automatic processing and modeling capabilities.

SpeechSkimmer [18] is an early tool that lays the foundation in addressing the challenges and difficulties of navigating and browsing speech, expressing how *"there is no natural way for humans to skim speech information because of the transient nature of audio—the ear cannot skim in the temporal domain the way the eyes can browse in the spatial domain"*[3]. SpeechSkimmer developed a system that enables users to quickly skim through an audio file **by listening** to compressed segments and choose to continue to listen to compressed segments in more detail, demonstrating the effectiveness of hierarchically structuring audio. Notably, the system is paired with a hard-

[3]Taken from Section 1. Introduction of SpeechSkimmer: A System for Interactively Skimming Recorded Speech [18].

ware touchpad with `begin, skim, no pause, normal` controls to allow users to listen and navigate audio in forward and backwards temporal directions. To initially find skimmable sound bites, SpeechSkimmer leverages natural dialog structures such speakers pauses and pitches as a basis for segmentation and compresses the audio in a manner that retains intelligibility and voice quality. It is important to note that SpeechSkimmer operates entirely on the audio domain, without any text, and is a system that assists with *within-patch* navigation.

Another example of an early work focused on helping users navigate speech is SCAN [19], a prototype speech retrieval and browsing system that aims to help users navigate poor automatic transcriptions and retrieve multiple speech transcripts. Unlike SpeechSkimmer, SCAN operates on the text level with automatic speech transcription and assists with *between-patch* navigation. Critically, there is a difference between systems such as SpeechSkimmer and SCAN where the former is focused on enriching *within-patch* navigation and the latter addresses *between-patch* navigation. Other *between-patch* systems (i.e. a recommender system) help a user browse and decide between multiple audio files (podcasts) to listen to [20, 21, 22] based on their preferences. The idea of *between-patch* browsing and exploration is also applied to other domains, such as news [23]. In News Rover [23], the system structures and links multiple news sources in a unified user interface.

Navigation systems [24, 25] also attempt to visually represent audio content and investigate another angle of helping users navigate audio, particularly meetings, by presenting concepts discussed as timelines and concept maps that encapsulate the the underlying speech content. By identifying salient elements of the meeting content and presenting them through an augmented storyline visualization, MeetingVis [24] helps users better recall and reflect on portions of the meeting quickly. This approach is also extended beyond communicative and expository audio; MusCat [26] is a system that helps users browse through music (non-informative content) audio through abstract visual image representations. The system reduces music files into features and hierarchically clustering them to generate abstract pictures, MusCat helps users associate music features to images, enabling users to browse audio visually.

Visual Media Browsing and Navigation. Videos can be defined as a "document" containing a succession of images over time, usually with additional modalities such as audio and text. For similar reasons to audio, browsing and navigating video is also challenging for users. However, unlike audio, visually browsing content is possible (ranging from browsing videos such as single and multiple thumbnails, playable video segments, to video collages [27]), prompting a different set of strategies with respect to how a user can navigate and browse audio. Accordingly, a host of different systems and methodologies have been developed to easily navigate through videos and movies by navigating to the video clip and allowing users to interpret content [28, 29, 30].

Video summarization [31] [32] as means for skimming and browsing for information [33] is a popular domain and active area of research. Early work [34] on video summarization automatically generated concise video summaries as a way for fast skimming and browsing. CueVideo [35] proposes a system where sequences with low motion (with the assumption of low information content) are played with fewer frames, allowing the viewer to gloss over scenes with less information.

SceneSkim [36] is an example of sensemaking that made lengthy multimodal data, video from movies and text from movie scripts, indexable, enabling users to efficiently search movies for specific segments. In the user interface, SceneSkim presented multiple modes of search, over captions, summaries, movie script, and movie clip, synchronized by timestamp. VidSceneDetect [37] similarly identified scenes within video and longform multimodal content by creating a hierarchical representation that covered various degrees of granularity for a recursive approach to video summarization. More recent work has also adapted the use of hierarchical information to provide users with multiple levels of summarization and information [38] for instructional videos.

Other work on video browsing and navigation investigate methods to enrich existing user browsing behaviors; Swift [39] and Swifter [40] tackle the challenge of real-time seeking in video scrubbing (where a user drops and drags a the playhead on a video timeline). When a user moves the playhead on a video timeline player, Swifter presents surrounding local video frames in a tiled manner, giving the user an idea of what video content is locally contained at this particular timeline location. Relating back to the concept of information cues, this provides the user with more information cues to more accurately infer what the content is in the underlying video content at this particular time. The similarities of successful content navigation systems are evident; providing users sufficient information cues that fit user browsing habits is paramount.

2.3.2 Summarization in User Applications

Summarization is employed in a variety of user applications across different domains, such as summarizing doctor notes [41]. In addition to the principal function of summarization to condense information, these applications have found additional innovative uses applying summarization to collaborative editing:

1. Wikum+ [42] is a system that presents groups with a way to interleave and summarize discussion on an online forum into summaries. A key contribution of their system was the interleave and summarize method in a recursive manner, creating a summary tree, which allowed ideas to flow back and forth into an iteratively refined summary between multiple editors. This work demonstrates the effectiveness of refining and summarizing information in a hierarchical and recursive nature, albeit with human editors. Moreover, by presenting the hierarchical summarization structure in a user interface, experiments demonstrate Wikum is able to help users better identify and explore main topics.

2. Arkose [43] presents a different approach towards incremental summarization to distill information from online discussion boards. The system proposes a method of collaborative summarization that involved merging at multiple hierarchical levels under the name of "incremental diagenesis" as a way to structure and process unorganized online large-scale community discussions. By deconstructing summarization into a (recursive) incremental process, writers are able to improve the quality of final summaries. Interestingly, this mirrors (and predates) automatic procedures in later automatic summarization models [44] that find content selection in summarization is better for content selection[4].

[4]Given an input text, the system (summarization language model) must determine what information should be retained in the final output [45]

Other systems join automatic summarization language models with multi-modal content [46], applying them to new domains such as summarizing UIs [47], medical videos [48], and generating pictures, textual summaries and structure of complex text [49].

2.4 Automatic Speech Recognition (ASR)

2.4.1 Speech-to-Text

Automatic Speech Recognition systems (ASR) [50, 51] are used to transcribe audio (using word recognition) into a source language transcript [52]. Concretely, in speech-to-text ASR systems, an audio file, such as a .mp3 file, is converted a text transcript. Such systems have recently made relatively significant strides in terms of practical performance. and have seen recent widespread adoption in various practical language applications.

State of the art (SOTA) ASR [53] is no longer constrained by vocabulary and remains relatively robust, encouragingly extending word recognition to topical domains and noisy audio. SOTA systems [53] also offer a wide suite of useful services such as automatic punctuation insertion [54, 55, 56], where raw transcribed text has punctuation such as capitalization and sentence endpointers (periods, exclamation marks, and question marks) are automatically inserted using another language model. Advances in audio source separation can be used to identify speakers (speaker diarization) [57] and enhance difficult-to-hear dialog while suppressing background noise. Cleaner transcriptions enable better downstream processing opportunities, such as dialog summarization, spoken machine translation, and dialog classification systems. Figure 2.2 demonstrates how ASR can convert speech into transcripts and the subsequent multiplicity of use cases of downstream use.

Note that speaker diarization classifies all words w with a speaker label in a given transcript S. This can induce a segmentation and create *speaker turns* that are chunks of sentences where the breaks that create segments are the breaks from alternating speaker dialog turns. Formally this gives $\mathbf{S} = (S_1, S_2, S_3, ..., S_n)$. Many systems and works [18, 19, 58] use the concept of speaker turns as part of their speech processing.

ASR systems have already re-invented how we interact with audio. SCANMail [59], an early work, lays such foundations of user interactions with ASR. Combining an intersection of HCI, ASR and information retrieval (IR), SCANMail allows users to browse and search voicemail messages according to content. Such parallels can be seen today in everyday Google Home and Alexa speaker interactions of voice information retrieval [60]. State-of-the-art speech recognition models have also been deployed to automatically caption subtitles for videos [61] to improve accessibility for deaf users and have demonstrated that automatic speech recognition can be applied towards use cases in various settings such as classrooms [62], home environments, and meetings. The system [62] utilizes ASR techniques in classrooms to automatically analyze classroom discourse and collect feedback for teachers.

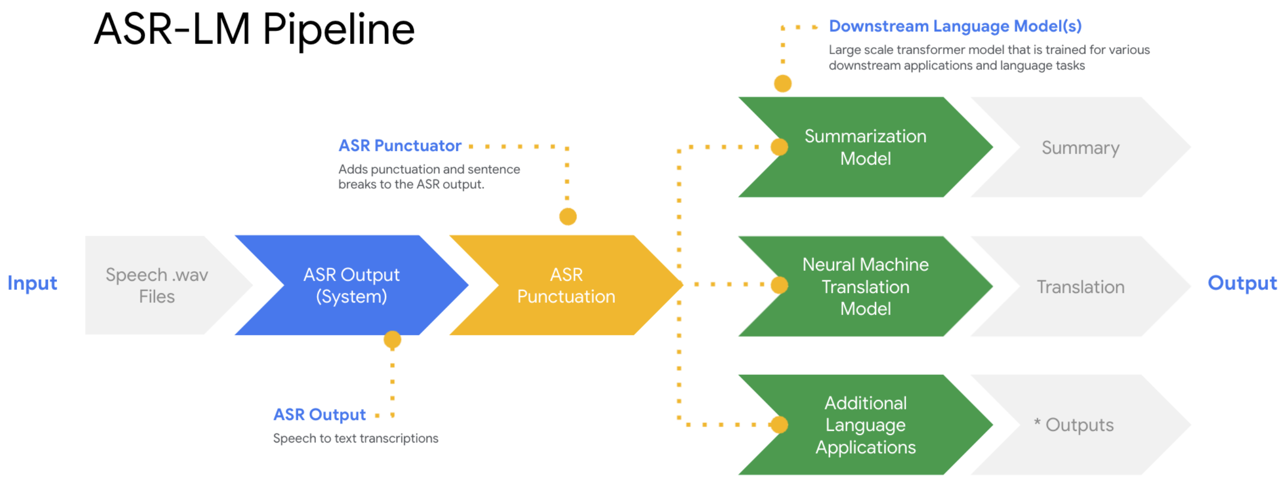

ASR-LM Pipeline
Downstream Language Model(s)
Large scale transformer model that is trained for various downstream applications and language tasks
ASR Punctuator
Adds punctuation and sentence breaks to the ASR output.
Input
Speech .wav Files
ASR Output (System)
ASR Punctuation
Summarization Model
Summary
Neural Machine Translation Model
Translation
Additional Language Applications
* Outputs
Output
ASR Output
Speech to text transcriptions

Figure 2.2: **Automatic Speech Recognition and Language Model Framework**. ASR systems are typically used as a starting point from where audio is transcribed into text for further applications. This figure demonstrates the possibilities of downstream language modeling.

For the scope of this thesis, we make the distinction that the vast field of ASR is not the focus of our research[5]. We treat ASR systems as a separate black box and the starting point for our work and subsequently employ summarization language models for our applications.

[5]All references to ASR in this thesis are towards Google's Speech-to-Text service which is widely considered state of the art.

2.4.2 ASR and Speech Challenges towards Language Modeling

While transcribing audio, ASR systems may introduce problematic errors, *word recognition* and *context fragmentation*, that propagate and affect downstream language modeling (Fig 2.2 blue and yellow segments).

1. **Word Recognition.** The first immediate challenge for ASR is word recognition; transcribing words that are phonetically similar by different meanings remains a challenging task for ASR (e.g. "`weather`" vs "`whether`"). This problem has been explored extensively in ASR models [63, 64] and typically addressed through domain and context specificity [65]. Though rarely an issue, we find that word recognition errors are occasionally present in audio files where speakers with accents or large amounts of background noise are present. We defer correctness to Google Speech-to-Text API. It is evidently apparent how an incorrectly transcribed word can ruin a downstream language model's text comprehension and lead to poor performance. Consider the perfectly transcribed sentence "`There will be day of reckoning`" compared to one containing a word transcription error "`There will be a loud rectangle opening`". It is inconceivable that a downstream language model (such as Alexa) will be able to sufficiently recover the correct semantic meaning.

2. **Segmentation and Context Fragmentation.** Generally, longform input text has to be appropriately split (segmented) because of the limited input size of language models and used as separate independent inputs during encoding. This is typically referred to as the task of *text segmentation* where a text is broken into semantically topical segments (or other information units), typically preserving and using existing sentence boundaries (.!?) [66].

   Unfortunately, segmentation leads to problems such as context fragmentation where the input has information that is predicated on context that is no longer part of the input, leaving inadequate information for a language model to process. Ensuring text inputs that are not only free of context fragmentation but also semantically cohesive is instrumental towards downstream LM performance.

When a large input text requires segmentation, two dimensions of the underlying text must be considered: fine syntactic segmentation where words in a complete sentence are grouped together and not divided, and coarse semantic segmentation where semantically similar sentences are grouped together. While this can usually be done by segmenting along pre-existing syntactic boundaries (periods, commas, question marks) for written text, ASR transcripts obtained through word recognition speech models either do not contain punctuation or have punctuation that is predicted by separate models and is often imperfect, creating sentence fragments and incorrectly compounding sentences.

Furthermore, due to the stylistic nature of spoken language and intrinsic lack of structure, unlike that of a written article, topics are not presented in a cohesively segmented manner:

(a) There are no topic sentences to rely on.

(b) Speakers can stop mid sentence and backtrack their thought or never complete it.

(c) Maintaining coherency is challenging when multiple speakers are making different points simultaneously.

These speech domain unique attributes of spoken language make proper segmentation into topical chunks particularly difficult and is a wide field of active research. We can easily observe how poor syntactic segmentation's impact on model inference is immediately apparent - context in a sentence could be missing because of an incorrect split or incorrect information could be added due to an erroneous sentence join. Poor topic semantic segmentation behaves similarly, but on a larger scale. Particularly, joining sentences containing distinctly different information can confuse a model and output a nonsensical and grammatically incorrect summary. Conversely, splitting sentences too aggressively would defeat the purpose of summarization. Accordingly, finding a careful balance on semantic grouping presents itself as a unique challenge in speech segmentation.

These errors can be addressed at the ASR system level or at subsequent downstream processing stages (done in our system). Some examples of downstream system [2] post processing of ASR include: running a parts of speech tagger (POS) to identify and remove disfluencies, repetition removal, and sentence boundary (.!?) correction of improperly joined sentences. In the following enumerated examples, we go over some technical challenges posed by speech data and their ramifications *specifically* towards speech summarization.

1. *Lack of Spoken Language Summarization Training Data.* The most apparent (and generally machine learning) concern is the general lack of annotated training data for **longform** speech and dialogue summarization (see Section 2.7 for existing datasets), leading to the issues of domain adaptation and model robustness towards spoken language specific noises. In domain adaptation, existing pretrained models are dataset oriented, trained on domains such as news [67] or social media [68]. As a result, these models *incur specific memorized behaviors* [69]. For example, a news based model may insert a preamble of "`According to the Huffington Post`" before an output, regardless of the subsequent text's origins. In model robustness, speech and conversation contain spoken user disfluencies such as "`like you know like the actual underlying prices`" that do not convey meaningful information. Without speech and dialogue specific annotations, standard training data does not contain sufficient diversity of training pairs leading to summarization models struggling to filter out unimportant phrases.

2. *Propagated ASR Errors.* As noted previously, ASR errors such as improper segmentation and word recognition errors are propagated downstream and given as the language model's input. The impact of these errors are immediately apparent as the lack of context (or addition of unnecessary / confusing context) from poor segmentation and erroneous transcribed words would lead to a decrease in semantic understanding for the language model, thereby producing poor summarizations. More specifically, abstract summarization already requires a high degree of semantic comprehension [70] of the underlying transcript meaning in order to generate a cohesive summary; multiple contrasting topics introduced as a single input could easily confuse the model. Due to the tendency of run-on sentences in speech and the

limited input size of language models, accurate coarse structuring of topics and fine segmentation of sentences are paramount towards proper conceptual LM understanding.

2.5 Text Summarization Overview and Challenges

Language Modeling and Text Generation. With recent deep learning advances, most notably in the form of powerful transformer architectures [8], language models now have far greater contextual modeling and understanding capabilities. Using the attention mechanism design as a base, subsequent modifications towards modifying transformer complexity [71, 72] and reframing of novel semi-supervised pre-training objectives [73, 74], broadly benefit many language tasks ranging from text generation (content understanding) tasks such as abstractive summarization [75] and Machine Translation (MT) to classification tasks like Named Entity Recognition (NER) [76] and coreference resolution [77].

By reducing the complexity of the transformer's attention mechanism from quadratic to pseudo-linear [78], newer models are able to handle larger text input sizes. Through providing a larger direct input size, language models are able to process a greater amount of contextual information and potential semantic dependencies, improving semantic understanding for any particular language task. This allows for such architectures to be extended towards other domains such as conversations (spoken language) that traditionally could not be processed due to text length constraints. Furthermore, newer word tokenization schemas [9] are specifically optimized for neural text processing and allow for specific data instance based training of subword segmentation, allowing for more flexible and independent end-to-end language modeling.

Text generation is the task of automatically producing text subject to certain contexts and constraints [79, 80], that typically consists of an encoder and decoder language model [81] where, given an input passage or source text $S = (s_1, ..., s_n)$, the objective is to output a sequence of tokens (the hypothesis $H = (s_1, ..., s_n)$) that fit the task. In the example of summarization, the output sequence of tokens would be a summary that best compresses information from S. The encoder first encodes the passage into a fixed hidden representation [81] containing some representation of the input passage; intuitively, one can consider this vector representation as containing the most salient information in the passage. Afterwards, a decoder language model interprets the hidden representation into a sequence of words that best summarizes the input [75]. This procedure broadly generalizes to all text (token) generation tasks, such as neural machine translation [82], controllable generation [83], summarization [75], and even music composition [84].

Extractive and Abstractive Summarization. In text summarization, a subset of the text generation natural language task, the goal is to generate a concise and accurate summary of a larger input text and attend towards key sections. Text summarization techniques can be classified into two categories: *abstractive* [75] and *extractive* [85]. Abstractive summarization generates (token by token using the previously described seq2seq text generation framework) a new unique summary of text given a context whereas the extractive summarization "quotes", selecting relevant portions of the input text (usually formulated as a binary classification problem [86]), and concatenates relevant portions to compose into a summary. Because of spoken language noise effects in ASR transcripts, extracting transcript segments verbatim often leads to poor summaries. Therefore, we opt for abstractive summarization in our system. However, abstractive summaries are more difficult to generate but more closely mirror human written summaries as they may contain key words and phrases that may are not present in the original text but better describe the source text [87]. Conversely, for abstractive summarization language models [88], as with most large scale LMs on text generation tasks [89], learning to output syntactically correct and fluent summaries is not challenging and is less of a concern.

Challenges and Considerations in Abstractive Summarization. Purely generating a summary requires a high degree of semantic understanding, as opposed to the "cut and paste" approach of extractive summarization, to produce a cohesive and fluent summary. When a language model is unable to properly understand a given input, it is prone to *hallucinations* [69, 90], phrases or entities that appear to be semi relevant but are not actually present in the underlying text. Typically issues in model comprehension are attributed to insufficient context, training data challenges of domain shift, or model complexity limitations.

Another important consideration in abstractive summarization is the factual accuracy of a generated summary with respect to the input text. Differently stated, is the summary's meaning *consistent*[6] and *faithful* with the original source text? The possible ramifications of giving a user inaccurate summaries can lead to arbitrarily detrimental results. Ensuring faithful and accurate is vital to real-world applications of abstractive summarization.

Several different approaches exist to tackle the issue of faithfulness and can be broadly categorized as follows:

1. Improving models to become more robust and generate more consistent summaries at the model level through architectural and training regimen modifications [91, 92, 69, 93, 94].

2. Evaluating generated summaries for factual inconsistencies with external language models (such as Question Answering that automatically ask questions about a summary and determining if the answers are consistent with the source) [94, 95, 96, 97]

3. Correcting and rewriting factual errors in generated summaries [98, 99].

Of course, while having factually consistent summaries is paramount towards our longform dialog browsing system, this research area is not a primary technical focus of this dissertation.

2.6 Summarization Models and Summarization Methodologies

This section, with an emphasis on speech and dialog, gives a brief background on abstractive summarization models and current challenges and strategies of summarizing longform text. While there are many summarization models and work [100, 101, 102] that address speech induced specific technical challenges [103, 104, 105, 106] (Chapter 2.4.2) as well as new and interesting modeling approaches (i.e. an iterative processing model that allow for longform podcast summarization [107]), we specifically focus on the input size challenge that longform speech poses.

[6]Faithful, consistent, and factually accurate are dimensions to describe how well the text generated summary's content compares to the source text. See Section 2.8 for a complete list of definitions for summarization evaluation dimensions.

Popular Abstractive Summarization Models. In the most popular[7] abstractive summarization models, the language model's core architecture remains unchanged from the original transformer block [8]. Instead, work has largely centered on modifying on modifying the language model's objective function (training criteria).

1. BART [108] is a denoising autoencoder that modifies the pretraining of a standard transformer based neural machine translation (text generation) architecture. Additional operations such as token masking, token deletion, text infilling, sentence permutation, and document rotation extend the existing word masking and next sentence prediction objectives in BERT [73]. Specifically, for text generation (summarization) tasks, the BART instance comprises of an encoder and autoregressive decoder.

2. PEGASUS [109] is another popular summarization model. By innovatively changing the pre-training process from standard word level masked language modeling (where models learn language conventions and syntax by predicting individually removed words within sentences), to sentence level masked language modeling (where entire sentences are removed and then recovered), the language model is tasked with a substantially more challenging task. Not only does this new pre-training objective encourage the model to learn more generalized knowledge (through posing a difficult inference problem), it also closely mirrors summarization as a downstream task.

2.6.1 Hierarchical and Recursive Summarization

Encoding longform text for summarization (typically referred to as longform summarization) poses technical obstacles due to language model size and memory constraints; sometimes text inputs are still too lengthy to be processed in one forward pass of an encoder. Length constraints occur when the input size of a summarization model exceeds the typical allotted 1024 tokens, though some language models may have larger maximum inputs. This is particularly problematic

[7] A quick way of determining this is by looking at HuggingFace's most downloaded summarization language models.

in the spoken dialog domain where meetings, conversations, and interviews can span hours, for reference a 20 minute speech audio file may contain anywhere from 4,000 to 6,000 words[8], far exceeding a typical transformer based language model token input limit. The inability to jointly encode text is problematic as important information and context that is critical to producing a summary may not be available to the LM during decoding and is similar to the segmentation context fragmentation challenges discussed in Chapter 2.4.2.

As a result, two prominent strategies[9] for processing extremely long form input sequences such as multiple documents [111] or long conversation transcripts are used: modifying input sequences to become more tractable with existing LMs (i.e. reducing the input length) or designing new hierarchical deep learning architectures to accommodate greater context.

Text Level Hierarchical and Recursive Summarization. In the first approach, techniques are concerned with breaking up the input source text $S = (s_1, ..., s_n)$ into a more manageable form. At a high level, this typically is done by breaking S into smaller segments: $\mathbf{S} = [(s_1, ..., s_i), ..., (s_j, ..., s_k), ..., (s_l, ..., s_n)]$ where $1 < i < j < k < l < n$. Observe how S now becomes $\mathbf{S}$ as it is a collection of segments of sentences, such as $S_i = (s_j, ..., s_k)$, which is now an individual input to a summarization model. For each $S_i \in \mathbf{S}$ the summarization model generates an output H_i (meaning now there are i summaries). The outputs $H_i \in \mathbf{H}$ are all then concatenated to form the final concatenated summary H^c. The superscript ^{c} indicates this is the concatenation of all individual $H_i \in \mathbf{H}$[10]. From here, the process is typically repeated where the output H^c now recursively becomes the new input source text.

To train a language model to that can handle and summarize a longform document that is broken up segments, Divide-ANd-ConquER (DANCER) [112] introduces (as its name suggests), a divide and conquer strategy to generate training data examples from any existing dataset. Given a dataset containing source text and summary pairs $(S_i, T_i) \in (\mathbf{S}, \mathbf{T})$, every *single* (S_i, T_i) training pair is broken into *multiple* shorter training pairs. This is done by iteratively taking a sentence

[8]Accounting for SPM (sentence piece word tokenization) and subwords there is likely a 1.3 multiplier on words to number of tokens. For example, `running` can be broken down into `run` + `ning` depending on training.

[9]We do not go into detail on retrieve-then-summarize pipeline models [110].

[10]Another way to imagine this is flattening the list of lists $\mathbf{H}$ by one level.

$s^{T_i} \in T_i$ in the summary and using ROUGE [7] as a metric, match it to all similar sentences $s^{S_i} \in S_i$ in the source text. Intuitively, the matching attempts to find all the information (sentences $s^{S_i} \in S_i$) that may be relevant to generate the individual sentence $s^{T_i} \in T_i$ in the summary. Now, there are far more training pairs, with each new target summary being substantially shorter than the original dataset's summary. $SUMM^N$ [113] expands upon the Divide-ANd-ConquER process by repeating the process recursively generate target summaries. Here a generated output summary $H^{c,j}$, which has already been concatenated from all the individual components, at hierarchical level j (containing a total of J recursive levels), would become the input S^{j+1} to another summarization model[11]. This is repeatedly done until a short summary of a longform document remains. Note that this creates J separately trained summarization language models. By processing longform text is this manner, $SUMM^N$ improves upon context fragmentation and can handle even longer inputs.

OpenAI [114] successfully applied recursive task decomposition to abstractive summarization. The authors first observe that large pretrained LMs do not produce good quality summaries and train a reward model to learn what summaries are preferred by humans. From here, the authors fine tune an instance of GPT-3 [115] according to the previous reward model. This process is repeated recursively until a short summary is achieved. Impressively, this model is able to ingest and summarize entire books with the aid of intermittent human summarization annotations and feedback.

In our work [116], we similarly hierarchical processing approach to break down longform spoken dialog into distinct and concise summaries. However, we repurpose existing summarization models and approach the problem from a non-training stance. Specifically, Chapter 3 address segmentation and concatenation in between recursive hierarchical levels and Chapter 5 discusses non-training speech specific optimizations that improve longform dialog summarization.

Hierarchical Architectures in Summarization Models. The second approach towards summarizing longform text (and most commonly longform speech an dialog) is to develop new hierarchical attention [117, 118] transformer architectures[12] to increase long range modeling capacity.

[11] S^{j+1} would need to be segmented again into $\mathbf{S}^{j+1}$ before being used as $j+1$ level's input.

[12] These are in addition to the improved transformer models discussed in Chapter 2.5.

While hierarchical architectures are adept at modeling longer range input sequences, they do not fit our need of producing hierarchical summaries in various levels of detail. Thus, it is important to understand the difference and make a distinction between a *text level* hierarchical processing methodology and *model level* hierarchical architectural modification. Note that work in neural network architecture, namely imposing structure such as memory [119], and attention mechanisms [120] is a a popular and active area of research and go far beyond the scope of this related work portion.

In order to include more global context and better model long range dependencies[13], current works have proposed:

1. Adding recurrent hierarchical modules within a transformer block to propagate document level information at the token level [121].

2. Assuming a hierarchical latent top level structure to capture information at a coarser granularity to share with a bottom token level representation which captures finer details [122].

3. Encoding multiple documents in a hierarchical manner via a shared attention mechanism to learn latent inter-document relations [123].

4. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents [124] which uses a hierarchical attention encoder (RNN structure) to explicitly attend to important points in an input document.

In particular, to model long range dependencies in the meeting setting, HMNet (Hierarchical Meeting summarization Network) [125] creates a two-level hierarchical structure with a speaker turn level transformer and word level transformer. Though the authors encounter the same data scarcity challenge of insufficient spoken dialog summarization training data, they adapt news domain data and simulate a multi-person turn based meeting conversation.

[13]Long range dependencies refers to a concept in the source text that requires a reference to information contained in another section of the source text (typically temporally far away) or even different document to obtain a full context and properly comprehend.

2.7 Summarization Datasets

In this section we provide an overview and short description of select popular datasets used to train and evaluate summarization models. While the popularity of abstractive summarization datasets continues to grow, especially in the speech domain, it is still difficult and prohibitively expensive to collect. Unfortunately, for longform dialog, existing datasets (i.e. AMI [126] & ICSI [127]) are primarily focused on meetings and do not cover the variety of topics that are discussed in interviews and podcasts. In the following datasets, articles refer to the source text S_i.

1. CNN / DailyNews [128] consists articles and summaries from CNN (93k) and the Daily Mail (220k).
2. XSum [129] consists of BBC articles (227k) with single summary-sentences.
3. Newsroom [130]: consists of 1.3M human written articles and summaries from 38 sources between 1998 to 2017.
4. arXiv and PubMed [131] consists of longform papers from arXiv (113k) and PubMed (215k) where the paper's content is the source text and the provided abstract is the summary.
5. ENTSUM [132] is an entity focused dataset where human written summaries were a sequence of annotation instructions were given to write a summary prioritizing a particular entity.
6. GovReport [133] is a longform document summarization dataset from U.S. Government Accountability Office on national policy issues.

Spoken Dialog Domain Datasets

1. AMI [126] & ICSI [127] consist of longform meeting transcripts and summaries. The meeting transcripts contain an ASR word error rate of 36% and 37% respectively.

2. QMSum [134] consists of a query-based meeting summaries. Because content selection difficulty directly scales with source text length (i.e. it is clearly more difficult to summarize longer passages), authors try to guide annotators with targeted queries to keep summaries focused.

3. MediaSum [135] & SummScreen [136] are conversation summarization datasets where the source texts are speech transcripts and the summaries are mined from recaps and overviews.

4. SamSum [137] consists of short exchanges of text messages and summaries (16k), is one common dataset used to train and benchmark conversational summarization. Each text message source is very short, containing on average 94 tokens per conversation.

5. DialogSum [138] is a conversation summarization dataset (13.5k) that attempts to increase diversity compared to SamSum [137] and contains an average of 39.8% longer dialog for source text.

6. ForumSum [139] is a conversation summarization dataset (4k) containing human annotated summaries collected from internet forums.

7. Spotify Podcast Dataset [140] contains ASR transcriptions of over 100k podcasts. Authors hand annotate a small subset (303 summaries) to determine a quality threshold for using content creator (podcast author) provided descriptions. It is important to note that a content creator is not incentivized to write a high quality summary as much as writing an attention grabbing advertisement for a description.

Because no dataset with intermediate text level summaries exists, we are unable to compare the longform dialog browsing system's hierarchical summaries to human annotated gold standard summaries. Though, interestingly, summaries that were collaboratively written (albeit not for spoken dialog) using systems such as Wikum+ [42] should retain intermediate text representations that can be used as hierarchical summaries. In Chapter 5, we discuss how we harvest our hierarchical longform spoken dialog dataset's summaries while addressing challenges in the summarization such as subjectivity and content selection.

2.8 Text Evaluation for Automatic Summarization

Evaluating the a language model's text generation is essential towards understanding the model's performance and suitability for usage [141]. For this section, we give a background and overview of text evaluation methods concerning summarization; though it should be noted that many of automatic metrics used here are nearly identical in evaluating text generation in other tasks such as machine translation.

Human evaluation of text generation is generally considered to be the best metric and gold standard method for assessing generated text quality [142]. For the most part, humans are able to successfully coarsely estimate a text's quality based on innate heuristics; however, on a finer scale and for more nuanced tasks, there may be issues of human annotator bias annotator disagreement [143, 144]. Because of the time intensive nature of human evaluation, automatic evaluation procedures were developed to quickly and inexpensively assess a generated summary's quality. Work in human evaluation has proposed assessing summary quality through a "Pyramid" perspective with "Summarization Content Units" (a departure from previously the used DUC, Document Understanding Conference, evaluation procedure) to address the fact that no single summary itself is the best [145] when comparing multiple summaries. The procedure that [145] had outlined is also beneficial in showing not only the semantic content overlap between summaries, but also what content is missing across summaries. Subsequent work in Pyramid Evaluation [146] demonstrated its effectiveness and human correlation.

In an automatic text evaluation setting, there are three components: the source text (such as transcript) $S = (s_1, ..., s_n)$ which is encoded by the language model, the generated hypothesis (output summary) $H = (s_1, ..., s_n)$, and the reference summary text $T = (s_1, ..., s_n)$. Typically the hypothesis H is compared to the reference summary T, known as reference-based evaluation. Depending on data availability however, T may not always be provided and H's quality must be estimated leveraging other information such as S and external language models; this is known as reference-free evaluation.

For both of our systems, Chapter 3 and Chapter 5, we perform qualitative human assessments of summary quality. Due to the lack of references at the time of development, the automatic evaluation is in Chapter 3.3 is still limited and cursory; however, with gold summaries harvested in Chapter 4, we are able to more rigorously analyze the Chapter 5's system performance.

Human (Manual) Summary Evaluation For human evaluation, human annotators are tasked with assessing the quality of machine generated summaries, scoring along some predefined dimensions [90, 147]. This is known as *intrinsic* evaluation[14]. Different authors may use different terms for previously listed dimensions, though they typically encapsulate similar meanings. Some examples of these dimensions are:

1. *Fluency* or *Readability*: evaluates grammaticality, spelling, and syntax (such as capitalization and formatting). This is done without consulting the source text as it is content independent [148].

2. *Coherence* or *cohesion*: measures how well sentences are in relation to each other (do they form a coherent block of information?) [144].

3. *Accuracy* or *Factuality* or *Consistency*: evaluates the factual agreement between the source text and target summary, in other words, are claims made in the summary supported by the source text [149, 150].

4. *Informativeness*: how well does the summary capture the important ideas found in the source text, or colloquially, is the generated text useful as a summary [130].

5. *Completeness* or *Focus* [144]: how many of the "main ideas" in the source text are captured by the generated summary.

[14] *Extrinsic* concerns evaluating a model's performance at a specific task and is outside the scope of this section.

6. *Adequacy*: how much of the meaning in the original source text is also conveyed in the hypothesis [151]. This dimension was originally introduced in machine translation[15].

7. *Semantic Coverage*: How much semantic content units (meaning) in the reference summary is captured by the generated summary [145, 146].

8. *Relevance*: this is a term where conflicting definitions were provided and remains vaguely defined.

    - Evaluates how consistent the summary is with the source text. This can be rephrased as *"Are the details provided by the summary consistent (and appropriate) with details in the article?"* [130].

    - Assesses the content selection of the generated summary, does it contain only the important information [150]. This shares characteristics with *informativeness* and *completeness*.

Out of these dimensions, *adequacy* appears to be quite similar to *semantic coverage*. The key difference here is that the *adequacy* is comparing the the generated summary H to the information content contained in the entirety of input source text S, while *semantic coverage* is comparing the summary H to strictly less information contained in the gold standard reference summary R. It is also interesting to point out that the information summarized by the reference summary R is not always perfect: summarization is inherently subjective [152]. Thus, additional information contained in H could cause this particular summary to score well in *adequacy* while scoring poorly on *semantic coverage* given how *semantic coverage* is defined.

Completeness and *focus* also appear to be similar in *adequacy*, with the salient difference being an emphasis on "main ideas", which again is inherently subjective to readers. They (*completeness* and *focus*) also share characteristics with *informativeness*; the difference here can be found in that *informativeness* describes how well a generated summary expresses the important ideas in the source text whereas *completeness* and *focus* are concerned with how many (different) main ideas

[15]`https://catalog.ldc.upenn.edu/docs/LDC2003T17/TransAssess02.pdf`

are captured. Clearly many dimensions can be left up to human interpretation and lead to difficulty in obtaining consistent and unbiased assessments of generated summaries.

Reference Based Evaluation. Untrained automatic metrics measure some form of string overlap, content overlap, string distance, or lexical diversity [144] between the generated summary H and the reference summary R. The core idea behind these metrics is that the closer the predicted summary sequence is to the reference, the better of summary sequence. Some of the most common reference based evaluation metrics are:

1. BLEU [153] which aims to capture the adequacy and fluency of translations, but has been adapted for use in other text generation, such as summarization. This is calculated with a geometric mean of the precision of the hypothesis n-grams present in the reference n-grams. In other words, how many n-grams in the generated summary H are present in the reference summaries R.

2. ROUGE-n [7] (where n indicates the number n-grams used) which aims to assess semantic coverage and was developed for summarization. Unlike BLEU, ROUGE is a recall based metric that count how much reference summary R n-grams appear in the generated summary H.

3. METEOR [154] which improves upon BLEU by adding additional features such as stemming and synonym matching.

Furthermore, it is important for automatic metrics to correlate closely with human evaluation of summaries. In SummEval [150], authors comprehensively ranked 14 automatic metrics against human judgments along 4 dimensions (coherence, consistency, fluency, and relevance). In Table 2 [150], authors found that different ROUGE variants tend to correlate well with human judgment across the 4 dimensions.

More recent evaluation works propose different ways of leveraging pretrained language models for semantic evaluation of text such as BERTScore [155], which uses pre-trained contextual BERT embeddings to match a generated summary with a reference summary at the token level with cosine similarity. In a different vein, BARTScore [156] utilizes BART's pre-trained word probability distribution to score a generated sequence in 4 different categories. BLEURT [142] proposes several changes to training using synthetic data pairs where target labels are human ratings and automatic metrics in order to adapt a BERT model text evaluation.

Reference Free Evaluation. Recent research has also explored the far more difficult challenge of automatic evaluation instances where a reference summary R is unavailable. SUPERT [157] circumvents the lack of references by generating pseudo references and calculates the word mover's distance[16] between the generated summary and pseudo references. Wu et al. 2020 [158] train an evaluator with contrastive learning on augmented negative (poor artificial summary) examples to subsequently predict summary quality. In the machine translation domain, RUSE [159] employs learned sentence embeddings from three different models, paired with a multi-layer perceptron to predict translation quality. [160] finds that modifications to a multilingual model can match BLEU performance, demonstrating further promise in reference-free evaluation.

Annotator and Qualitative Evaluation Considerations. Unlike evaluating other more "constrained" instances of language modeling such as machine translation where the totality of correct translations is limited (i.e. even accounting for stylistic and vernacular considerations, there are only so many correct ways to translate a given sentence into another language), evaluating summarization may is inherently subjective and under constrained [152]. As a result, it is importantly to clearly define evaluation dimensions and properly scale annotators to close standards.

[16]This is similar to Earth Mover's distance but uses word embeddings to create a space where sentences with no shared words can still be semantically compared.

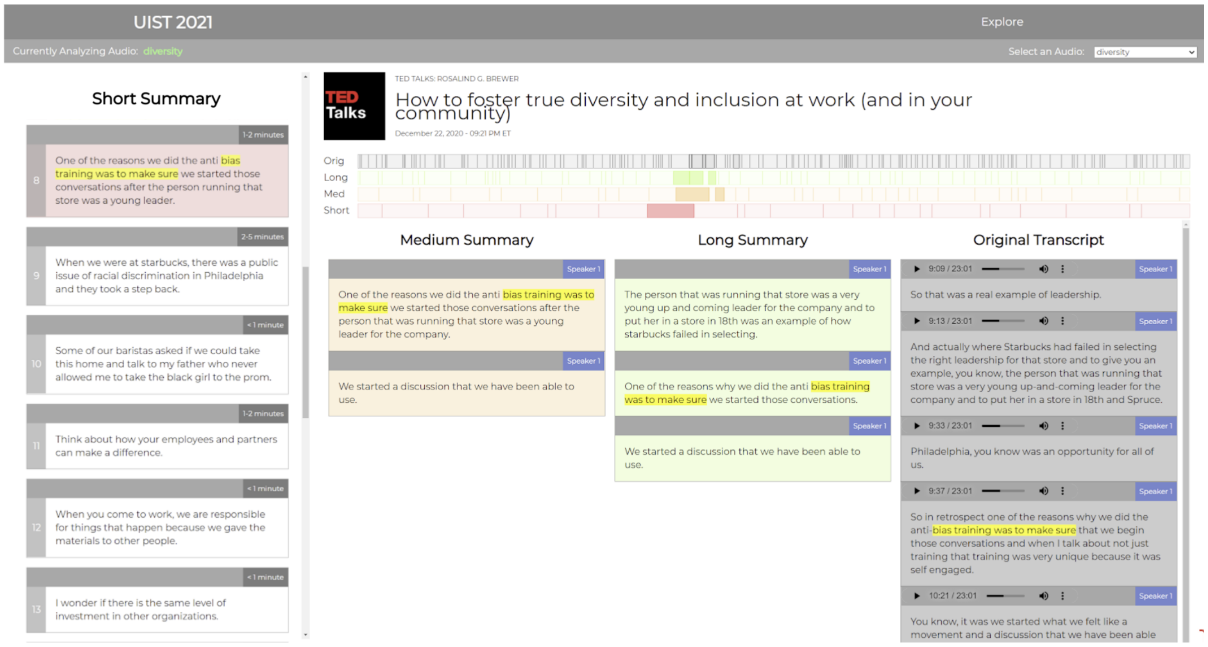
Chapter 3: Hierarchical Summarization for Longform Spoken Dialog

Chapter 3 lays the foundational framework and proof of concept system (again referred to as System 1) of decomposing longform spoken audio in a hierarchical nature and presenting it to the user in a user interface, enabling users to browse and navigate content to find things that are interesting to them.

We begin with the motivation in designing a system to browse longform dialog content (Section 3.1), along with selected related work (Section 3.2). We then conduct a formative study and investigate the impact of longform dialog on existing summarization language models and their implications for immediate practical usability (Section 3.3). From these findings, we develop a user interface and the necessary language modeling components (Figure 3.1, Section 3.4) to create a proof of concept longform audio browsing system (System 1). This is accompanied with a walk-through demonstration of how a user might use the system (Section 3.5). Lastly, we conduct user studies and evaluate (Section 3.7) the system.

3.1 Overview and Motivation

An ideal solution would be to automatically summarize the content and distill it to its most interesting points, but this is problematic for three reasons. First, despite many advances in machine learning, Automatic Speech Recognition (ASR) and summarization are not yet mature enough to accomplish this. Second, there is a question as to whether the ASR transcripts and summaries can be trusted to be accurate, especially in the presence of informal language, minimal structure, and speech disfluencies. Third, what each user wants from a summary will differ based on their previous knowledge and expertise on the subject matter – summaries are not one-size-fits all. This makes it difficult to provide training data for summaries that would be acceptable to a wide range of users, even if machine learning algorithms were perfectly accurate. We want to explore solutions that can leverage the strengths of machine learning, while overcoming many of its weaknesses.


UIST 2021
Explore
Currently Analyzing Audio: diversity
Select an Audio: diversity
TED Talks
Orig
Long
Med
Short

Short Summary

8 — 1-2 minutes — One of the reasons we did the anti bias training was to make sure we started those conversations after the person running that store was a young leader.

9 — 2-5 minutes — When we were at starbucks, there was a public issue of racial discrimination in Philadelphia and they took a step back.

10 — <1 minute — Some of our baristas asked if we could take this home and talk to my father who never allowed me to take the black girl to the prom.

11 — 1-2 minutes — Think about how your employees and partners can make a difference.

12 — <1 minute — When you come to work, we are responsible for things that happen because we gave the materials to other people.

13 — <1 minute — I wonder if there is the same level of investment in other organizations.

TED TALKS: ROSALIND G. BREWER

How to foster true diversity and inclusion at work (and in your community)

December 22, 2020 - 09:21 PM ET

Medium Summary

Speaker 1 — One of the reasons we did the anti bias training was to make sure we started those conversations after the person that was running that store was a young leader for the company.

Speaker 1 — We started a discussion that we have been able to use.

Long Summary

Speaker 1 — The person that was running that store was a very young up and coming leader for the company and to put her in a store in 18th was an example of how starbucks failed in selecting.

Speaker 1 — One of the reasons why we did the anti bias training was to make sure we started those conversations.

Speaker 1 — We started a discussion that we have been able to use.

Original Transcript

9:09 / 23:01 — Speaker 1 — So that was a real example of leadership.

9:13 / 23:01 — Speaker 1 — And actually where Starbucks had failed in selecting the right leadership for that store and to give you an example, you know, the person that was running that store was a very young up-and-coming leader for the company and to put her in a store in 18th and Spruce.

9:33 / 23:01 — Speaker 1 — Philadelphia, you know was an opportunity for all of us.

9:37 / 23:01 — Speaker 1 — So in retrospect one of the reasons why we did the anti-bias training was to make sure that we begin those conversations and when I talk about not just training that training was very unique because it was self engaged.

10:21 / 23:01 — Speaker 1 — You know, it was we started what we felt like a movement and a discussion that we have been able

Figure 3.1: **System User Interface.** Example of the system's summarization display for each unique audio file. The left part of the interface contains several short summaries which, when clicked, displays the medium and long summaries along with the corresponding original transcripts and audio clips. The top part of the interface shows what part of the transcript each summary encapsulates information about. When the top part is clicked, users can navigate to any part of the summaries or transcripts.

We present a system that produces hierarchical summaries of spoken dialog that allow a user to browse and navigate the content to find things that are interesting to them. Hierarchical summarization allows users to first see a high level summary of the content and then drill into progressively longer and more detailed summaries - or listen to the raw audio itself. This approach addresses two key issues:

1. It allows users to be in control of what information they read at a high level and what information they consume in greater detail.

2. When machine learning (ML) models makes mistakes in ASR and summarization, users can quickly recover the ground truth.

Although the typical approach to creating automated summarization systems requires training data that is difficult to obtain, our approach allows us to employ previously trained ML models recursively to generate shorter and shorter summaries. However, reusing models that were trained on different data requires careful model selection as well as novel algorithms to semantically segment the input text and thus output coherent summaries.

This system makes the following contributions:

1. An end-to-end system that automatically generates hierarchical summaries of longform spoken dialog.

2. A novel semantic segmentation algorithm that allows the reuse of existing machine summarization models rather than training a new one.

3. A user study demonstrating:

   (a) the system is 72% accurate in producing condensed *Short Summaries*.

   (b) system hierarchical features enable users to recover their understanding of 98% of summaries despite ASR and ML summarization model errors.

   (c) the average time that users spent to reach an understanding of an audio recording was 27% of the original audio length.

4. Qualitative findings about how people use *Short Summaries* as navigational tools to help them "skim" audio and find the content most interesting to them.

3.2 Related Work

We discuss the primary areas in natural language that our work builds upon[1]. Specifically, we leverage several of the techniques used in both the user studies and the summarization works to create our system.

Using NLP to Generate Multimodal Interactions

[1]Please refer Chapter 2 for the complete background related works.

Researchers have developed models and systems to easily navigate through videos and movies by navigating to the video clip and allowing users to interpret content [28, 29, 30]. However, these videos require users to search visual information in a video they may know little about and is inapplicable to pure audio files. To solve these issues, some researchers have employed summarizing key content in text as a means of helping users easily digest long-form content [161], [162]. More recent work has adapted the use of hierarchical information to provide users with multiple levels of summarization and information [38]. We build on top these systems targeting multi-party audio transcripts which pose novel challenges because these transcripts necessitate proper semantic segmentation to preserve meaning across speakers while simultaneously leveraging the usefulness of hierarchical information.

Still other work utilizes NLP to generate multimodal interactions such as images for video editing or even adding visuals to existing audio files [163, 164]. However, they rely on human-created transcripts, hurting the ability for the system to scale without automatic processes. Furthermore, visual representations only represent higher level abstract topics not the summarizations needed to represent the speaker.

Summarization of Multi-Party Audio

Creating meaningful summarizations from multi-party audio has been a difficult problem for researchers, often requiring hierarchical transformers and speaker segmentations to effectively retain information. Many of these papers, however, require full end-to-end training on transformers and even custom datasets [125, 165, 166, 167, 168]. Still others also employ graph-based summarization and coreferences to better summarize discourse [169]. Meanwhile, current unsupervised abstractive summarizations do not utilize deep learning summarization modules and require the use of word graphs and ranking algorithms [170]. These systems and models focus on learning end-to-end summarization which is not practical across multiple domains. Instead, we focus on utilizing these summarization systems as part of a larger unsupervised abstractive system to generalize and reduce the overhead needed to deploy and scale such a solution.

Recursive and Hierarchical Summarization Summarization of long complex material into recursively shorter and more tractable artifacts has been previously explored and found to provide an effective avenue for gaining useful comprehension of content [171]. Notably, this work showcased an interface displaying multiple summaries with varying levels of detail resulting in users having superior substantive recall and enabling non-linear exploration of the source material. However, this prior work employed crowd-sourced techniques to generate summaries and targeted solely threaded discussions typically found in forums. We build off these findings by developing a novel system employing automatic summarization and speech recognition techniques to spoken dialogue in order to generate a similar hierarchical exploration of content without requiring human-in-the-loop summary generation.

The utility of hierarchical summarization has also been shown for multimodel instructional videos that use audio and video to demonstrate each instructional step [38]. By using computer vision, ASR, and domain-specific heuristics they automatically group fine-level actions into coarse-level events (with summary text) that users can navigate at their own pace. We build on these ideas by using machine summarization to provide multiple levels of summarization detail and allow users not only better navigation but also time savings in consuming media.

3.3 Formative Study and Preliminary Investigation

There has been much progress on machine learning models for natural language processing, including ASR and summarization. If possible, we want to use existing pretrained models as a component of our system to avoid the costly process of collecting longform summarized speech training data, as none exist or are readily available. This is particularly difficult for summarization because every user may want a slightly different summary.

Because of spoken language noise effects in ASR transcripts, extracting transcript segments verbatim often leads to poor summaries. Therefore, we opt for the current state of the art abstractive summarization model, PEGASUS [109], which is able to achieve much higher human-quality summaries. Though promising, like most language models, it is important to note that PEGASUS is tailored towards specific benchmark datasets such as news or social media and that performance does not translate across different data domains, especially when applied to speech specific noise and disfluencies.

Moreover, there are two key problems:

1. ASR and summarization models are far from perfect and have inherent pre-existing challenges.

2. Summarization models are almost always trained on text rather than speech data. If a text trained summarization model is deployed on speech data, there would be a data domain mismatch, leading to considerably degraded model performance.

To evaluate the practical performance of existing ASR and summarization models and determine which models to use as the basis of our system, we investigate the following criteria:

1. *Coherency*, are the final output summaries coherent? If this constraint is not met, the model is not usable. Aside from re-training and adapting a model towards speech data, we have no tractable strategies for compelling model coherence.

2. *Information retention*, because output summaries are shorter and lossy, we check if they still retain salient information from the original passage. If a shortened summary does not contain useful or relevant information, it has no value.

In the formative study, we identified three models that had various summarization properties and tested each model's reusability. Each model was applied to seven different recordings and an automatic evaluation score was computed to determine the quality of the summarization. To further substantiate each model's summarization, we check each model's performance with qualitative analysis.

3.3.1 Evaluation Data

We evaluate on a test set of seven recordings of longform spoken dialog that span different topics, domains, and speech styles (Table 3.1). ***Important note:*** *this is the beginning of our cu-*

Transcript Name	Length	Word Count	Source	Edited?
NPR: M. Night Shyamalan	48 minutes	9184 words	How I Built This podcast	Yes
NPR: Chipotle	48 minutes	7847 words	How I Built This podcast	Yes
NPR: Health	29 minutes	5102 words	How I Built This podcast	Yes
NPR: Teach for America	22 minutes	3909 words	How I Built This podcast	Yes
Diversity and Inclusion	23 minutes	4201 words	Recorded Ted Talk Interview	No
Bill Ackman on Economy	29 minutes	5140 word	Recorded Bloomberg TV Interview	No
Ray Dalio on Economy	29 minutes	3971 words	Recorded Bloomberg TV Interview	No

Table 3.1: Dataset metadata used in formative study and final evaluation

rated dataset from Section 4. Section 4 builds off of this dataset and harvests further gold human annotated summaries that do not yet exist for this work.

The average length of each of the recordings is 32.5 minutes and the average word count output from ASR is 5622. Of these seven recordings, four are edited interviews from the NPR podcast "How I Built This", and 3 are unedited recordings from live events. Two are Bloomberg interviews regarding finance and one is a conversation about "How to foster true diversity and inclusion at work (and in your community)." These recordings were selected based on being content rich and of reasonable length. Information rich dialogue serves as a useful medium for this experiment by providing a sufficient density of information to showcase summarization. Additionally, choosing sources from the same producer reduces variance and provides a consistent structure for experiments. Finally, our experiment included both edited and unedited recordings of dialog to expose our system to both more coherent and structured conversations as well as free form dialogue.

3.3.2 Automatic Speech Recognition Model

For word recognition, we use a state of the art ASR model, publicly available with the Google Speech-to-Text API. This system is already robust to a variety of domains and speech noise, while providing features such as diarization (speaker detection) and punctuation prediction. While we suspect the ASR component will not be a large contributing factor to poor summarization, we conduct a brief investigation on word recognition errors (word errors, i.e. homonyms such as `weather` compared to `whether`) as they could non-trivially impact downstream summarization

Model	Domain / Fine-Tune Data	Max Words	Output Size
`M1`	XSUM News / BBC News	64 words	1 sentence
`M2`	News / CNN, DailyMail	128 words	3-5 sentences
`M3`	Paraphrase / Quora, PAWS	60 words	1 sentence

Table 3.2: Model nomenclature where **`M`***i* indicates Model *i*, training data descriptions, and model maximum input and typical output sizes.

performance.

3.3.3 Summarization Models

For summarization, we investigate the current abstractive state of the art language model PEGASUS [109]. While PEGASUS is noticeably improved over other summarization methods in terms of producing human level quality summaries, it requires fine-tuning onto domain specific summarization data. It is also important to note that a pre-trained only instance of PEGASUS is not normally used without modification; the pre-training procedure is different from summarizing and the authors focus solely on fine-tuned downstream summarization datasets. Appropriately, we select fine-tuned instances from `huggingface.co` [172] that generate complete and grammatically correct passages (i.e. not a few keywords) and are still in considerably general domains (i.e. not a medical field model instance) to assess PEGASUS coherence and information retention. Model details are given in Table 3.2.

We begin by processing audio files to obtain raw ASR transcripts. We use three instances of a PEGASUS model (**`M1, M2, M3`**) that are fine-tuned according to Table 3.2; all models have the same architectures. However, because of the nature of longform dialog, the number of words per transcript greatly exceeds the maximum input length that **`M1, M2, M3`** can accept. Transcripts must be processed and split into manageable lengths. We naively segment the transcript in fixed 60 word length segments set to **`M3`**'s maximum input length[2]. For example, if an input transcript segment had a total of 154 words, it would be broken into a list of 3 individual segments,

[2]We also experimented with increasing the input size to 128 for **`M2`**, but still observed poor results (in fact, noise artifacts and incorrect model behaviors were more exaggerated than using 60 word length segments)

each containing $[60, 60, 34]$ words. To maintain evaluation consistency across all models, any evaluation involving naive fixed segmentation is set to 60 words. These are then summarized by **`M1`**, **`M2`**, and **`M3`**, which are set to output summaries containing at most half of the original passage's words.

3.3.4 Heuristic Score

We evaluate a summarization model's coherency and information retention using a heuristic score consisting of state of the art automated metrics in natural language processing. It is important to note that we do not yet possess gold human annotated summaries for our dataset (Section 2.7) that are later collected in Section 4).

For coherence evaluation, we use a `BERTScore` [155] between a reference ASR segment and a model generated summary (candidate input). This method correlates well with human evaluation and uses word level contextualized embeddings to capture dependencies and word ordering. For retained information, we use the cosine similarity between `Sentence Transformer` [173] embeddings of a reference ASR segment and a model generated output summary. A higher cosine similarity between the reference ASR segment and output summary suggests the summary captures the reference ASR segment's semantic content. The final heuristic is the simple average of the two and has a range of $[-1, 1]$. In practice, cosine distance based metrics used to determine similarities between word embeddings are positive, with a general range of $0 - 0.5$ for a weak correlation, $0.5 - 0.8$ for a moderate correlation, $0.8 - 1$ for a strong correlation, and 1 for a perfect correlation [174]. As a sanity check, we observe a correlation of 1.0 when we set the reference and candidate text inputs to be the same. Intuitively, as **`M1`**, **`M2`**, and **`M3`** outputs are still summaries, they will contain at least some semantic similarity to the reference ASR segment; therefore we expect to observe a somewhat moderate correlation $(0.5 - 0.6)$ with our heuristic. After determining which model can be feasibly re-purposed, we use the heuristic score again to evaluate our method's impact towards improving summarization.

3.3.5 Heuristic Score Limitations

In evaluation, a longer reference text (an ASR segment) and a model's generated summary are passed in as inputs to an evaluation model that computes a numerical score describing the summary's accuracy. Usually a text generation's quality estimation focuses on adequacy (content faithfulness) and fluency (coherence). Unlike typical evaluation, our dialog's evaluation setting is unsupervised which poses an extra set of challenges.

Our heuristic adopts two individually unsupervised components, `BERTScore` and `Sentence Transformer` to measure adequacy (information retention, *or semantic content*) and to a lesser degree fluency (coherence *or grammaticality*). Evaluating a generated summary's fluency without a reference is notably challenging (discussed below) but is, to some extent captured, within `BERTScore`'s token embedding matching.

Unsupervised systematic and automatic evaluation of a summarization model's text generation is particularly difficult as it requires comparing generated sentences to non-existent annotated references. The benefit of the unsupervised reference-free evaluation setting of the heuristic score fundamentally incurs similar trade-offs to that of the underlying automatic evaluation methods `BERTScore, Sentence Transformer`.

Because of the lack of human authored summaries in both training and evaluation data, we are unable to use standard summarization evaluation standards such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [7]. Moreover, existing *n*-gram evaluation metrics cannot be effectively used on summaries as they tend to fail at robustly capturing performance; shortened paraphrases containing semantically critical changes in word positions may be unfairly penalized due to a mismatch with the reference text.

One subsequent trade-off is the possibility of achieving adequacy at the cost of low fluency. Recall that semantic content is compared to an entire (unsummarized) dialog segment as a reference; thus a model can theoretically fool the metric by outputting many *shorter* and *less relevant* segments in an incoherent manner instead of a single relevant segment to achieve a better score, as is the case with **`M2`**'s marginally better score over **`M3`**'s.

ASR Transcript Segment	Summary	Description
opportunity is inclusion because you know, I have heard this stories so many times about there's you know, no Black Talent out there. No Latin next talent for these particular roles. The talent is out there. I will tell you that is underdeveloped because I think we have spent more time trying to reach numbers. Then we have changing our `[60 words]`	**`M1`**: *In our series of letters from African-American journalists, journalist and columnist Richard Lawson* looks at the importance of inclusion in the workplace.	**Hallucinations**, **`M1`** produces words and concepts (*italicized red*) in the summary are not present in the given input.
	`M2`: opportunity is inclusion because you know i have heard the stories so many times about theres no black talent out there no latin next talent. i think we have spent more time trying to	**Speech noise sensitivity**, **`M2`** loses abstractive summarization ability and outputs unimportant sections.
	`M3`: I think we have spent more time trying to reach numbers than we should because there is no black or latin talent out there.	**Semantic Inaccuracy**, **`M3`** combines two different sentences into a semantically incorrect output.

Table 3.3: Example summary generations on the "Diversity and Inclusion" dialog showing an instance of the ASR transcript for one speaker turn and errors frequently made by the three summarization models. Text is color coded to indicate shared regions between the input and output.

3.3.6 Formative Study Findings

We discuss Table 3.3 throughout this section. It contains an example of the ASR transcript segment of one speaker in the "Diversity and Inclusion" recording and the corresponding summaries generated by the three models. Text is color coded to indicate shared regions between the ASR transcript and the summary.

3.3.7 Google Speech-to-Text Automatic Speech Recognition Quality

We quickly and quantitatively evaluate the word error rate (WER) of the ASR system. Because Table 3.3 only consists of audio data and perfect transcripts (human transcribed) are non-existent, we benchmark ASR performance on a random subset of Ted Talks as they are somewhat similar in terms of speech and data structure to Table 3.3 and thus would likely be indicative of ASR performance. We find an average WER of 10%[3], slightly above the reported 6.7% WER [175],

[3]This number should be treated as an upper bound as the human transcribed transcripts contain artifacts such as "`(Applause)`" or "`(Laughter)`".

and far below a usability constraint of 30% [176].

As seen in the provided ASR Transcript example in Table 3.3, the ASR Speech-to-Text makes very few errors. However, rare words, unfamiliar phrases, or new words not yet encountered still degrade performance. For example, in the NPR: Chipotle dialog, *"mise-en-place"* was mistakenly transcribed as *"knees in place"*. In the "Diversity and Inclusion" dialog, *"rectangle. Opening"* was mistakenly transcribed from *"reckoning"*. Additionally, performance can fluctuate due to a variety of noising factors such as speech disfluencies, foreign accents, and audio recording quality. Although ASR makes few errors, they will propagate to downstream tasks and create challenges for generating a practical audio summarization system.

Model Name	Segmentation Strategy	Heuristic Score
`M1`	Naive Fixed Length	0.61
`M2`	Naive Fixed Length	0.70
`M3`	Naive Fixed Length	0.68

Table 3.4: Automatic evaluation heuristic scores for various segmentation strategies.

Summarization Model Quantitative Analysis

Table 3.4 gives the automatic evaluation heuristic scores for **`M1`**, **`M2`**, and **`M3`** ranging from 0.61 – 0.70. Despite generating summaries on out of domain speech data, we can conclude that all the baseline language models can still reasonably function and retain a moderate amount of information with a summary containing at most half the words as the input ASR segment. Nonetheless, the the tight spread of the heuristic score range indicates a moderate correlation and merits further investigation into the re-usability of **`M1`**, **`M2`**, and **`M3`** to fully understand model behaviors. While the heuristic score is telling, it is not a replacement for human level evaluation; it provides only a limited perspective into performance that is subject to intrinsic methodology constraints. To get a sense of what types of errors the automatic summarization models are making and whether they could potentially be addressed, we studied various segments by hand.

Summarization Model Qualitative Analysis

This style of evaluation was not formal; the errors were pronounced, ubiquitous, and immediately apparent. Such poor performance severely impeded practical usability and therefore did not necessitate a formal evaluation. Unfortunately, we observe that all three summarization models make frequent and substantial errors; however, **`M3`** stood out as containing problems that were addressable.

`M1` produced summaries that contain frequent hallucinations [69] – phrases or entities that appear to be semi-relevant but are not actually present in the underlying text. This can be attributed to its news based training data.

For example, in Table 3.3 **`M1's`** summary contains the text "African-American journalists" and "Richard Lawson." Neither of these entities are mentioned in the input (or entire audio file). However, these entities are in **`M1's`** training data. This is a typical problem seen in language models when deployed on new data that is not encountered in training. Only recently, an attempt at fixing hallucinations has resulted in improved ROUGE precision and increased human preference [177], but still requires additional dataset generation. These errors are in almost every summary produced by **`M1`**. Thus, fixing **`M1`**'s hallucinations would be nontrivial and require a new training dataset.

`M2` does not contain hallucinations but unfortunately it introduces many grammatical errors and performs especially poorly with regards to fluency: sentences trail off without finishing and summaries consist of concatenated phrases that may be individually sensible but holistically incomprehensible. Moreover, it fails to produce an abstractive summary and defaults to an extractive behavior; it mostly picked sections of the input rather than summarizing the entire input. This is likely because **`M2`** is trained to produce longer summaries than **`M1`**, and thus it is not forced to produce abstractive summaries. Reiterating Section 2.4, it is essential for a speech summarization model to be abstractive. These errors are frequently in summaries produced by **`M2`**.

`M3` has more fluent text with no hallucinations. However, it makes an egregious error of misrepresenting the content. The transcript clearly states that '*'The [Black and Latin] talent is out* *there,"* but the summary introduces a negation to say that the talent is not there. The root of this problem is that `M3` coerces two different segments into a semantically incorrect summary. These errors occur when multiple non-sequitur or different topics are provided as a single input. Because abstractive summarization generates words that are not necessarily present in the source input text, they require a high degree of content understanding of the underlying semantic information in the passage [70] to successfully generate a semantically faithful summary; a poorly segmented input containing multiple different concepts would be exceedingly detrimental towards a model's semantic comprehension. Thus, `M3`'s resulting coherent and abstract summaries (albeit with contextual misrepresentation errors) signal that:

1. A successful semantically accurate segmentation that can group similar topics and ideas together, while splitting dissimilar sentences into a separate chunk can improve a model's semantic comprehension, and transitively improve summary generation accuracy.

2. The summary context's input accuracy issue is now reframed as a processing challenge that does not require changes to the model's architecture, re-training, or additional annotated training data.

3. `M3` is able to maintain its abstractive nature, which is essential to summarizing dialog due to speech disfluencies and other noise artifacts.

Formative Study Key Takeaways

Based on this exploration of the three models, we hypothesize that `M3` is the best one to build on top of and reduces the challenge of practical dialog summarization to a tractable problem. `M1` and `M2` errors are exceedingly difficult to correct without significant amounts of specialized speech training data. `M2`'s marginally higher score over `M3` is immaterial given `M2`'s disfluency and incoherence. Although incoherent topic grouping is rarely the case in written language where ideas are well-formed and presented in a manner that is optimized for ease of understanding, it is usually the norm in spoken language where topics shift over time as speakers react to the last thing that was said. Concretely, if we can segment transcripts into semantically cohesive segments, creating easier inputs and facilitating improved summarization performance, `M3` may be an effective summarization tool. When errors do remain, there is the fallback of the user using the hierarchical browsing features to investigate surprising or suspicious claims to see if the summary is consistent with the text.

3.4 System

We present a system that produces hierarchical summaries of spoken dialog that allow the user to browse and navigate the content to find things that are interesting to them. Hierarchical summarizing allows users to first see a high level summary of the content and to then drill into progressively longer and more detailed summaries - or listen to the raw audio itself.

As shown by our formative study, pre-existing technology performance drastically suffers when applied to speech and is still considerably below the requirement for practical usage. Therefore, in addition to the borrowed pre-existing ASR system (Google Speech-to-Text API) and language summarization model (`M3`, paraphrasing adapted PEGASUS), we develop a method to identify semantically related segments of text that can be input into the summarization model, then merged back together to maximize coherent summaries. This process can be done recursively to get increasingly shorter and more abstractive summaries.

The core technical novelty and contributions within our speech summarization framework are as follows:

1. A novel segmentation algorithm that creates semantically similar input blocks from an input ASR segment in order to maintain conceptual cohesiveness
2. A semantic hierarchical clustering algorithm that joins conceptually similar ideas for logical subsequent (recurrent) abstract summarization

The inclusion of these procedures to the two-stage hierarchical framework enables not only improved grammatical and semantic cohesiveness but also facilitates various levels of summarization detail, which can be intuitively thought as the following:

1. *Long Summary*: Cleaned ASR Transcript. At this stage, a transcript's disfluencies and noises are cleaned and presented according to conversational order or speaker turns.

2. *Medium Summary*: Moderately Detailed Summarization. Similar *Long Summaries* are merged and further paraphrased, providing key concepts along with essential details.

3. *Short Summary*: High level Summarization. Similar *Medium Summaries* are further merged to provide the transcript's salient ideas in more concise language.

3.4.1 Interface

The interface (Figure 3.1) consists of three main sections: *high level summary* column on the left, the *segment data view* in the middle and the *timeline of segments* at the top of the interface. Users explore the content by first browsing the *Short Summary* column to get a high level overview of the content.

Users may click a *Short Summary* to see summaries of different length and additional levels of abstraction (*Medium and Long Summaries*) as well as the ASR transcript, or elect to listen to the corresponding audio. Yellow highlighting shows a key phrase in the short summary and it's corresponding phrase in the other summaries and the ASR transcript to help orient readers as they move from reading the short summary to the other summaries of the same underlying text. The timeline of segments shows how all the summaries are aligned. The user can see some *Short Summaries* that cover longer portions of the original transcript than others. Clicking on the timeline will take the user to the summaries of that section.

The interface was designed with two goals in mind:

Design Goal 1: Enable users to quickly identify useful information to them. Presenting high level summaries to the reader allows them to quickly grasp a general idea of what is being said. However, simply reading *Short Summaries* may not entirely satisfy the reader. By nature of being summaries, they may omit details that may be of interest. Additionally, the automatic summarization algorithms are imperfect and sometimes present summaries that are more vague than a user would prefer. However, the purpose of the *Short Summaries* are not necessarily to fully summarize, but to allow the user enough information scent [178] to decide if they want more detail. If they want more detail, they can use the hierarchy of summaries to read *Medium* or *Long Summaries*, read the ASR transcript, or listen to the underlying audio.

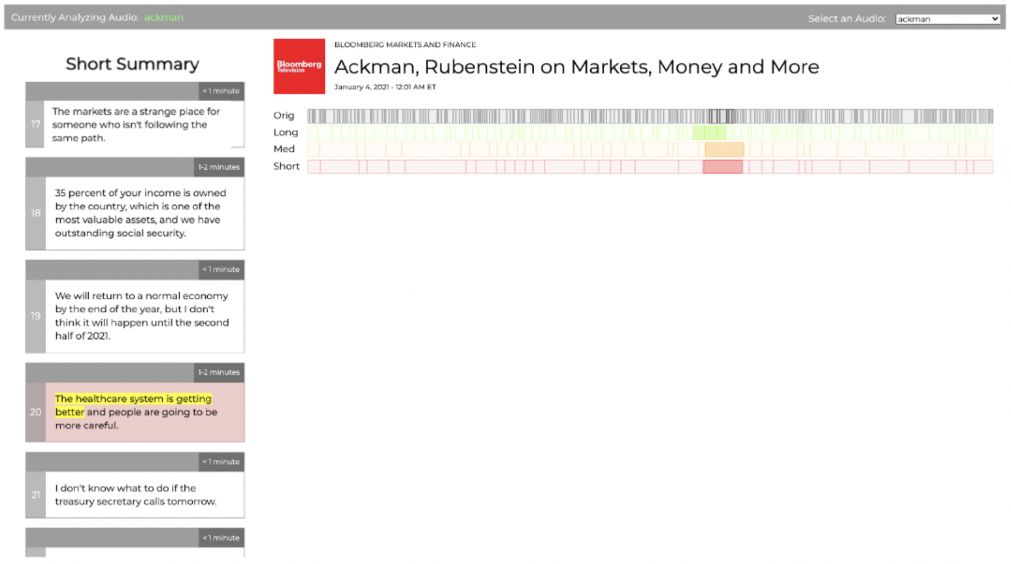
Design Goal 2: Support error recovery. As both automatic speech recognition and summarization may produce errors at various stages of the system, the interface provides multiple tiered layers of information for users to fall back on in order to recover comprehension of any given set of summarization data in the event that either a portion of the transcribed audio or summaries has erroneous text.

Listening to the raw audio will provide the full information a user should need to recover from confusion or loss of comprehension due to an ASR or summarization error. However, listening to audio takes longer for most people than reading text. If users wants near-full fidelity information in a form they can read (or scan), they can refer to the ASR transcript. Many find transcripts of dialog difficult to read because of the informal language and speech disfluencies. Thus, users may find the *Long Summaries* easier to read - they retain almost all the information of the ASR transcript, but the text is cleaned up to remove these artifacts of speech. Recall that **`M3`** is an instance of PEGASUS trained on a paraphrasing dataset, resulting in summaries that tend to be cleaned of superfluous text, including speech artifacts. Users who want more actual summarization can refer to the *Medium Summary*.

By presenting users with these options for recovering from ASR and model errors, users can decide how much time and effort they want to put into recovering from the error. However, there is a possibility that offering users multiple options could provide a negative experience by overwhelming them with choices. Over time, we expect users will become familiar with the nature of each level of detail and get a sense of which option to select. This is an issue we address in the Section 3.7.

3.5 System Interface Walkthrough

To give an idea of how the system can be used, consider the following Figures: 3.2, 3.3, 3.4 with the given 29 minute podcast on financial markets hosted by David Rubenstein interviewing Bill Ackman. These figures demonstrate an example thought process of a user browsing the podcast content and how the hierarchical structuring of the system allows the user to further dive deeper into content he or she finds interesting.



Short Summary

BLOOMBERG MARKETS AND FINANCE
Ackman, Rubenstein on Markets, Money and More
January 4, 2021 - 12:01 AM ET

17 — < 1 minute — The markets are a strange place for someone who isn't following the same path.

18 — 1-2 minutes — 35 percent of your income is owned by the country, which is one of the most valuable assets, and we have outstanding social security.

19 — < 1 minute — We will return to a normal economy by the end of the year, but I don't think it will happen until the second half of 2021.

20 — 1-2 minutes — The healthcare system is getting better and people are going to be more careful.

21 — < 1 minute — I don't know what to do if the treasury secretary calls tomorrow.

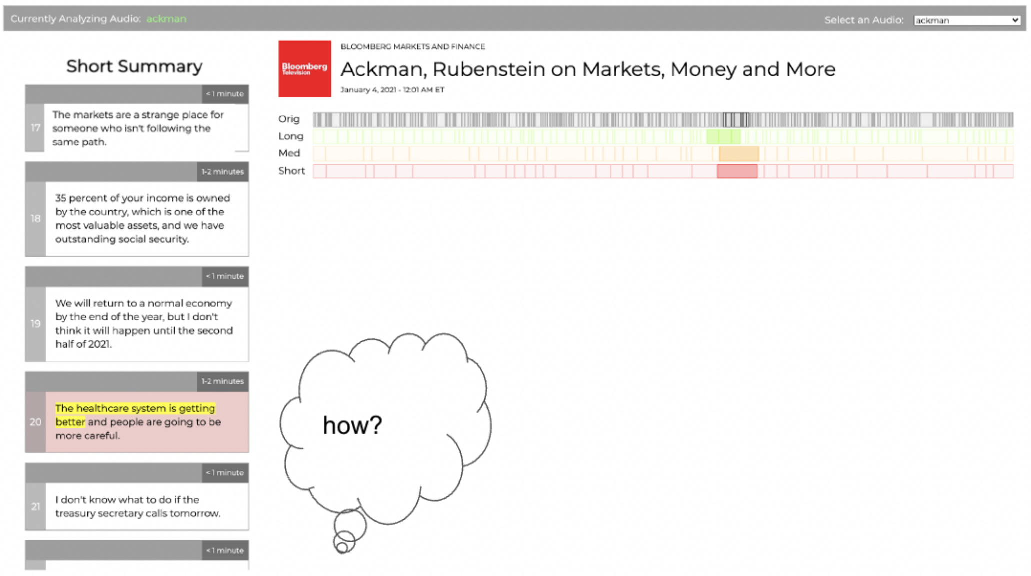
< 1 minute

Figure 3.2: **System View Part 1.** Initially in this view, the user is presented with *Short Summaries* that capture the most salient ideas in a podcast. The user can read these in a quick and linear fashion.

3.6 Summarization Algorithm and Implementation

Figure 3.5 shows the steps through which an audio file is recurrently processed to obtain different levels of summarization (*Short, Medium, Long*). At a high level, the system segments an ASR transcript and iteratively summarizes previously combined conceptually similar segments to obtain increasingly abstract summaries while preserving semantic meaning. Note, the system is inference only and is not trained.

Stage 1 (Fig. 3.5) of the system employs ASR to create a speaker diarized transcript of the input



BLOOMBERG MARKETS AND FINANCE

Ackman, Rubenstein on Markets, Money and More

January 4, 2021 - 12:01 AM ET

Short Summary

17 — <1 minute — The markets are a strange place for someone who isn't following the same path.

18 — 1-2 minutes — 35 percent of your income is owned by the country, which is one of the most valuable assets, and we have outstanding social security.

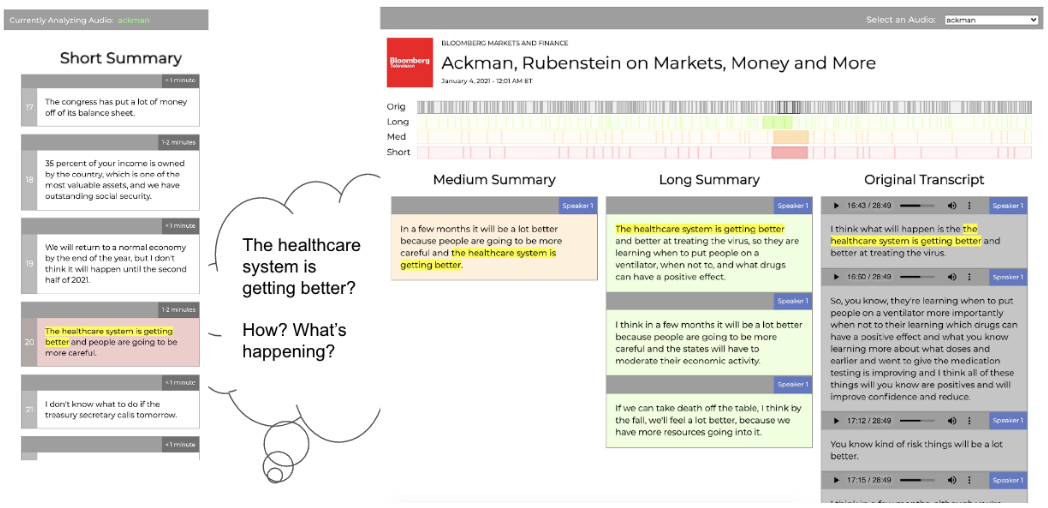
19 — <1 minute — We will return to a normal economy by the end of the year, but I don't think it will happen until the second half of 2021.

20 — 1-2 minutes — The healthcare system is getting better and people are going to be more careful.

21 — <1 minute — I don't know what to do if the treasury secretary calls tomorrow.

<1 minute

Figure 3.3: **System View Part 2.** Upon further inspection, the user decides that the 20th *Short Summary* piques his/her particular interest. The user finds the summary "the healthcare system is getting better and people are going to be more careful" particularly interesting as it is widely known the United States healthcare system is not the greatest and decides to further investigate to hear the podcast's perspective on this issue.


Currently Analyzing Audio: ackman
Select an Audio: ackman
Orig
Long
Med
Short
The healthcare system is getting better?
How? What's happening?

BLOOMBERG MARKETS AND FINANCE

Ackman, Rubenstein on Markets, Money and More

January 4, 2021 - 12:01 AM ET

Short Summary

17 — <1 minute — The congress has put a lot of money off of its balance sheet.

18 — 1-2 minutes — 35 percent of your income is owned by the country, which is one of the most valuable assets, and we have outstanding social security.

19 — <1 minute — We will return to a normal economy by the end of the year, but I don't think it will happen until the second half of 2021.

20 — 1-2 minutes — The healthcare system is getting better and people are going to be more careful.

21 — <1 minute — I don't know what to do if the treasury secretary calls tomorrow.

<1 minute

Medium Summary

Speaker 1 — In a few months it will be a lot better because people are going to be more careful and the healthcare system is getting better.

Long Summary

Speaker 1 — The healthcare system is getting better and better at treating the virus, so they are learning when to put people on a ventilator, when not to, and what drugs can have a positive effect.

Speaker 1 — I think in a few months it will be a lot better because people are going to be more careful and the states will have to moderate their economic activity.

Speaker 1 — If we can take death off the table, I think by the fall, we'll feel a lot better, because we have more resources going into it.

Original Transcript

16:43 / 28:49 — Speaker 1 — I think what will happen is the the healthcare system is getting better and better at treating the virus.

16:50 / 28:49 — Speaker 1 — So, you know, they're learning when to put people on a ventilator more importantly when not to their learning which drugs can have a positive effect and what you know learning more about what doses and earlier and went to give the medication testing is improving and I think all of these things will you know are positives and will improve confidence and reduce.

17:12 / 28:49 — Speaker 1 — You know kind of risk things will be a lot better.

17:15 / 28:49 — Speaker 1

Figure 3.4: **System View Part 3.** By clicking on the *Short Summary* 20, the user can see the progression of detail from reading the *Short Summary* to the *Medium Summary*. The *Medium Summary* provides the user with context that the speaker is referring to healthcare in the context of recent months. Not the overall US healthcare system as a whole. Further investigating and reading the *Long Summary* segments, the user finds that the healthcare system is in reference to getting better at treating the coronavirus with ventilators.

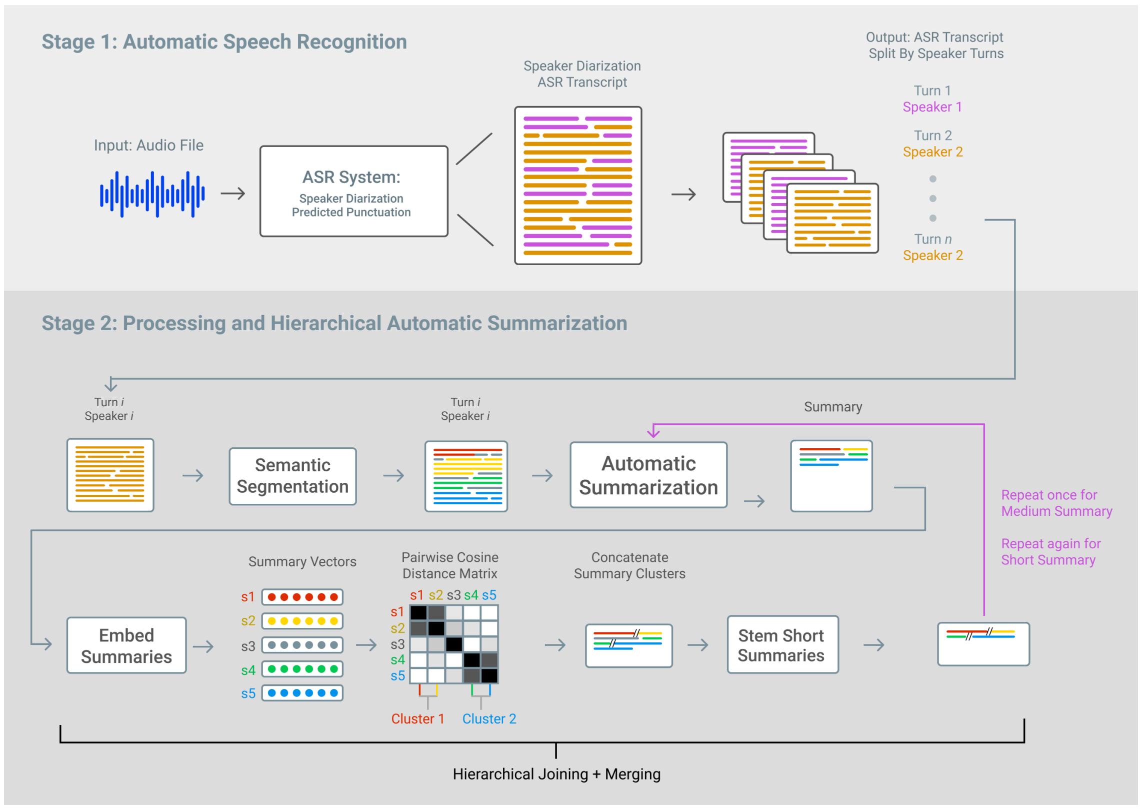


Figure 3.5: **Summarization Generation Pipeline.** Our system enables the conversion of audio files to multiple tiers of summarization. In the first stage, we convert the audio file into a speaker-segmented and punctuated transcript and process the transcript, split by speaker turns. In the second stage, we take each speaker turn and cluster conceptually similar summaries via semantic segmentation. Each cluster's summaries are joined (concatenated) based off of semantic similarity. We remove small summarizations and then repeat the summarization and merging process to obtain the *Medium* and *Short* summaries.

audio file. These speaker turns are further processed by a semantic segmentation algorithm which divides a given speaker turn into chunks of semantically related sentences. The now refined speaker turns are iteratively given as inputs into stage 2 (Fig. 3.5) of the system where processing and hierarchical automatic summarization occurs. After each speaker turn is individually summarized, its summaries are embedded which are then used to cluster sentences of the summary. Clusters are concatenated (merged) and shorter summaries, which generally contain little salient information, are stemmed. The first resulting summary of this iteration through the pipeline represents a *Long Summary*. This *Long Summary* is fed back into the the automatic summarization model and follows the same embedding, hierarchical clustering, and stemming steps once more to generate a *Medium*

Summary. One further cycle using the *Medium Summary* yields a *Short Summary*.

Details of each system component are as follows:

Automatic Speech Recognition. We begin by using the Speech-To-Text Google API to obtain transcripts with speaker diarization and predicted punctuation for initial sentence boundaries. Speaker turns are alternating blocks of text separated by changes in speaker; they provide a very coarse starting point for transcript segmentation. Speaker turns frequently discuss multiple different ideas and may result in a long monologue before another speaker interjects.

Coreferenced Semantic Segmentation. The procedure is given in Algorithm 1 and a visualization is given in Figure 3.6. We directly refer to variables in the pseudocode, in the following walk through and explanation of the algorithm.

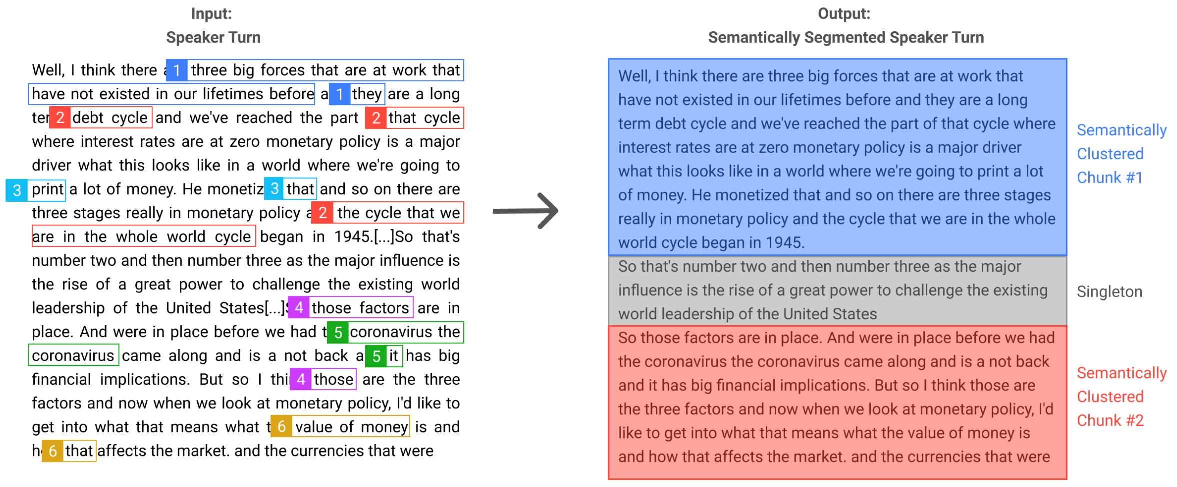
To understand why we employ coreference resolution [179] and speaker shifts to semantically link sentences together and correct poor segmentation, we must recognize two linguistic tendencies:

1. Unlike written prose, conversation can be far more vague; nouns and objects, herein referred to as entities, are usually initially mentioned and then sporadically referenced, while all other mentions are pronouns (it, s/he, they, etc.). Generic topic modeling of dialogue performs poorly due to the nature of conversations, since conversations have both local and global topic structures that have weak signals in conversation [180]. Individuals can talk about a topic using specific references, but a model not trained to recognize these topics could fail to recognize the boundaries of the topic effectively. To eliminate a dependency on custom training data, we instead choose to identify expressions that refer to the same entity using coreference resolution. To employ this technique, our algorithm seeks to group sentences spanned by an entity, which is defined as the sentences contained by the start and end of the entity. This method of using coreferenced entities to model text has historically been shown to be successful [181] [182] and is still used in current state of the art models [169]. We use the publicly available Allen NLP API [183] for state of the art coreference resolution [184].

2. Speaker shifts may begin relatively different concepts [185] and repeated references to the

Algorithm 1: Coreferenced Semantic Segmentation

```
Input: List $\mathbf{T}_{in}$ of $n$ speaker turn segments $\{S\}_{i=0}^{n}$ where word $w \in$ sentence $s \in S$,
       hyperparameters $m = 100$ for maximum coreference word span and $p = 3$ for
       minimum number of coreference mentions.
$\mathbf{T}_{out} \leftarrow$ list() # all semantically segmented speaker turns;
Initialize $e_0, e_f$ # coreference span start and end pointers;
Initialize $\mathbf{B} \leftarrow$ list("I", "me") # stop tokens;
for $i = 0, 1, ..., n$ do
    $\mathbf{C} \leftarrow$ Coreference($S_i$) # Allen NLP API;
    for coreference entity $c \in \mathbf{C}$ do
        if $c.span > m$ and $|c| < p$ then
            delete $c$ from $\mathbf{C}$;
        if $c \subseteq \mathbf{B}$ then
            delete $c$ from $\mathbf{C}$;
    $\mathbf{S}_{new} \leftarrow$ list() # instantiate new speaker turn block;
    $\mathbf{P} \leftarrow$ list() # semantic topic cluster within speaker turn $\mathbf{S}_{new}$;
    $e_0, e_f \leftarrow c_{min}$ with $min(w_0 \in c \in \mathbf{C})$ # entity with earliest word index;
    $\mathbf{C}_{used} \leftarrow$ list($c_{min}$);
    for $s \in S_i$ do
        $s_0, s_f \leftarrow w_0, w_f \in s$ # start and end word indices of $s$;
        for coreference entity $c \in \mathbf{C}$ and $c \notin \mathbf{C}_{used}$ do
            $c_0, c_f \leftarrow w_0, w_f \in c$ # start and end word indices entity $c$;
            if $s_0 \leq e_0$ and $e_f > s_f$ then
                $\mathbf{P}$.append(s) # s within span, add to semantic block;
                if $c_f > e_f$ and $s_0 \leq c_0 \leq s_f$ then
                    $e_0, e_f \leftarrow c_0, c_f$ # update maximal entity span;
                    $\mathbf{C}_{used}$.append($c$);
                break;
            else
                $\mathbf{S}_{new}$.append($\mathbf{P}$) # add topic cluster to speaker block;
                $\mathbf{P} \leftarrow$ list(s) # begin new topic cluster;
                $e_0, e_f \leftarrow c$ with $min(w_0 \in c \in \mathbf{C})$ and $c \notin \mathbf{C}_{used}$;
                $\mathbf{C}_{used}$.append($c$);
                break ;
    $\mathbf{T}_{out}$.append($\mathbf{S}_{new}$) # add semantically segmented speaker;
return $\mathbf{T}_{out}$ # all semantically segmented speaker turns
```


Input:
Speaker Turn
Output:
Semantically Segmented Speaker Turn
Semantically Clustered Chunk #1
Singleton
Semantically Clustered Chunk #2

Well, I think there a 1 three big forces that are at work that have not existed in our lifetimes before a 1 they are a long ter 2 debt cycle and we've reached the part 2 that cycle where interest rates are at zero monetary policy is a major driver what this looks like in a world where we're going to 3 print a lot of money. He monetiz 3 that and so on there are three stages really in monetary policy a 2 the cycle that we are in the whole world cycle began in 1945.[...]So that's number two and then number three as the major influence is the rise of a great power to challenge the existing world leadership of the United States[...]S 4 those factors are in place. And were in place before we had t 5 coronavirus the coronavirus came along and is a not back a 5 it has big financial implications. But so I thi 4 those are the three factors and now when we look at monetary policy, I'd like to get into what that means what t 6 value of money is and h 6 that affects the market. and the currencies that were

Well, I think there are three big forces that are at work that have not existed in our lifetimes before and they are a long term debt cycle and we've reached the part of that cycle where interest rates are at zero monetary policy is a major driver what this looks like in a world where we're going to print a lot of money. He monetized that and so on there are three stages really in monetary policy and the cycle that we are in the whole world cycle began in 1945.

So that's number two and then number three as the major influence is the rise of a great power to challenge the existing world leadership of the United States

So those factors are in place. And were in place before we had the coronavirus the coronavirus came along and is a not back and it has big financial implications. But so I think those are the three factors and now when we look at monetary policy, I'd like to get into what that means what the value of money is and how that affects the market. and the currencies that were

Figure 3.6: **Alg 1. Semantic Segmentation** Example of one speaker turn input into our coreference resolution algorithm. On the left, the coreference tags generated from the AllenNLP [183] coreference resolution [184] module are shown with six different references highlighted. Our algorithm groups sentences with references into semantic chunks with a minimum limit of references and words so that the semantic chunks are still meaningful. In Semantically Clustered Chunk #1 (blue), the first sentence includes three different references (1,2,3), of which two (2,3) are still used in the second sentence, hence why it is included. The singleton contains no references and is segmented out. In Semantically Clustered Chunk #2 (red), the first sentence contains two references (4,5) of which one (4) is in the second sentence. Although there is a new reference (6), the reference terminates within the second sentence so no further sentences are added.

same entity indicate the same concept is still being discussed. Sudden changes in speakers can be correlated with topic boundaries [186] and this concept serves as a common segmentation approach in NLP [125]. All iterative instances of the algorithm on each speaker turn segment S_i are therefore independent of each other.

Here we walk through a single speaker turn pass of Algorithm 1. Speaker turn S_i contains sentences s that were previously divided by ASR predicted punctuation. For each sentence s, we first obtain the coreferenced entity c_{min} (`line 13`) that maximally spans the current sentence s and future consecutive sentences $s \in S_i$ (`lines 4:10`), subject to constraints. This denotes the start and end pointers (e_0, e_f) of the current semantically similar chunk, **P** (herein referred to as cluster, `line 13`). Sentences contained by **P**'s span e_0, e_f, are assigned to **P** (`lines 19:20`). If sentence s contains another entity that spans further than **P**'s current end pointer, e_f is updated to the sequentially higher indexed sentence (`lines 21:23`). When **P**'s span is exhausted and cannot be further extended, the algorithm begins a new cluster **P**. Sentences that contain no entities are singleton clusters.

We restrict the entity span to at most $m = 100$ words and require each valid span to contain at least $p = 3$ mentions (coreferences).[4] We also do not consider "`I`" and "`me`" entity references since these references do not indicate a semantic change. Figure 3.6 demonstrates an instance of the sentence entity spans procedure.

Coreferenced Semantic Segmentation Effectiveness on *Long Summaries*

Model	Segmentation Strategy	Heuristic Score
`M3`	**Coreferenced Semantic**	**0.83**
`M1`	Naive Fixed Length	0.61
`M2`	Naive Fixed Length	0.70
`M3`	Naive Fixed Length	0.68

Table 3.5: Automatic evaluation heuristic scores for various segmentation strategies on *Long Summaries* compared to **`M3`** using Coreferenced Semantic Segmentation.

To determine if semantic segmentation is effectively grouping information for **`M3`**, we evaluate the core coreferenced semantic segmentation algorithm, Alg. 1 on *Long Summaries*. Specifically, we use our heuristic score to **only** evaluate *Long Summaries*, as subsequent (*Short, Medium Summaries*) evaluation using a previous longer summary as input would induce a circular dependency due to `Sentence Transformer`'s usage in Algorithm 2 and the heuristic itself. Because *Short, Medium Summaries* use `Sentence Transformer` to determine which input passage segments (that would ultimately become a circular reference transcript) to semantically include for summarization, their outputs would likely score artificially high due to the heuristic's intrinsic incorporation of `Sentence Transformer`.

We find that **`M3`** with semantic segmentation obtains a heuristic score of 0.83, a 0.13 improvement over the best naively segmented model, **`M2`**, suggesting Alg. 1 is effective and facilitates increased summarization performance. The marked improvement is important as mediocre initial summarization (*Long Summaries*) would lead to poor downstream hierarchical summaries (*Short,*

[4]We empirically observed that entities below these requirements had low relevance to the underlying concept.

Medium Summaries).

Summarization Model. We opt to reuse the paraphrasing `M3` instance of PEGASUS. The implementation of `M3` was taken from `huggingface.co`, using checkpoint `Tuner/007`. We also make the key observation that multiple recurrent forward passes of `M3` (independent of `Short, Medium, Long` heirarchical summarizations) removes speaker disfluencies and other speech artifacts of low importance.

The tradeoff for increased robustness towards speech noise and artifacts is also inherently found in `M3`'s paraphrasing nature; `M3` struggles to reason out semantically different ideas and suffers substantially from contextual errors (Table 3.3, `M3`). However, when ASR transcripts are preprocessed with Algorithm 1, our framework is able to generate not only cohesive and semantically logical summaries, but also achieve practical accuracy.

Hierarchical Concept Clustering and Merging. The next challenge is to determine which of the previous level's summaries to concatenate for further abstract summarization (Algorithm 2). Recall that semantically similar summaries must be joined or the model output can be factually incorrect (Table 3.3, `M3`) as abstractive summarization requires a high degree of semantic understanding of the underlying input passage. As a means to compare summary content similarity, we first use `Sentence Transformer` [173] to individually embed summaries (still contained in their own speaker turns **S**) $s \in S^n_{i=0}$, into vectors in a semantic space. By transforming the text segments into vector representations, we can now quantitatively compare their similarities via cosine distance. Summaries within each speaker turn S_i are then merged through usage of a pairwise cosine distance matrix for hierarchical (agglomerative) clustering. Merges are done within speaker turns to enforce a proximity constraint of only merging local summaries due to the long lengths of ASR transcripts.

Next, identified summary clusters are sequentially concatenated and concatenated summaries containing 5 or fewer words are stemmed. We observed that summaries which are not merged and contain few words are very frequently speech artifacts that contribute no value.

Note that the final level of *Medium* to *Short* summaries contain far fewer summaries than *ASR*

Transcript to *Long Summary* and *Long Summary* to *Medium Summary* due to previous merges. We remove the proximity constraint from *Medium* to *Short* and allow agglomerative clustering is across all summaries instead of within speaker turns.

Algorithm 2: Hierarchical Concept Clustering and Merging

```
Input: List of summaries (sentences) $s \in S_i \in \mathbf{S}$ in speaker turn $S_i$ for all speaker turns $\mathbf{S}$,
       Embedding Sentence Transformer Model $M$, $cut\_off = 5$ determining
       the smallest concatenated summary allowable
1 $L_{out} \leftarrow$ list();
2 for $S_i \in \mathbf{S}$ do
3     $\mathbf{E} \leftarrow M.embed(s \in S_i)$ # embed all summaries within current speaker turn;
4     $\mathbf{D} \leftarrow$ create_pairwise_cosine_distance($\mathbf{E}, \mathbf{E}$);
5     $labels \leftarrow$ agglomerative_clustering($\mathbf{D}$);
6     $cluster \leftarrow [s \in S_i].group\_concatenate(labels)$;
7     $cluster_{filtered} \leftarrow [s >$ cut_off $\in S_i]$;
8     $L_{out}.append([cluster_{filtered}])$;
9 return $L_{out}$
```

3.7 Hierarchical Summarization System Evaluation

We performed user studies to evaluate the following:

1. Human assessment of the quality of *Short Summaries*.
2. The system's ability to help users recover from errors in summaries.
3. The amount of time users saved by using our system when used in an unconstrained setting with their own browsing styles and comprehension goals.

Additionally, we present qualitative findings on how and when people would use the system to find interesting information in spoken dialog.

3.7.1 User Study Methodology

We recruited 10 recent university graduates from diverse professions (5 women, average age = 26) for our study. Each study lasted 1.2 to 2 hours and averaged 1.5 hours; subjects were paid

$20 per hour for their time. To begin, participants were provided with a scenario where a dialog summarization tool would be potentially useful: *imagine you get an email from a friend or colleague about an exciting interview on YouTube about "Diversity and Inclusion in the Workplace." It's 23 minutes long, you're not sure if you want to commit to watching the whole thing, but you want to know if there's anything new or interesting in it. We're trying to help people explore audio clips to find key takeaways.* They were also informed that the summaries were generated by an AI and might be imperfect.

Participants were then given a link to the interface with the audio for "Diversity and Inclusion" loaded in. During the warm-up, we explained the different UI components, *Short, Medium,* and *Long* levels of summarization, the original transcript, and the media button to play and scan the corresponding original audio section. Figure 3.1 is an example of what the user would see. To familiarize users with the system, we instructed them to read the first *Short Summary*, its corresponding *Medium Summary*, *Long Summary*, and ASR transcript segment, as well as to play the audio segment.

After the warm-up, participants were asked to perform three tasks:

1. `Short Summary Quality Assessment`: assess *Short Summary* quality for two audio files ("Diversity and Inclusion" and one of their choice). Here participants were asked to think aloud so we could understand how participants built an intuition and what their interpretation of the system was like. We asked participants to rate each *Short Summary* for two things: 1) grammatical correctness and 2) semantic comprehensibility. Users answered "yes" or "no" to each question. For semantic comprehensibility, we ask them whether they were able to understand the *Short Summary* and if it matched the corresponding audio segment's content. Users were able to check for semantic meaning by comparing against *Medium, Long Summaries*, original ASR transcripts, and audio.

2. `Short Summary Error Recovery`: if participants were confused on a *Short Summary*'s meaning, they were asked if 1) they could regain comprehension of the *Short Summary*'s meaning and 2) what system features they used to recover the meaning. Participants

were allowed to spend as much time as they needed to rate all the *Short Summaries* for the two audio files and were encouraged to think aloud as much as possible.

3. `Practical Usage Assessment`: use the system as they would in an every day situation. Participants were asked to choose the audio file that interested them most from the 5 remaining audio files in Table 3.1 and to use the system to find interesting information in that dialog quickly. They used the system without any restrictions and without thinking aloud. We timed participants' usage and observed their browsing strategies.

We concluded the study with a semi-structured interview about their experience using the system.

3.7.2 User System Evaluation

***Short Summary* Accuracy**. During the study participants rated a total of 556 *Short Summaries* from the test dataset (Table 3.1). The overall average accuracy across all users and recordings was 71.4% (see Table 3.6). The overall accuracy includes both grammatical correctness (84.9%) and semantic comprehensibility (75.9%). Overall accuracy is a measure of how many *Short Summaries* had any kind of error (either grammatical or semantic).

Criteria	Accuracy
Grammatical Correctness	84.9 ± 5.1%
Semantic Comprehensibility	75.9 ± 4.8%
Overall Accuracy	**71.4 ± 4.9%**

Table 3.6: Average *Short Summary* accuracy and standard deviations.

An overall accuracy of 71.4% means that many *Short Summaries* can be read and understood without any issues. The system's grammatical correctness is reasonably high (84.9%), but the system's semantic comprehensibility is lower (75.9%). Generally, grammatical errors are not detrimental to user experience because most grammatical errors do not distort the meaning of the sentence. However, poor semantics often requires users to investigate further to comprehend the meaning [187]. In this study, to fully comprehend every *Short Summary*, users would need to investigate semantic errors for 1 in 4 *Short Summaries*.

Because of the current state of machine summarization, we were not expecting the *Short Summaries* to be perfect. However, with 71% accuracy, we are encouraged that a usable and valuable system can be built in presence of errors. The subsequent evaluations are focused on whether the the system can help users achieve their goals despite these errors.

Short Summary Error Recovery When a participant encounters a confusing *Short Summary* during the `Short Summary Quality Assessment` task, we evaluate whether the user can recover regain holistic text comprehension by using the interface. In total, there were 140 *Short Summaries* with unclear meaning, and users recovered from **92.9**% of them. This recovery rate is high – the interface allows them to recover from all but **7.1**% of them. This indicates that although *Short Summaries* contain errors, the system can still allow users to have full comprehensions with some extra effort – the post-recovery success rate is nearly perfect: 98.2%.

Required Hierarchical Traversal Level	Fraction (%) Used
Medium Summary	11.5%
Long Summary	19.2%
ASR Transcript Segment	40.8%
Audio Segment	20.8%
Querying Neighboring Segments	6.9%

Table 3.7: Distribution of the hierarchical levels users explored in order to recover from an inaccurate *Short Summary*.

The hierarchical summarization features of the interface were designed to help users recover from errors quickly and easily. We wanted to know to what degree participants used these features during recovery. We found that participants used all the hierarchical summarization features to some degree. Participants had two main styles of using the summarization: either they traversed down the hierarchy in order (from most summarized to least summarized forms of the semantic chunk or skipped to their preferred source of information. Table 3.7 shows the breakdown of how often each feature was used. Participants used *Medium Summaries* a small amount (in 11.5% of recoveries) and used *Long Summaries* more (in nearly 20% of recoveries). However, participants used the ASR Transcript Segment the most (for 40.8% of recoveries.) This behavior is explained by users noting how *Medium, Long Summaries* were either too similar or contained the same semantic errors as *Short Summaries*. As a result, users defaulted to reading the ASR Transcript Segment more often.

There are some instances where the transcript and summaries are insufficient for error recovery. In 20.8% of recoveries, users chose to go back to the audio segment. Although audio takes more time to listen to, the audio contains information that the transcript does not - it contains emphasis and tone of voice (as well as avoiding any errors in the transcript). In a small number of cases (6.9%) users chose to read neighboring segments to recover from an error. This is almost always because users needed more context that lay outside of the current semantic chunk to recover full comprehension.

Lastly, some participants noted that some of these errors may not be possible to recover from because the underlying audio content was difficult to understand or incoherent.

Time Savings. In the `Practical Usage Assessment`, when participants used the system at their own pace (without thinking aloud or rating tasks), we found that on average, the time participants spent was **27.1**% of the original audio time to reach a level of understanding that they were satisfied with (Table 3.8). The fastest user spent only 6.9% of the original audio time, and the slowest spent 75.9% of the original audio time, but most spent between 20% and 30% of the original audio time. We believe the system presents a sizable time savings. Although users do have strategies for processing audio faster (such as listening at 1.25x speed or skipping the first minute of the audio), these strategies are unlikely to provide dramatic speed ups and often lack the control and freedom that our system provides.

How much time users spent varied with what level of detail they wanted to understand. The two users with the highest time percentage (75.9% and 45.5%) were looking for details. The other users took much less time and wanted a cursory understanding of the material (Table 3.8) such as getting a broad gist of the conversation, wanting a few key takeways, or wanting only a specific piece of information.

Participant	Self-Declared Style Browsing	% of Audio Time
P_1	Detailed	75.9%
P_2	Cursory	12.1%
P_3	Cursory	14.9%
P_4	Cursory	20.7%
P_5	Cursory	20.8%
P_6	Cursory	6.9%
P_7	Cursory	25.9%
P_8	Moderately Detailed	24.7%
P_9	Cursory	24.3%
P_{10}	Moderately Detailed	45.5%
$P_{average}$	-	27.2%

Table 3.8: Individual participant times using the system as percentage of original audio length alongside intended browsing style.

3.7.3 User Qualitative Evaluation

. During the semi-structured exit interview, participants were asked about their experience using the system, use cases where they would or would not consider using it in their life, and potential improvements to the system.

While browsing for interesting nuggets of information, users sometimes found the *Short Summary* sufficient, but often leveraged hierarchical summarization features to dig further. In the podcast discussing Teach For America during Covid, P_7 found this *Short Summary* to be interesting on its own: "15 to 16 million children don't have access to broadband internet." Likewise, in the podcast on Health, P_8 found this *Short Summary* interesting without reading further: "You can prevent systemic bias by hiring nurses who speak Spanish and are bilingual." However, P_2 selected Ray Dalio's interview and found the first *Short Summary* intriguing ("There are three big forces at work have have not existed before") but had to read the transcript to discover what the three forces were. Similarly, P_9 selected Bill Ackman's interview and was intrigued by the *Short Summary*: "The uncertainty of the future can affect the model that analysts use to value securities." P_9 said: "I found this as a thesis statement and read more into it." Although *Short Summaries* may be sufficient, that is not always the case and as a result it is critical that *Short Summaries* provide good information scent to indicate to users when to use hierarchical features to investigate further.

Users reported they would consider using a tool like this for media they considered "condensible." They mentioned news, YouTube videos (particularly reviews), interviews, and podcasts as media that could be condensed. Users also stated they would prefer to use the tool for topics they were curious about, but "didn't want to spend too much time on" (P_7). One such use case was if a friend suggested they listen to a long audio file and they "don't wanna be rude" (P_7). Another use case presented was for situations when a given topic is familiar, but the presentation could give background that could be condensed (P_2). Finally, a third case users noted is when they wanted specific information from an audio recording, such as "learning what a company does in interviews with CEOs" (P_{10}). 6 out of 10 participants said they would not use the tool for detailed or technical topics, particularly if they were responsible for learning the material at work or school. Additionally, they would not use it for personal things they were deeply interested in because "I'd want to read those in detail" (P_2) or for fictional narratives where there is pleasure in enjoying the flow of the story rather than extracting information. Clearly, this is not a tool for all use cases. Similar to how people wish to skim text using a tool, our system can allow users to "skim" audio.

Throughout the experiment users frequently relied on their own knowledge to guide them towards exploring content in more detail. P_5 works in the medical field and was intrigued by a *Short Summary* in the Health dialog: "There was a spike in demand for specific types of nurses." P_5 wanted to know what those specific types were. The *Medium Summary* did not contain that information but the *Long Summary* did - ICU nurses and ED (emergency department) nurses were the two specialties named. P_{10} was familiar with "How I Built This" podcast and was specifically interesting in understanding the "pain points with building the company." He spent most of his time in the middle of the interview because he knew from the structure of the podcast that the information would probably be there. P_4 was already familiar with finance and with Bill Ackman's philosophies, but the tool was useful to him as he skimmed the *Short Summary* to see if there would be anything new and interesting, given his background. For users with background knowledge, tools which provides user control and freedom enable more efficient navigation in order to locate valuable information.

The hierarchical features were more useful for some dialog than for others. 8 of 10 users of the Diversity dialog did not mention reading *Medium, Long Summaries*, and always went to the transcript. However, 5 of 5 readers of Ray Dalio dialog used the *Medium Summaries*. This is likely due to the nature of the underlying audio and the quality of the summaries. Ray Dalio tends to speak in structured and organized paragraphs creating longer but structured ASR transcript segments. This created enough of a distinction between *Short* and *Medium Summaries* where *Medium Summaries* contained a good balance of interesting information while retaining an attractive length. Meanwhile, the *Medium, Long Summaries* of the Diversity dialog did not add enough information causing users to read ASR transcript segments to obtain desired additional details. As these factors are difficult to control for and user dependent, the solution of presenting summaries at multiple levels of granularity was successful.

Short Summaries were imperfect, but users found strategies to recover understanding of the underlying material. A common complaint about the tool was use of ambiguous pronouns in the *Short Summaries*. For example, in the Chipotle interview the *Short Summary* says. "It's hard to say what he is like because he was an amazing visionary." Here "he" refers to the CEO's mentor - a head chef at a famous restaurant, but users had to read the transcript to find this information. A related complaint is that "the short summaries were disconnected from each other" (P_7). Summarization often removes segues and other transitional elements in order to surface meaning. However, this provides a disjointed experience for users and requires them to "rewind a little bit" (P_2) to recover context or flow. When using the hierarchical summarization features to recover users found "Word per word highlighting indicates where it was to quickly see the segment to read to resolve" (P_{10}). This type of consistency across the interface makes information easier to scan. In future work, we could explore ways to make the *Short Summaries* flow better, such as resolving pronouns and linking related information across the *Short Summaries*.

3.8 Limitations and Future Work Direction

Users generally found the system useful, though there are several ways the underlying technology could be improved to provide better summaries and to generalize to more types of dialog.

3.8.1 ASR Limitations

Although ASR works well for the two-person, studio-recorded interviews in this system, it still has many limitations. ASR may incorrectly transcribe audio with speakers with accents or in the presence of background noise, limiting user and context settings. Additionally, ASR makes diarization errors when multiple speakers are present and interrupt one another, such as in a panel discussion where participants argue or get excited. These diarization errors in turn limit the types of conversations ASR works for.

3.8.2 Summarization Language Model Limitations

User Information Retention. The goal of this system is to help users navigate longform audio and find interesting information quickly which creates user comprehension trade-offs. A more difficult challenge would be to make a system that helps users find *all* interesting information. We did not measure the precision of the system, but acknowledge that users missed interesting information. In particular, we observe the summarization language model occasionally omits named entities that can signal interest to readers. For example, in the interview with hedge fund manager Bill Ackman, he mentions he plays tennis. The system correctly summarizes this fact and several study participants found it interesting. The system then produced the *Short Summary* saying "Ackman: He spent the rest of the match trying to hit me back after I hit him with the overhead." Many participants overlooked this. However, the person Bill Ackman mentions hitting is tennis legend John McEnroe. It is likely users missed this potentially interesting fact due to summarization omission. Future work could experiment with replacing references with corresponding named entities in *Short Summaries* to provide better information scent for users. A first step would be to quantify the interesting information that users missed and then identifying the source: ASR system level errors (transcription, diarization), segmentation errors, or language model summarization errors. Understanding where errors stem from can inform subsequent research where the system should be improved upon to provide more complete information.

Subtle System Errors May Create User Misunderstanding. In addition to overlooking information, users may be misled by the system if output summaries contain errors that users do not detect. Typically errors are often obvious due to the surrounding context or the reader's prior knowledge of the topic. However, subtle non-obvious errors may go unnoticed, obscuring system inaccuarcies to users. This can lead to misunderstandings as users may not realize a correction is needed to properly understand the underlying information. Our evaluation did not address this limitation. An important next step is to study whether or not the summarization system contains these subtle errors, and if so, how frequent they are. Depending on the nature and severity of such errors, systems could contextually reason entire dialog transcripts (as opposed to individual local semantic segment) or query outside information to check summary adequacy.

External Knowledge (Co-)Referencing. Speech summarization is particularly challenging when speakers reference ambiguous objects without proper introduction. Such instances include referencing world knowledge (i.e. current events and facts), local knowledge (speaker dependent context), and deictic references (entity a speaker is directly pointing at). In our study recordings, speakers often referenced world knowledge. For example, in the Diversity and Inclusion audio, our system produced the *Short Summary*: "The two children I have look like dante[sic] and rashad." Some users picked up that this referred to Daunte Wright and Rashad Turner, recent tragedies that were foundational to the Black Lives Matters movement, while other users missed this fact. In future work, it may benefit readers to link indirect references in the summaries to outside knowledge.

3.9 Conclusion

Longform spoken dialog is a rich source of information that people encounter every day. However, for individuals who lack the time or inclination to listen to complete audio, the information is not easily accessible. Given the marked practical performance improvement in ASR and large scale summarization language models, new opportunities and affordances exist that were previously unexplored. These possibilities give hope that content rich longform spoken dialog could be easier to access for individuals who lack the time, inclination, or ability to listen to complete audios. However, current state of the art ASR and summarization models still present several sources of error, ranging from word recognition failures during speech recognition to the introduction of hallucinations, grammatical errors, and misrepresented contexts during summarization. Such errors ultimately suggest that immediate fully automated approaches are still out of reach and careful consideration must be given towards creating systems and algorithms to compensate for such shortcomings. This chapter presents an end-to-end dialog summarization system that efficiently repurposes existing ASR and language models while mitigating their innate flaws, and presents an accompanying user interface that allows for user controllable consumption of hierarchical information.

While prior work has established the benefits of hierarchical browsing of information [171, 38], we demonstrate that automatic summarization is now feasible through large scale language models. Moreover, hierarchical browsing can help people skim information and recover from errors made by automatic tools. Interestingly, we also observe that users can apply *Short Summaries* as a navigational tool to identify interesting parts of audio recordings and drill deeper. From a technical perspective, our system achieves practical summarization accuracy of spoken dialog without custom data sets [167] or subsequent retraining of large language models. Rather, by using an improved segmentation approach we were able to employ out of the box industry standard models to produce readable and meaningful summaries in a novel traversable interface.

Chapter 4: Creating a Hierarchical Summarization Dataset for Longform Spoken Dialog

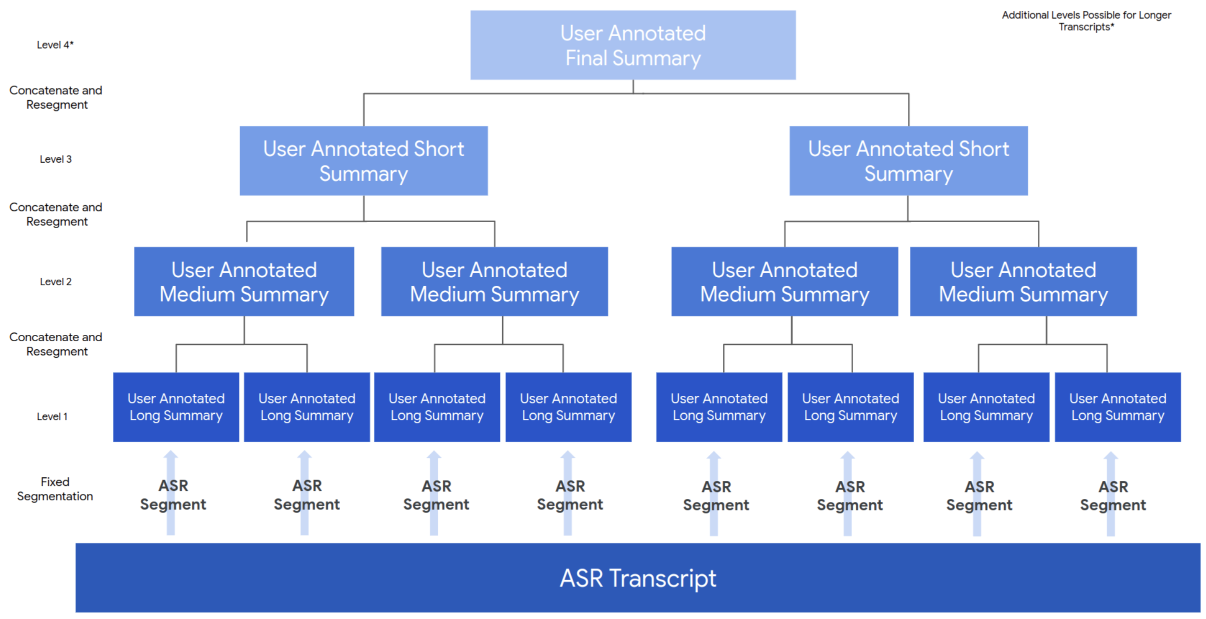

Additional Levels Possible for Longer Transcripts*
Level 4*
User Annotated Final Summary
Concatenate and Resegment
Level 3
User Annotated Short Summary
User Annotated Short Summary
Concatenate and Resegment
Level 2
User Annotated Medium Summary
User Annotated Medium Summary
User Annotated Medium Summary
User Annotated Medium Summary
Concatenate and Resegment
Level 1
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
User Annotated Long Summary
Fixed Segmentation
ASR Segment
ASR Segment
ASR Segment
ASR Segment
ASR Segment
ASR Segment
ASR Segment
ASR Segment
ASR Transcript

Figure 4.1: **Hierarchical Annotation Procedure**. We propose a bottom up hierarchical annotation protocol that dynamically segments inputs presented in a dedicated summarization annotation user interface (UI). Annotators recursively summarize text in recursive a manner when harvesting gold label human written summaries. This allows for a number of benefits: (i) intermediate text representations of summaries in varying levels of detail (ii) information of gradual content selection in summaries (iii) system guided and constrained summarization. For conventions, levels can be interpreted as depth (y-axis) or the longitudinal plane, whereas the lateral intricacies (ASR segments, individual summaries) within a level can be interpreted as the x-axis of the longitudinal plane.

In this chapter we explain the process of curating a hierarchically annotated dataset for long-form spoken dialog. As with the previous proof-of-concept system discussed in Chapter 3, we do not currently have a human annotated, gold label dataset that contains intermediate text level summaries to automatically benchmark the system against. Figure 4.2 gives an visual overview of the dataset collection process. We point out that this dataset affords many interesting research questions that are outside the breadth of this dissertation[1]. The same conventions defined in Chapter 2.1 are also used here.

4.1 Motivation

The natural language task of summarization (refer to Section 2.5), where the goal is to compress a long input text by identifying and extracting the most important parts and output them as a shorter text, provides an instinctive solution for refining speech. After using automatic speech recognition (ASR) to transcribe audio into text, lengthy audio transcripts can be automatically processed and distilled into their most important components. In doing so, longform spoken dialog can be transformed into a more manageable and viable form. Note that a transcript (obtained through state of the art Google Speech-to-Text [53]) will be the starting input.

While applying automatic summarization language models (LM) to speech in of itself is not a new concept, abstractive longform spoken dialog summarization work is still relatively nascent. Broadly, current research in abstractive speech summarization can be broken down into two categories: model architecture improvements and creating speech summarization datasets (Refer to Section 4.3). However, current research do not adequately address challenges stemming from longform speech summarization.

Challenges arising from longform text inputs. Length constraints occur when the input size of a summarization model exceeds the typical allotted 1024 tokens, though some language models may have larger maximum inputs. This is particularly problematic in the spoken dialog domain where meetings, conversations, and interviews can span hours, for reference a 20 minute speech audio file may contain anywhere from 4,000 to 6,000 words[2], far exceeding a typical transformer based language model token input limit. Speech and dialog transcript lengths easily exceed the input size limits of language models and make encoding them in a single forward pass intractable, forcing the input to be broken up. This leads to *context fragmentation*, where the input has informa-

[1]Further experiments on validating data quality and training baselines will appear in a separate work. The gold summaries are currently only used in the automatic evaluation of System 2 in Chapter 5.

[2]Accounting for SPM (sentence piece word tokenization) and subwords there is likely a 1.3 multiplier on words to number of tokens. For example, `running` can be broken down into `run` + `ning` depending on training.

tion that is predicated on context that is no longer part of the input, leaving inadequate information for a language model to process.

Language models already require a higher degree of complexity to successfully reason through an input text's content in order to successfully determine what information should be conveyed in the generated abstractive summary (herein referred to as *content selection* [145, 146]). *In a longform input setting, this is even more challenging.* Given the substantially larger information load, far more content must be reasoned through. Moreover, summarizing such a long (20 minute or more) audio file in a few sentence is likely insufficient; standard length summaries are too short to capture sufficient amount of information.

4.2 High Level Overview

Proposed Dataset Solution. From the previously enumerated challenges, we can observe how curating a dataset that is *hierarchical*, *recursive*, and *modular* is immensely beneficial to longform[3] abstractive summarization research.

1. *Hierarchical*. By providing intermediate summaries in increasingly levels of brevity and particularity, users are afforded the option of gold summary references at multiple stages. This provides training or test data for summarization models adapted to output various degrees of length and detail.

2. *Recursive*. Bottom-up [44] recursive summarization gives insight towards content selection by providing a progressive account as to what details are deemed insignificant and summarized out as well as what concepts are important to retain and/or merge with other concepts.

3. *Modular*. Accounting for long range contextual dependencies and segmenting longform input into independent smaller chunks, the task of content selection is far easier. The independent smaller inputs provide a means for size memory constrained language models to

[3]We primarily focus on the challenges that arise from longform inputs to summarization models.

process longform dialog. Importantly, this also provides transparency to high level summaries and their respective summarized contents; it is possible to obtain an alignment from any subsequent (recursive) summary back to the input transcript.

High Level Annotation Procedure Description. Therefore, we propose creating a hierarchical abstractive summarization dataset. With a longform transcript as the source input, the hierarchical recursive annotation procedure, also shown in Figure 4.2, is as follows:

1. The protocol breaks the input text into smaller manageable *segments*, reducing the complexity of annotator content selection by narrowing the scope of input information.
2. Annotators are then tasked with individually summarizing each segment.
3. All the user provided summaries are collected, finishing the current level. For the next level, all the summaries are concatenated and dynamically re-segmented and the annotator repeats step 2. Observe how the text is compressed at each level by the summary compression rate.
4. When the concatenated text is short enough (stop condition), the text is simply summarized into a *short* 5-7 sentence (300 word) summary.

Our contributions primarily lie in creating a novel hierarchical summarization annotation procedure and the resulting dataset, which to the best of our knowledge, is the first of its kind and not limited to only the spoken dialog domain. This is presented along with a dedicated annotation user interface (UI)[4]. We also discuss special considerations towards biases that arise from *content selection* and *context fragmentation* in the annotation process.

4.3 Related Work

For more detail on the related work below, refer to Chapter 2.5 - 2.7.

Hierarchical Recursive Summarization. Two prominent strategies for processing extremely long form input sequences are used: (i) designing new hierarchical deep learning architectures to

[4]`https://resubstrate.github.io/lex-client/index.html#/login`

accommodate greater context [71, 72, 78] and (ii) modifying input sequences to fit with existing LMs (i.e. reducing the input length). We will primarily focus on the latter strategy, referred to as hierarchical or recursive summarization, which our annotation procedure closely mirrors. It is important to understand the difference and make a distinction between a (i) *model level* hierarchical architectural modification and (ii) *text level* hierarchical processing methodology.

(i) To model long range dependencies in the meeting setting, models typically employ a form of hierarchical attention architecture [121, 122, 123, 124, 107]. Specifically for meetings and speech, HMNet (Hierarchical Meeting summarization Network) [125] creates a two-level hierarchical structure with a speaker turn level transformer and word level transformer.

(ii) Hierarchical recursive techniques [112, 113, 114, 116] are generally concerned with segmenting a longform input into smaller and more tractable inputs and recursively using a summarization model's output as its input to obtain increasingly shorter summaries.

It is important to recognize [188] and its relationship to this dissertation; the procedures and challenges (recursive summarization, hierarchical clustering) the authors encountered are widely similar to this chapter, except perceived from a human annotator's perspective. However, [188] is concerned with multi-document summarization (MDS) and does not provide a dataset containing gold-label human written intermediate hierarchical summaries. As a result, [188] is evaluated with a series of subjective questions. In this chapter, we provide such intermediate annotations, thereby allowing automatic evaluation for hierarchical summarization systems.

4.4 Data Preliminaries

All audio recordings consist 1, 2, or 3 speakers engaging in a dialog about different topics ranging from business to geopolitics to social sciences and range from 25 minutes to 1.5 hours. All interviews are conducted in English, although not all participants are native English speakers. We prepared a dataset of 25 transcripts (Table 4.1), of which, 10 are completely human hierarchically annotated. The data consists of audio recordings from a diverse range of speakers and topics from:

Transcript Source	Transcripts	Minutes	Word Count	Total Hierarchically Annotated
Bloomberg Wealth	6	155	27,801	2
NPR: How I Built This	14	674	122,807	7
TED Talks	5	147	23,097	1
Total	25	976	128,685	10

Table 4.1: Audio file metadata. These are typical podcasts, interviews, and debates. *Only a subset of these files are used in the evaluation of Chapter 5.* Currently 10 (proportionally from each data source) of the transcripts are fully annotated and used in evaluation in Chapter 5; annotations of the remaining transcripts is ongoing and to appear in this chapter's corresponding full paper.

1. NPR: Podcasts from the "How I Built This" series

2. Bloomberg TV: interviews predominately on finance

3. Ted Talks: topics range in discussing COVID-19 to Diversity and Inclusion

The audio was transcribed with Google Speech-to-Text. Additionally, we use this service's speaker diarization and automatic punctuation to obtain speaker turns and sentence endpointers respectively. Word recognition errors and erroneous punctuation are not corrected, accurately depicting an archetype input transcript for later test or training use. As such, gold label *transcripts* do not exist in this dataset. *Gold label* herein refers to the human authored summaries.

4.5 Methodology

Reiterating conventions, levels can be interpreted as depth (y-axis) or the longitudinal plane, whereas the lateral intricacies (ASR segments, individual summaries) within a level can be interpreted as the x-axis of the longitudinal plane. A segment-summary pair refers to an individual input segment and its corresponding annotated summary.

Each segment is obtained using a fixed segmentation procedure which truncates at a maximum length (500 words) while preserving sentence boundaries. In other words, if only part of a sentence can fit in a given segment without exceeding the maximum allowable length, the sentence is removed from the segment and used as the beginning of another segment. Whenever segmentation is discussed in our system, it is referring to this fixed segmentation procedure. Addressing context fragmentation is discussed in Section 4.5.3 and 4.5.5.

The summarization process begins at the ASR transcript, or level 0. Levels refer to the actual text and not the concatenation and segmentation process. For example, annotators are at level 1 when summarizing the segmented ASR transcript, and at level 2 when summarizing level 1's summaries. For simplicity, when referring to a level's input, we assume it is already segmented unless stated otherwise.

The *n*-th level denotes the final summary and the input is short enough where it no longer requires segmentation. This is triggered by a termination condition when the number of words in the input falls below a fixed threshold. Lastly, observe how the compression rate of a level (the fraction of total number of words in all the level's annotated summaries divided by the number of words in the input) affects the number of levels. A smaller compression rate will result in reaching the termination condition quicker, resulting in potentially fewer levels.

4.5.1 Annotator Design Considerations

Next we examine challenges due to speech and annotator biases and their implications. It is necessary to suppress and minimize such complications at any early stage; due to the hierarchical recursive aspect, initial errors can propagate and become exponentially detrimental towards later recursive summaries. Consider the case where an an annotator writes an erroneous summary. This erroneous summary is now a part of the next level's input, likely guaranteeing the output will again contain some amount of the erroneous input[5]. Interestingly, some of the challenges annotators need to contend with. are similarly reflected in challenges that summarization models also face.

Speech Transcript Comprehensibility. Reading a speech transcript is quite challenging: aside from imperfect punctuation, word recognition errors, and vague pronoun references, it can be difficult for a person to track and follow the conversational discourse. Consider the following

[5]It is possible for users to "summarize out" the erroneous input summary, for example if the erroneous input summary contains no useful information. However, more likely than not, some amount of information will be propagated and this will continue recursively bias future levels.

examples 1 and 2:

1.
   - Speaker 1: `"There was a sale at the store.".`
   - Speaker 2: `"Yeah I bought apples and then returned home."`
2.
   - `"There was a sale at the store.  Yeah I bought apples and then returned home.".`

Normally, listening to a conversation, it is easy to distinguish who is talking and thus accurately comprehend the situation as in example 1. However, reading a transcript without any indication of who is speaking, can contribute to materially different understandings of the same situation. As with example 2, it is unclear if the same person bought the apples and returned home or if it was someone else. Note that there may be occasional clues when reading a transcript such as an interjection or the obvious referring to someone by name. Appropriately, it is important for the system to convey speaker turns to annotators.

Annotator Content Bias. A "good" summary is inherently subjective; in other words, what one individual finds interesting and worthwhile in a summary may differ from another individual. Given an input passage (or transcript) there conceivably countless summary possibilities that may result in a "good" quality summary. As a result, human written summaries may conceptually differ due to annotator content selection biases (i.e. What does this annotator deem to be important vs. another annotator?) and still remain adequate and of high quality. The importance of content selection is further magnified when summarizing longform text. It becomes clear how summarization is a fundamentally *under-constrained* problem [90]. These effects are further magnified when summarizing longform text. Still, it is worthwhile to point out that breaking up longform input and recursively summarizing enforces a shared tiered structure between annotators, and should yield more consistent results summaries [43].

Accordingly, we ensure at least 2 annotators summarize any given transcript. Using more than 1 reference summaries for evaluation (and possibly training) dives into the multi-reference domain

[189, 190, 191] and outside the scope of this dissertation[6].

Annotator Writing Style. Second, different annotators have different prose tendencies. Some individuals may write with an active voice while others write with a passive voice, meanwhile vocabulary and vernacular is also highly dependent. In order to minimize the variance of summaries, annotators were instructed to write with a consistent tone of voice, avoid ambiguous pronouns if possible, incorporate a mix of extractive summarization, and try to not introduce any proper nouns in the summary if they were not present in the input segment. The last instruction is particularly important to prevent memorized or *hallucinated* entities if these summaries are to be used as training data [69].

Annotator Summary Lengths. Critically, the type of content in the input segment directly impacts the length of an annotator's summary; after all segments will vary as some can be more information dense and others can be non-informative. While having approximately summary lengths is instrumental towards consistency in intermediate summaries, a fixed length summary cannot be forced upon users without sacrificing summary content quality. Observe how the distribution of informative content is not uniform in an input text, suggesting that some segments can have shorter optimal summaries while others require more text to achieve adequacy.

An annotator's writing style will also decidedly vary. By default, an annotator that is more verbose will have a higher average word count than an annotator that is terse resulting in different summary lengths. From here, we take away the observation that the system needs to be flexible in accommodating different summary lengths, while maintaining a depth uniformity.

4.5.2 Hierarchical Summary and Level Design Considerations

Collecting longform human annotated abstract summary datasets is already a challenging task with regards to differences in annotator summaries [192]. With the hierarchical recursive nature, this is even more difficult as there are new elements to consider. For the same input transcript, hierarchical summarizations from different annotators need to have the same number of interme-

[6]This will be addressed in this chapter's corresponding paper (future work).

diate levels if corresponding intermediate summaries (i.e. level 1 from annotator 1 and level 1 from annotator 2) are to be used as multiple references. Summary levels between annotators are only comparable if they are aligned. Note that the final n-th summary will likely be comparable in length between annotators due to the termination condition catching the n-th layer's input and coercing it to be similar lengths regardless of annotator.

Ensuring consistent levels between annotators is deceptively complex. Naively, the system can use a fixed rule pairwise concatenating summaries and recursively summarizing pairs, resulting in the same number of inputs and levels regardless of annotator. However, this would tremendously sacrifice quality and runs counter to the requirements set forth from Section 4.5.1 on annotator summary lengths.

Balancing concerns of flexible inputs and summaries motivates the choice of a concatenation and re-segmentation procedure [112, 113]. By concatenating and re-segmenting to the maximum length allowed per segment, level j's summaries (outputs) give level $j + 1$ inputs a wider context (receptive field). Now the system's backend design question becomes, how can the system ensure consistent hierarchical levels without fixed rules that constrain annotation quality?

The answer lies in adjusting the compression rate between levels where we have the invariant: a smaller compression rate leads to fewer levels while a larger compression rate produces additional levels. Laterally, compression for a level is commutative; it is equal to the average compression rate of all the individual segments. This reduces the problem to adjusting individual segment compression rates.

There are two key observations that need to be understood:

1. *Individual Input Segment Length.* As Goldestein et al. [193] empirically notes: *"summary length [is] independent of document length, and that compression ratios became smaller with longer documents"*[7]. Rephrased, a segment's annotated summary output length has a nontrivial degree of inelasticity, or stays relatively the same, with respect to changing an input passage's length. Intuitively, it is difficult for a short(er) passage to be further sum-

[7]See Figure 1 [193].

marized without sacrificing quality; meanwhile, a long(er) passage's summary loses utility once it becomes too long. If the system can alter the segment lengths, it can govern a level's compression rate.

2. *Number of Inputs Segments Per Level.* Of course, changing segment lengths requires modifying the total number of segments. By increasing the number of segments, the length of a segment is reduced while decreasing the number of segments increases the length of a segment.

As a result, the system these are the two knobs that can be adjusted automatically and dynamically to tailor towards the individual annotator (Section 4.5.4). From a high level, the system sets an ideal compression rate (0.45) per level. If annotators deviate from it, the system adjusts subsequent levels' segmentation lengths and segment counts to guide the annotator back on track to the ideal compression. Intuitively, if an annotator tends to write less, writing summaries that have an average compression of 0.3, they can expect segments shorter in length but more numerous in quantity. Conversely, if an annotator tends to be verbose, they can expect fewer but longer segments.

4.5.3 Annotation Protocol Technical Details

In this section we explain the technical details in the hierarchical annotation process, shown in Figure 4.2. First we are given an input source text (ASR transcript) which contains n sentences, denoted as $S^0 = (s_1, ..., s_n)$. The segmented instance $\mathbf{S}^0$ is defined as $S_i^0 \in \mathbf{S}^0$. The superscript j indicates the current hierarchical level (e.g. S^j).

The intermediate summary outputs from each level is given as $T_i^j \in \mathbf{T}^j$. The concatenated summary is given by $T^{j,c}$; the superscript c (for concatenated) is indicates the individual summary outputs have been concatenated. Outputs can be used as individual segment-summary pairs $(\mathbf{S}^j, \mathbf{T}^j)$ for $j \in |levels|$ for training and test data. Additionally, inputs and outputs can be used at the document level where all segment-summary pairs are respectively concatenated in order $(S^j, T^{j,c})$.

Procedure Walk-through. For simplicity, we drop the j superscript notation denoting level. The input source text $S = (s_1, ..., s_n)$ is first segmented into a smaller, more manageable inputs: $\mathbf{S} = [(s_1, ..., s_i), ..., (s_j, ..., s_k), ..., (s_l, ..., s_n)]$ where $1 < i < j < k < l < n$. Observe how S now becomes $\mathbf{S}$ as it is a collection of segments of sentences, such as $S_i = (s_j, ..., s_k)$, which is now an individual input to a summarization model.

For each $S_i \in \mathbf{S}$ the annotator writes a summary $T_i \in \mathbf{T}$ creating segment-summary pairs. Here, the annotator has the option of including additional sentences as context for an input segment S_i from preceding ($S_{0<i}$) and succeeding ($S_{i<n}$) segments. An instance of S_i is made where the additional context is prepended or appended accordingly ($S_i^{'}$) and set aside for final individual segment-summary pairs that now have all the required context[8]. $S_i^{'}$ is not used for future levels. This concludes the process for one level and is repeated identically until a termination condition is reached (Section 4.5.4).

4.5.4 Controlling Segmentation

In order to facilitate relatively uniform depth across hierarchical summarization annotations, annotators were loosely guided by a dynamically calculated segmentation boundary algorithm. This algorithm began with the follow static assumptions:

1. A maximum segment word length of 500 to comfortably fit within the maximum length limitations of current state of the art summarization models (1024 tokens) [108]. Recall that the sentence piece model tokenization will yield more tokens than words.

2. A preset ideal compression ratio of 0.45 based upon past research investigating empirical optimal summarization compression [193, 194].

Based on these assumptions, an optimal segment length L_c is determined at every summary depth for an annotator's current summarization. The deviation from this optimal segment length is

[8]Because of the possibility of adding context, the maximum length per each segment was given additional tolerance to fit within a transformer model.

then calculated and the maximum segmentation length of the annotator's subsequent depth is adjusted to offset this deviation. By adjusting the maximum segmentation length at each depth (level) based on each individual annotator's compression rate, the overall depth across all annotations can be guided to a fairly convergent and uniform depth for each given source transcript.

We define the following:

- L_m = Maximum Segment Length (500 words)
- L_r = Ideal Compression Rate/Ratio (0.45)
- L_c = Ideal Segment Length
- L_s = Ideal Segment Count
- R = Realized Compression Ratio (of current annotation level)
- S = Realized Segment Count (of current annotation level)
- S_t = Segment Count (of current source text)
- W_s = Word Count (of current source text)
- O = New segment length (in words, not tokens)

$$L_c = W_s L_r^{\log_{L_r} S_t^{-1}} \tag{4.1}$$

$$L_s = \frac{R \cdot W_s}{L_c} \tag{4.2}$$

$$O = \frac{R \cdot W_s}{L_s} \tag{4.3}$$

Finally, at every depth annotations are validated for a terminal criteria which would conclude the hierarchy of summarizations based on either crossing a minimum word count or minimum segment count threshold. For this dataset a terminal threshold of less than 700 words or less than 10 segments was utilized. The termination criteria is given below:

$$W_s < 700 \quad or \quad \frac{W_s}{O} > 10 \tag{4.4}$$

4.5.5 Annotation User Interface

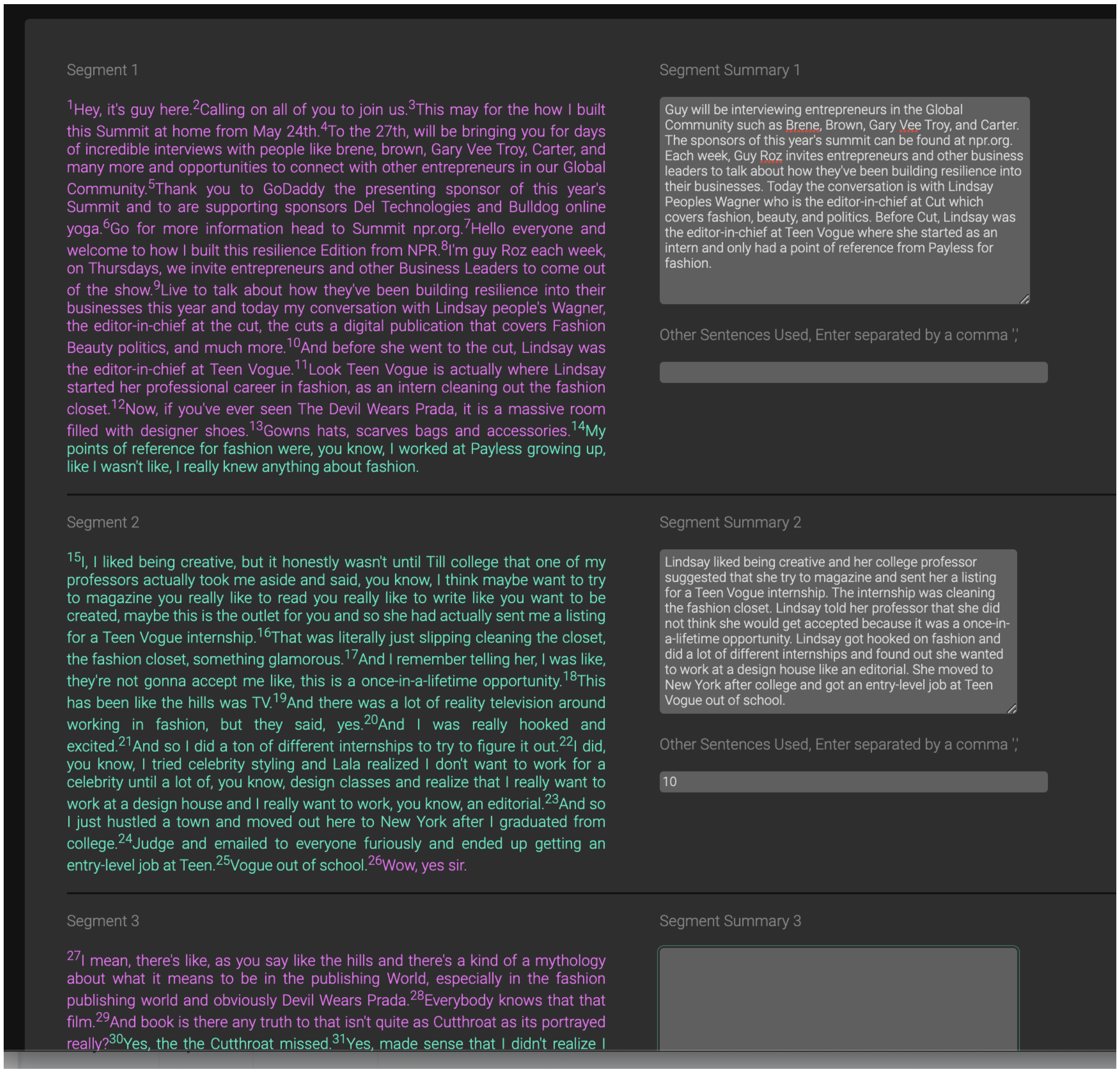
The annotation user interface is given in Figure 4.2 for the first level, the segmented ASR transcript. On the left hand side, users are given the segmented text input where each of the sentences are numbered. On the right hand side, users are provided a text box for writing their summary and a smaller box to include external context. At the bottom of the page, there is a submit button for the user to click when all the segments are complete, which finishes the current level. Subsequent UI views differ as they no longer contain speaker diarization coloring. The final level's UI view only contains one segment as the previous level's concatenated summaries are not segmented.

Notably, the UI has the following aspects:

1. *Speaker Diarization.* Speaker turns are color coded pink and teal to help users follow the conversation.

2. *Context Fragmentation.* Due to the fixed length segmentation, it is possible to truncate important information that is necessary to summarize a segment. As a result the user has the option to bring in sentences from preceding and subsequent segments for the segment-summary pair. Now each segment-summary pairs have the necessary context to be used individually.

3. *Text Box.* Annotators have the option to include no text as a means of explicitly excluding all the information in the input segment from the subsequent level. This was intentionally allowed as users may feel a segment contains neither relevant nor important information worth summarizing.



Segment 1

^{1}Hey, it's guy here.^{2}Calling on all of you to join us.^{3}This may for the how I built this Summit at home from May 24th.^{4}To the 27th, will be bringing you for days of incredible interviews with people like brene, brown, Gary Vee Troy, Carter, and many more and opportunities to connect with other entrepreneurs in our Global Community.^{5}Thank you to GoDaddy the presenting sponsor of this year's Summit and to are supporting sponsors Del Technologies and Bulldog online yoga.^{6}Go for more information head to Summit npr.org.^{7}Hello everyone and welcome to how I built this resilience Edition from NPR.^{8}I'm guy Roz each week, on Thursdays, we invite entrepreneurs and other Business Leaders to come out of the show.^{9}Live to talk about how they've been building resilience into their businesses this year and today my conversation with Lindsay people's Wagner, the editor-in-chief at the cut, the cuts a digital publication that covers Fashion Beauty politics, and much more.^{10}And before she went to the cut, Lindsay was the editor-in-chief at Teen Vogue.^{11}Look Teen Vogue is actually where Lindsay started her professional career in fashion, as an intern cleaning out the fashion closet.^{12}Now, if you've ever seen The Devil Wears Prada, it is a massive room filled with designer shoes.^{13}Gowns hats, scarves bags and accessories.^{14}My points of reference for fashion were, you know, I worked at Payless growing up, like I wasn't like, I really knew anything about fashion.

Segment Summary 1

Guy will be interviewing entrepreneurs in the Global Community such as Brene, Brown, Gary Vee Troy, and Carter. The sponsors of this year's summit can be found at npr.org. Each week, Guy Roz invites entrepreneurs and other business leaders to talk about how they've been building resilience into their businesses. Today the conversation is with Lindsay Peoples Wagner who is the editor-in-chief at Cut which covers fashion, beauty, and politics. Before Cut, Lindsay was the editor-in-chief at Teen Vogue where she started as an intern and only had a point of reference from Payless for fashion.

Other Sentences Used, Enter separated by a comma ','

Segment 2

^{15}I, I liked being creative, but it honestly wasn't until Till college that one of my professors actually took me aside and said, you know, I think maybe want to try to magazine you really like to read you really like to write like you want to be created, maybe this is the outlet for you and so she had actually sent me a listing for a Teen Vogue internship.^{16}That was literally just slipping cleaning the closet, the fashion closet, something glamorous.^{17}And I remember telling her, I was like, they're not gonna accept me like, this is a once-in-a-lifetime opportunity.^{18}This has been like the hills was TV.^{19}And there was a lot of reality television around working in fashion, but they said, yes.^{20}And I was really hooked and excited.^{21}And so I did a ton of different internships to try to figure it out.^{22}I did, you know, I tried celebrity styling and Lala realized I don't want to work for a celebrity until a lot of, you know, design classes and realize that I really want to work at a design house and I really want to work, you know, an editorial.^{23}And so I just hustled a town and moved out here to New York after I graduated from college.^{24}Judge and emailed to everyone furiously and ended up getting an entry-level job at Teen.^{25}Vogue out of school.^{26}Wow, yes sir.

Segment Summary 2

Lindsay liked being creative and her college professor suggested that she try to magazine and sent her a listing for a Teen Vogue internship. The internship was cleaning the fashion closet. Lindsay told her professor that she did not think she would get accepted because it was a once-in-a-lifetime opportunity. Lindsay got hooked on fashion and did a lot of different internships and found out she wanted to work at a design house like an editorial. She moved to New York after college and got an entry-level job at Teen Vogue out of school.

Other Sentences Used, Enter separated by a comma ','

10

Segment 3

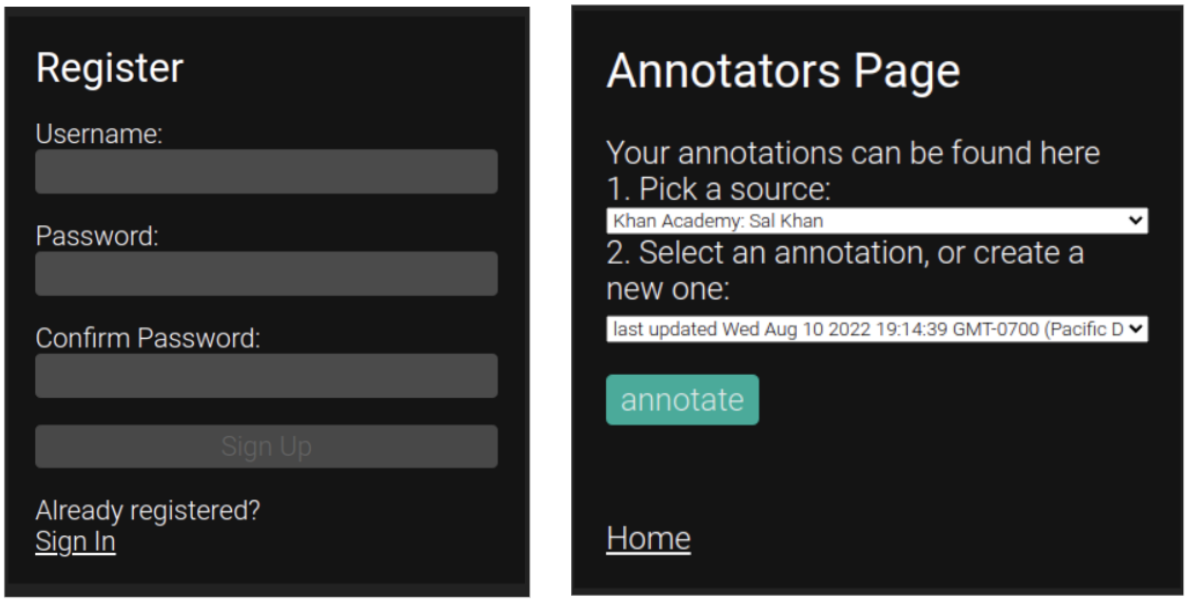
^{27}I mean, there's like, as you say like the hills and there's a kind of a mythology about what it means to be in the publishing World, especially in the fashion publishing world and obviously Devil Wears Prada.^{28}Everybody knows that that film.^{29}And book is there any truth to that isn't quite as Cutthroat as its portrayed really?^{30}Yes, the the Cutthroat missed.^{31}Yes, made sense that I didn't realize I

Segment Summary 3

Figure 4.2: **Main Annotation/Summarization View**. Users are presented with segmented summaries on the left hand side, the right hand side has two text boxes, one for the user to write their own authored gold label summary and another to include sentences beyond the current source text. By allowing the annotator to include any extra information that may have been used to write their summary, the system minimizes context fragmentation and preserves long range dependencies. Speaker diarization is used to to color code pink and light green to indicate speaker turns to users. Each segment and respective summary is referred to as a segment summary pair.

4.5.6 User Interface Engineering and Technical Details

Due to the extensive and taxing nature of summarizing longform spoken dialog, we implemented user accounts and saving to allow intermittent working to prevent annotator fatigue and



Khan Academy: Sal Khan

last updated Wed Aug 10 2022 19:14:39 GMT-0700 (Pacific D

Figure 4.3: **User Login and Progress Saving**. To facilitate multiple annotators the system is hosted online with different user accounts. To ensure summarize quality and prevent degradation due to annotator fatigue, the system saves all of the annotator's intermediate work and not requiring annotations to be completed in a single sitting.

possible decay in annotation quality. Annotators noted that a 25 minute transcript requires around 4 hours and a longer transcript (1 hour or more), requires around 6 hours.

- Annotator accounts: users are able to sign in from any machine to work on their assigned transcript.
- Saving annotator progress: instead of saving an annotator's progress at each level, each segment's summary is saved by the system.

4.5.7 Annotator Instructions and Guidelines

There were a total of 11 annotators, all native English speakers with a college degree. Each annotator summarized 2-3 transcripts and were compensated $20 an hour for their time. Annotators were provided with a detailed document demonstrating step by step how to use the summarization system. The following is an excerpt taken from the introduction of the document:

"The purpose of this task is to summarize a very long podcast transcript. This podcast has been

converted from audio using speech-to-text which means there will be some errors - do your best to try to make sense of what the text is saying if it does not make sense.

Podcast audio transcripts can be extremely long, and summarizing them can be a daunting task. What we are trying to do is summarize in a recursive nature, meaning you write a summary of the transcript, then summarize that summary, and again and so forth until you obtain a much shorter final summary.

This is aided by breaking up a very long chunk of text into smaller chunks. These smaller chunks are individually summarized and then concatenated. To create the next summary, the concatenated summaries are then broken up again into smaller chunks and recursively summarized."

First, annotators were instructed to create accounts in order to save their progress due to the likelihood of not finishing in a single session. Afterwards, users were given a walkthrough and example demonstrating how to write (according to Section 4.5.1) their summaries. Speaker turn color coding and the sentence numbering for additional user specified context were then explained. Annotators were also provided with the original audio file in case the ASR transcript was too difficult to understand.

Critically, annotators were instructed to keep their summaries to approximately 0.45 (45%) the length of the input segment as well as the importance of ensuring periods (or any other sentence boundaries) are properly written to prevent any issues with the segmentation backend. Lastly, users were told to take a break every few hours to prevent fatigue and degradation in summary quality.

4.6 Limitations

While we attempt to address many of the biases that are inherent towards collecting a summarization dataset, there are still biases that must be acknowledged.

1. *Context fragmentation.* While the system does allow for users to readjust local context boundaries by manually including sentences from other segments, the segment-summary pairs are processed discretely and bound to miss some long range dependencies.

2. *Under-determined summaries.* This problem is endemic towards summarization; simply put, it is very difficult to define what a "good" summary is and inherently subjective. Though mitigated by iterative content selection and recursive summarization, final resulting short summaries are still guaranteed to be dependent on the annotator.

3. *Lack of gold transcripts and error propagation.* Clearly, the starting point of collecting long-form spoken dialog transcripts was already imperfect; transcripts can be hard to understand due to incorrect word transcriptions, speaker diarization, and other factors. If an annotator makes a semantic comprehension mistake early on, and unless summarized out, that mistake will be recursively propagated all the way to the final short summary. However, to a user using this dataset for either testing or training speech, these errors are indicative of what is contained in a typical state-of-the-art ASR system's output.

4.7 Conclusion

We created a novel longform summarization annotation methodology where text is recursively summarized to create progressively shorter and more concise summaries in a bottom up manner, presented in a dedicated summarization annotation user interface UI. This results in comprehensive intermediate text summaries that give transparency towards the summary content selection process and flexible length summaries. The methodology is extensible to not only the spoken dialog domain, but also to any general longform text. We hope that the *hierarchical*, *recursive*, and *modular* aspects of this new dataset promotes more research into longform spoken dialog abstractive summarization[9].

[9]Again, data validation, trained summarization baselines, and other experiments are outside the scope of this dissertation and will appear in future work.

Chapter 5: Improving the User Browsing Experience in Summarization Systems

As with System 1, we have seen that automatic speech recognition (ASR) and automatic summarization can be used to build hierarchical summaries that allow users to browse speech data and to navigate and discover interesting content. But because these systems intake disfluent speech data, the subsequent automatic summarization often yields accuracy and readability problems: summaries may skip content, contain false information, or become difficult to read - especially when summaries contain ambiguous pronouns such as “it” instead of named entities. To improve the readability, coherence, and adequacy of summarization we develop a novel method for summary hallucination and quality detection through entity tracking and contradiction assessment while simultaneously correcting for erroneous summaries with guided text decoding. Additionally, we improve topic cohesion through entailment inspired clustering and correct for dialog ambiguity by imputing deictic references through recursive co-reference resolution.

User studies show our system enables users to more easily browse and discover content than previous systems, as well as provide better readability of high level summarization for navigation. From Chapter 4, it is now possible to properly benchmark our system (System 2) against human authored gold standard summaries. We discuss how summarization technologies can be harnessed to help people browse long unstructured information in trustworthy and readable ways.

5.1 Introduction

Although existing systems for hierarchical navigation have shown utility in browsing and navigating content without having to listen to an entire audio, they face several challenges [63, 64]. Summarization models are not well suited for handling speech errors as they are trained on well formed text rather than audio content. Whereas text is structured into paragraphs with topic sentences, audio is far less structured and riddled with speech specific challenges such as disfluencies, incoherence, and ambiguous pronoun references which prevent straightforward language modeling. In order to successfully enable users to browse and navigate longform audio, systems first have to ensure the hierarchically generated summaries are easily readable and coherent.

We identify three key elements within hierarchical summarization that impact user experiences in browsing and consumption of longform audio:

1. *Readability* [148] - summaries that are not immediately comprehensible due to vague pronouns or incoherent grammar impact a user's ability to quickly understand content.

2. *Accuracy* [149, 150] - summaries that are inconsistent with the source text may provide erroneous information and mislead users.

3. *Adequacy* [151] - summaries that fail to capture important meaning of the underlying audio or only capture partial meanings result in users missing possibly critical information.

To address these elements, we developed a training agnostic post processing approach that introduces a set of NLP techniques to improve summarization quality for speech data in ways which help users browse effectively. To improve *readability* we track entities and impute ambiguous coreferences to eliminate vague references. To improve *accuracy* we employ guided text decoding with contradiction assessment to enhance summary correctness. To improve *adequacy* we reorder sentences in the transcript to form a more cohesive input, in turn allowing summarization models to more easily reason through the input content and sufficiently convey the meaning in generated summaries.

We then present a speech browsing interface where the improved summaries are presented in a readable format along with visual information cues that are automatically derived from unstructured speech transcripts. These information cues quickly and intuitively provide information scent to users regarding the quality of summary segments, the degree to which the summarization model compressed the original content, and the amount of information the user has been shown. Thus, by providing users visibility of system status and insight into the state of the underlying audio content, users can make better informed navigation and browsing decisions.

To demonstrate technical performance gains in summary quality and subsequent improved usability in browsing, we evaluate our system in the following three ways:

1. Automatic comparisons of generated summaries with gold standard summary references. Results indicate our system yields a 19% ROUGE-2 relative improvement in Short summaries over baselines.

2. Via human annotation, our system's generated summaries improve by 25% on readability, 10% on accuracy, and 17% on adequacy for specific summary quality dimensions compared to baseline models.

3. A qualitative user study showing that the interface's information cues successfully assists users and that they are able to more easily browse content and forage for information.

Lastly, we discuss how advances in summarization technologies can be harnessed to assist users in browsing long unstructured information in trustworthy and readable ways.

5.2 Background on Automatic Hierarchical Summarization

5.2.1 Criteria for Improving the Usability of Hierarchical Summaries for Audio

Although many off-the-shelf abstractive dialog summarization models are available and have considerably improved, they invariably perform poorly on longform (20 minute+) audio; standalone usage of these models on longform dialog results in problematic summaries that impact usability. Our research carefully considers these concerns and posit them as the following three challenges:

Readability [148]. A summary is only as useful as its ability to be understood by a user. While readability can refer to fluency, it is critical to view this dimension in the context of speech. We make the important observation that diectic references in conversation make readability especially challenging due to constant ambiguous referencing. Confusing, vague and unintelligible text represents a significant pain point in automated summarization because not only does it render a summary useless, it can ruin the users' trust in a system by highlighting a particularly unpleasant failure. Poor readability example: *The store will have a sale despite rain tomorrow.* $\Rightarrow$ *There will have rains tomorrow but it will have sale.* Though the sentence has some syntactical errors, the pronoun "it" is ambiguous; without any additional context, it is impossible to understand what the output is referring to.

Accuracy [149, 150] . Outputting inaccurate summaries has insidious consequences if there is no indication when bad information has been communicated. Users may therefore walk away with incorrect assumptions of the underlying material, undermining trust in automatic summarization systems. Inaccuracy example: *There will be high winds and heavy rain tomorrow.* $\Rightarrow$ *The weather will be cloudy.*

Adequacy [151]. Despite summarization being a fundamentally lossy compression of information, the most useful summaries are ones that accurately communicate the input passage's key ideas. Conversely, if a user reads a summary without reviewing the underlying content, they could unknowingly miss possibly significant information (i.e. a Type II error). Adequacy refers to how much of the meaning in the original source text is also conveyed in the hypothesis [151]. Poor adequacy example: *There will be high winds and heavy traffic on the freeway due to storm congestion.* $\Rightarrow$ *It will be windy.*

In recursive hierarchical summarization, errors in summaries will compound and propagate. Thus, it is important to address these problems at early stages in the summarization process. In this system, our goal is to improve on these three key metrics.

5.2.2 Dataset

The same dataset from Chapter 4, Table 4.1 is used.

5.3 System

In this section, we briefly review the baseline hierarchical dialog summarization system as the initial foundation for our framework. Next, we detail our speech intrinsic design motivations and our corresponding technical contributions towards existing summarization systems, focusing on the criteria defined in Section 5.2.1. Our summarization framework is modular, not requiring any specific summarization model, and integrates external knowledge from language models trained on various natural language tasks with dialog heuristic constraints to construct a robust and unsupervised abstractive summarization system. Recent research has leveraged external models to improve broad domain summary quality by combining knowledge-based approaches with seq2seq neural models [195]. In a similar fashion with regards to leveraging external models (i.e. transformer models trained for different language tasks such as entailment [196]), we propose a hierarchical automatic summarization framework which emphasizes improving system robustness with a specific longform dialog domain focus. Refer to Section 2.1 for notation and conventions.

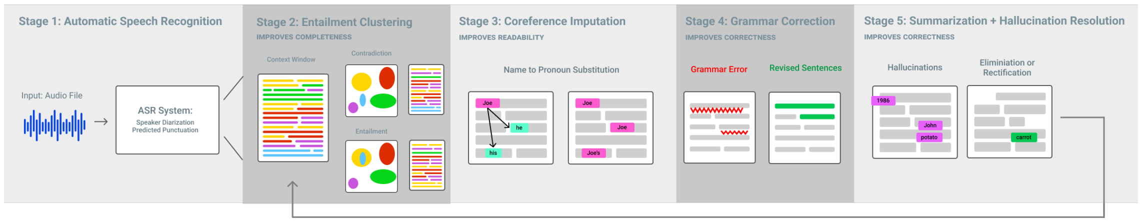


Figure 5.1: **Dialog Improvement Framework**. The overall contribution of the system are a series of training agnostic speech processing steps which improve overall summary text generation quality. Step 2 involves reordering input dialog to increase coherence. Step 3 and 4 imputes vague pronoun references to assist readability and addresses grammatical errors. Step 5 encourages the system to generate more factually consistent summaries. This process is repeated in between recursive summarization levels.

5.3.1 Baseline Dialog Summarization System

The baseline hierarchical summarization system is adapted from [116] and consists of three key components: a topic aware semantic segmentation algorithm for dividing longform text, a summarization language model, and a procedure for establishing higher order relationships [123, 197].

The initial semantic segmentation algorithm is already well suited towards chunking dialog as it fundamentally incorporates concepts that leverage coreferences and other speech cues [77, 198, 199] to begin an initial chunking of the longform transcript. Between our framework and baseline, the initial segmentation procedure is kept identical for subsequent analysis and comparisons, and is not the focus of this work.

We refer to the "baseline" hierarchical summarization instance as `Baseline` and an instance utilizing the framework containing our contributions as `System`. The `Baseline` and `System` both use two summarization language models a `BART-L` model that is finetuned on the SamSum Corpus [200] to handle larger segmented transcript chunks, and a `PEGASUS` paraphrase model for smaller inputs (30 words or less). In between hierarchical recursive summarizations, `System` performs Steps 1-4 (Fig 5.2, Section 5.3.2-5.3.4). Finally the `Baseline` also adopts the hierarchical procedure from [116] as its hierarchical merging procedure.

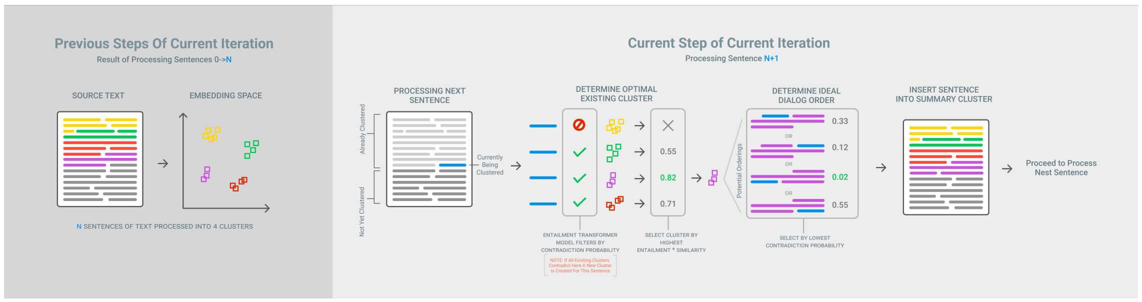
All semantic segmentation procedures, language models, and parameters are kept constant for a fair basis of comparison. Experiments were run on a single RTX 3090; instances where several large transformer models were required to be simultaneously loaded into memory (such as Alg 3), could be easily fit ($\tilde{1}$5GB) within the 24GB of VRAM. For transparency, `System`'s inference, with all steps, typically takes around 3-5x longer (usually 8-10 minutes per transcript, with the obvious exception for longer transcripts) than `Baseline`'s inference.

Another important aspect is the compression ratio of the generated summaries. The individual segments $S_i \in \mathbf{S}$ are iteratively summarized resulting in the same number of segment summaries $H_i \in \mathbf{H}$. As such, the compression ratio can obtained by treating the segments in a document level manner. This is done by concatenating $\mathbf{H}$ into H^c (where the c indicates its concatenated form) and using the unsegmented instance of the input S. The compression ratio is thus defined as:

$$\text{Compression\_Ratio}(\mathbf{S}, \mathbf{H}) = \frac{|\text{Concat}(H_i \in \mathbf{H})|}{|\text{Concat}(S_i \in \mathbf{S})|} \tag{5.1}$$

Note that the norm notation $|\cdot|$ is used to indicate the number of words in the text passage. While an input can be compressed to an arbitrarily length due to the recursive nature of the hierarchial summarization framework and is dependant on the original input's length, summarization levels are stopped at around a 15%-25% compression rate for both `System` and `Baseline` summaries. Regarding the initial input length, longer transcripts would be further recursively summarized.

5.3.2 Improving Adequacy: Entailment Clustering for Temporal Dialog Cohesion



Previous Steps Of Current Iteration
Result of Processing Sentences 0->N
SOURCE TEXT
EMBEDDING SPACE
N SENTENCES OF TEXT PROCESSED INTO 4 CLUSTERS
Current Step of Current Iteration
Processing Sentence N+1
PROCESSING NEXT SENTENCE
Already Clustered
Not Yet Clustered
Currently Being Clustered
DETERMINE OPTIMAL EXISTING CLUSTER
0.55
0.82
0.71
ENTAILMENT TRANSFORMER MODEL FILTERS BY CONTRADICTION PROBABILITY
NOTE: If All Existing Clusters Contradict Here A New Cluster Is Created For This Sentence
SELECT CLUSTER BY HIGHEST ENTAILMENT * SIMILARITY
DETERMINE IDEAL DIALOG ORDER
Potential Orderings
0.33
OR
0.12
OR
0.02
OR
0.55
SELECT BY LOWEST CONTRADICTION PROBABILITY
INSERT SENTENCE INTO SUMMARY CLUSTER
Proceed to Process Nest Sentence

Figure 5.2: **Entailment Clustering Process**. To increase the summarization language model's semantic comprehension of the input text, the temporal ordering of dialog is rearranged and re-ordered to improve cohesion.

A challenge in processing speech is that most speech and conversation is presented in an unstructured manner that may not be cohesive (the degree of logical consistency [201] and continuity [202] of text) or continuous. For example, a speaker discussing an idea could trail off and revisit that same very idea a few sentences later. This leads to fragmented context and incoherently ordered thoughts [203] in ASR transcripts, making it more difficult for a summarization language model to sufficiently capture higher order concepts. Given the nonsequential temporal ordering in which speakers communicate, a reordering speaker of sentences can lead to improved cohesion and context. This step leverages pretrained language models to determine entailment and similiarity to determine a new ordering and segmentation of speech.

Implementation. The operation starts with the previous hierarchical level's summary outputs $\mathbf{H}^{j-1} = H^{j-1}_{1\ldots n}$ as $\mathbf{S}^{j} = S^{j}_{1\ldots n}$ inputs where each individual summary S^{j}_{i} is iteratively processed in a streaming fashion. We consider three criteria to group summaries into semantic clusters. Pseudocode and full procedure is given in 3.

1. A well-formed semantic cluster should consist of summaries that do not contradict one another [69]. Specifically a high entailment beyond only a high semantic overlap (i.e. evaluated by the cosine similarity of `SBERT` embeddings [173]) may not necessarily ensure cohesiveness.

2. Summaries belonging to the same cluster should discuss the same entities. This is enforced by gathering the overlapping set of detected nouns and coreferenced pronouns and clustering if sufficient overlap is found.

3. Semantic clusters should have a natural order of entailment. A summary S_i^j entails another if one is logical predecessor, determined by a transformer language model trained on the MultiNLI dataset [204]. This is a key distinction from prior work as entailment allows dynamic assignment of cluster centers, in turn allowing for the procedure to reorient the temporal order of $\mathbf{S}^j = S_{1 \ldots n}^j$ in a manner that would be easier for a summarization model to process and output complete and semantically consistent summaries.

Entailment-based clustering is distinct because unlike the aforementioned criteria, entailment is not symmetric. Because of this, we can use entailment as a signal to dynamically reorder sentences into a logically coherent progression. This is essential as rapidly moving between different topics and discussing entities out-of-order is common in spoken dialog. For example, consider the following three sentence toy example, designed to demonstrate the incoherence of dialog:

1. *Sentence 1:* Because of the sale, Irene bought apples.

2. *Sentence 2:* The computer had a virus.

3. *Sentence 3:* The store had a sale on apples.

Clearly, a better logical ordering would be *Sentence 3*, *Sentence 1*, *Sentence 2*. Here the premise is set first (store had a sale) with the related event following immediately. The unrelated event (*Sentence 2*) is now not separating the two related events.

Algorithm 3: Entailment Clustering Procedure. The procedure is a core stick-breaking problem, determining which summaries should be concatenated for further summarization in a manner that maximizes cohesive similarity.

Input: This procedure uses the following inputs and models.

1. *Input*: List of summaries $\mathbf{S}^j = S^j_{1...n}$. For simplicity the hierarchical level superscript j is dropped.
2. Model M_{ST}: Embedding `Sentence Transformer` (SBERT), outputs a 0 to 1 score
3. Model M_{NLI}: MultiNLI Language Model (ROBERTA), entailment probability is outputted (0 to 1)
4. Model M_{ENT}: Entity Tagging Language Model (FLAIR [205]), number of overlapping entities is outputted
5. Hyperparameters:
   - $p_{th} = 0.05$: Threshold cutoff for similarity
   - $p_r = 3$: Maximum visible range for concatenating streaming summaries
   - $p_{len} = 72$: Maximum word length for a summary
6. *Output*: L_{out}, list of combined and reordered summaries

```
1  $L_{out} \leftarrow$ list();
2  $L_{clusters} \leftarrow$ list();
3  for $S_i \in \mathbf{S}$ do
4      if $|L_{clusters}| \leq 3$ then
5          $L_{clusters}$.append($S_i$);
6          continue;
7      if $|L_{clusters}| > p_r$ then
8          Remove $L_{clusters}[0]$ and append to $L_{out}$
9      $c_{candidates} \leftarrow$ list();
10     for $S_k \in L_{clusters} : |S_k| \leq p_{len}$ do
11         $sim \leftarrow M_{ST}(S_i, S_k)$;
12         $ent \leftarrow M_{NLI}(S_i, S_k)$;
13         $overlap \leftarrow M_{ENT}(S_i, S_k)$;
14         $weighted\_score \leftarrow$ NORM($sim \cdot ent$);
15         if $overlap > 1$ then
16             $c_{candidates}$.append($[S_k, weighted\_score]$)
17     $c_{chosen} \leftarrow$ MAX($c_{candidates}$);
18     $order \leftarrow$ Higher Entailment Between $M_{NLI}(S_i, c_{chosen})$ and $M_{NLI}(c_{chosen}, S_i)$;
19     $L_{clusters}$.append(concatenate($order$));
20 return $L_{out}$
```

5.3.3 Improving Readability: Coreference Imputation and Grammar Correction

Unlike written prose, conversation can be ambiguous. When an entity (such as a noun or other object) is introduced into a conversation, it is therein referred to using pronouns and other deictic references. While recursively summarizing concatenated inputs, the detrimental impact of vague references is increasingly intensified at subsequent levels, impacting summary readability. To rectify this, we propose using coreference resolution to impute missing entities into vague pronoun references.

Table 5.1: Coreference Imputation Example. The vague reference is given in blue with the coreferenced entity is bolded. Subsequent grammar correction is shown in red. $S_{1...n}$ are generated summaries (previously recursive outputs $\mathbf{H} = H_{1...n}$) for any arbitrary recursive level.

Process	Consecutive Text Segments
Vague Reference	S_1: [...] What do we know right now about this **variant**? [...] S_2, S_3 not related S_4: It's highly divergent from ...
Imputation	S_1: [...] What do we know right now about this variant? [...] S_2, S_3 not related S_4: **This variant** highly divergent from ...
Grammar Fixed	s1: [...] What do we know right now about this variant? [...] S_2, S_3 not related S_4: **This variant** is highly divergent from ...

Implementation. In this operation, each sentence in a summary is considered individually, $S_i = s_1, ..., s_n$. Since coreferences are typically accurate for a local context window, we only search for coreference pairs in a limited sentence span. For example, an "it" mentioned in the first minute of a speech is likely different from an "it" at the end of the speech. We set our limited context window to 3 summaries. Concretely, given a s_i and a context $L_{context} = [s_{i-3}, s_{i-2}, s_{i-1}]$, any identified coreference from $L_{context}$ to s_i is imputed into s_i. The process then iterates by a single sentence; for s_{i+1}, the new context is $L_{context} = [s_{i-2}, s_{i-1}, s_i]$. A key observation of the procedure is that iteratively imputing coreferences will propagate an initial reference to subsequent pronoun references.

Such imputations may induce syntactic grammatical errors (such as subject verb agreements) due to imperfect insertions. Accordingly we use a `T5` [206] based neural grammar rewriter trained on a fluency corpus [207] to correct for small grammatical mistakes. A full procedure is given in Alg. 4.

Algorithm 4: Coreferenced Imputation and Grammar Correction Procedure.

Input: This procedure uses the following inputs and models.

1. *Input*: List of summaries $\mathbf{S}^j = S^j_{1...n}$. For simplicity the hierarchical level superscript j is dropped.
2. Model M_{CRF}: AllenNLP coreference resolution model, outputs the start and end index of a coreferenced word pair
3. Model M_{GRM}: $T5$ trained neural grammar rewriter
4. Hyperparameter $p_w = 3$: coreference context window
5. *Output*: L_{out}, list of coreferenced and grammatically corrected summaries.

```
1  $L_{out} \leftarrow$ list();
2  for $i \in |\mathbf{S}|$ do
3      if $i < p_w$ then
4          $L_{out}$.append($S_i$);
5          continue;
6      $L_{context} \leftarrow [S_k \in \mathbf{S} : i - p_w \leq k < i]$;
7      $S_{impute} \leftarrow \mathbf{S}_i$;
8      $coreferences \leftarrow M_{CRF}(L_{context}, S_{impute})$;
9      for $c_j \in coreferences : c_j[1] \in S_{impute}$ do
10         Impute $c_j[0]$ word reference into $S_{impute}[c_j[1]]]$;
11         $S_{impute} \leftarrow M_{GRM}(S_{impute})$;
12     $L_{out}$.append($S_{impute}$);
13 return $L_{out}$
```

5.3.4 Improving Accuracy: Hallucination Resolution

Hallucinations are a common artifact of large language generative models due to the sheer corpus they are trained with. While these large language models are able to output sentences with high realism, we are interested in outputs that consistently and accurately reflect the input. As such, text generations containing blatant hallucinations or semantically differing meanings are unacceptable; they represent factually incorrect summaries. However, not all hallucinations are bad. In fact, some hallucinations are useful abstractions to generalize multiple ideas into a more

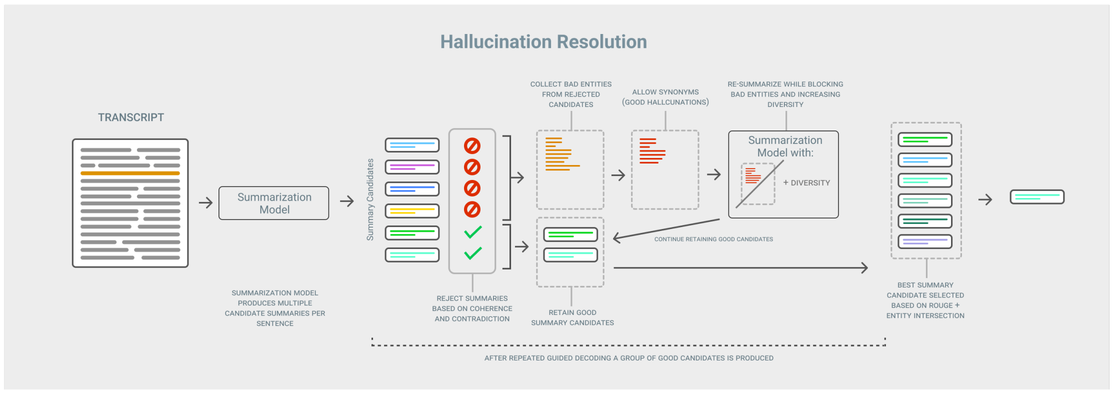

Hallucination Resolution
TRANSCRIPT
Summarization Model
SUMMARIZATION MODEL PRODUCES MULTIPLE CANDIDATE SUMMARIES PER SENTENCE
Summary Candidates
REJECT SUMMARIES BASED ON COHERENCE AND CONTRADICTION
COLLECT BAD ENTITIES FROM REJECTED CANDIDATES
RETAIN GOOD SUMMARY CANDIDATES
ALLOW SYNONYMS (GOOD HALLCUNATIONS)
RE-SUMMARIZE WHILE BLOCKING BAD ENTITIES AND INCREASING DIVERSITY
Summarization Model with:
+ DIVERSITY
CONTINUE RETAINING GOOD CANDIDATES
BEST SUMMARY CANDIDATE SELECTED BASED ON ROUGE + ENTITY INTERSECTION
AFTER REPEATED GUIDED DECODING A GROUP OF GOOD CANDIDATES IS PRODUCED

Figure 5.3: **Hallucination Resolution Process**. Through guided text decoding, the system re-ranks and selects the most plausibly accurate summaries for subsequent recursive summarization.

succinct categorization. An example of a positive hallucination would be replacing references to cars, buses, and boats with the label *vehicles*.

Implementation. The procedure is iterative: the system generates multiple summaries based on given parameters and subsequently filters out poor summaries, providing feedback to improve the next iteration's text generation. This procedure is limited to 3 runs per (S_i, H_i) pair.

Assuming the first summary has already been generated, we start by addressing negative hallucinations. These are characterized by the introduction of entities that are not generalizations of entities from the input passage. We use a Part-of-Speech (PoS) tagger [205] and Named Entity Recognition [76] to identify entities found in a generated hypothesis H_i and compare it to the set of entities found in the original input S_i. By using existing word knowledge systems [208], we can quickly and computationally tractably determine whether or not H_i's hallucinated entities are proper generalizations of existing entities or truly inconsistent with R_i. Second, we use an entailment model (identical to Alg. 3) with the original input S_i as the premise and H_i as the hypothesis and check for logical consistency [69].

We adapt BEAM search to decode multiple candidate summaries $H_i^{k \in K}$ where K = NUM_BEAMS [209] with the following parameters:

1. *Block tokens.* Summaries that contain flagged hallucinations have the respective tokens passed in as blocked tokens, suppressing the hallucinated words from being generated on

subsequent iterations[1].

2. *Increasing BEAM Search Space.* [209] Simultaneously, we increase the diversity parameters for grouped beam search on subsequent iterations to motivate generating more unique candidate summaries.

After running guided BEAM search, we process all candidate summaries and rank them based on their readability and accuracy using Eq 5.2. For readability, we use a language model LM_{wfd} trained on sentence well-formedness dataset[2] to score the syntax quality (0-1) of each summary. For accuracy, we use ROUGE-2 [7] to ensure that the summary and the input passage are similar in content. Eq 5.2 gives (0-1) an estimate encompassing both of these attributes.

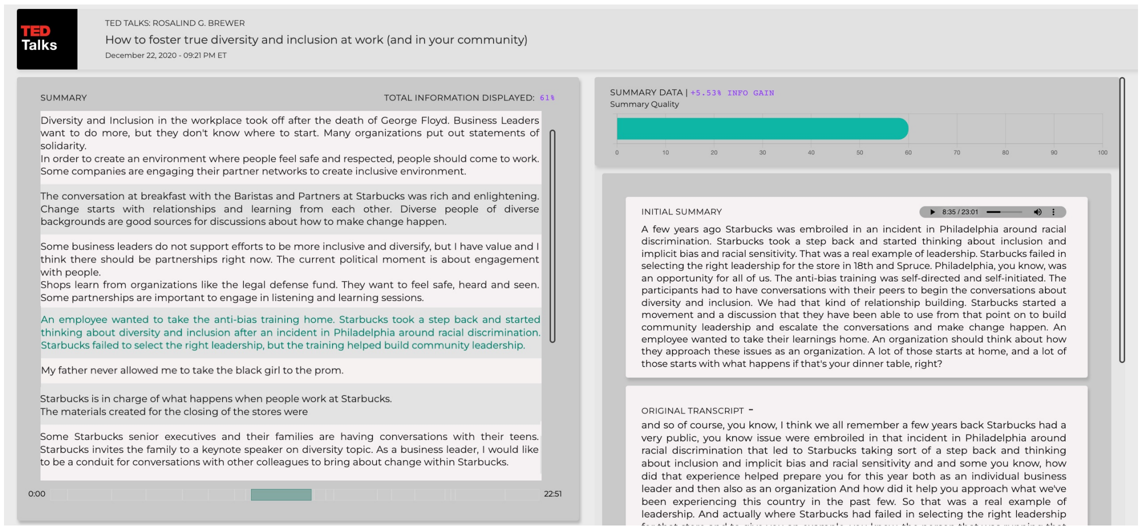
$$\text{Quality}(H_i, S_i) = \alpha * \text{LM}_{wfd}(S_i) + \beta * \text{ROUGE}_2(H_i, S_i) \tag{5.2}$$

5.3.5 User Interface

User Interface Overview. Having established a robust post-processing procedure tailored towards improving readability, accuracy, and adequacy of the underlying summaries, we are now well positioned to build a viable system for users to consume this content. We present a user interface 5.4 which provides multiple real time information signals to the user alongside a readable summary view. Building upon previous work which utilized multiple levels of hierarchical summaries to improve a user's ability to recover from accuracy and adequacy concerns inherent with generated summaries, we present a more streamlined hierarchical summary interface incorporating a more human readable *Short Summary* document while still including one hierarchical layer to capture the effect of previous hierarchical interfaces that allow users to drill deeper. In addition, we present meaningful heuristics to the user describing the estimated quality of the summary segments, the amount of information users would gain by reading more detailed content for a specific

[1]Critically, in the case of a false positive, where a hallucination is classified incorrectly, it would limit the BEAM search space and limit the abstractive capabilities of the language model.

[2]`https://huggingface.co/datasets/google_wellformed_query`, $\alpha = 0.5$ and $\beta = 0.5$.


TED TALKS: ROSALIND G. BREWER

How to foster true diversity and inclusion at work (and in your community)

December 22, 2020 - 09:21 PM ET

SUMMARY

TOTAL INFORMATION DISPLAYED: 61%

Diversity and Inclusion in the workplace took off after the death of George Floyd. Business Leaders want to do more, but they don't know where to start. Many organizations put out statements of solidarity.
In order to create an environment where people feel safe and respected, people should come to work. Some companies are engaging their partner networks to create inclusive environment.

The conversation at breakfast with the Baristas and Partners at Starbucks was rich and enlightening. Change starts with relationships and learning from each other. Diverse people of diverse backgrounds are good sources for discussions about how to make change happen.

Some business leaders do not support efforts to be more inclusive and diversify, but I have value and I think there should be partnerships right now. The current political moment is about engagement with people.
Shops learn from organizations like the legal defense fund. They want to feel safe, heard and seen. Some partnerships are important to engage in listening and learning sessions.

An employee wanted to take the anti-bias training home. Starbucks took a step back and started thinking about diversity and inclusion after an incident in Philadelphia around racial discrimination. Starbucks failed to select the right leadership, but the training helped build community leadership.

My father never allowed me to take the black girl to the prom.

Starbucks is in charge of what happens when people work at Starbucks.
The materials created for the closing of the stores were

Some Starbucks senior executives and their families are having conversations with their teens. Starbucks invites the family to a keynote speaker on diversity topic. As a business leader, I would like to be a conduit for conversations with other colleagues to bring about change within Starbucks.

SUMMARY DATA | +5.53% INFO GAIN

Summary Quality

INITIAL SUMMARY

A few years ago Starbucks was embroiled in an incident in Philadelphia around racial discrimination. Starbucks took a step back and started thinking about inclusion and implicit bias and racial sensitivity. That was a real example of leadership. Starbucks failed in selecting the right leadership for the store in 18th and Spruce. Philadelphia, you know, was an opportunity for all of us. The anti-bias training was self-directed and self-initiated. The participants had to have conversations with their peers to begin the conversations about diversity and inclusion. We had that kind of relationship building. Starbucks started a movement and a discussion that they have been able to use from that point on to build community leadership and escalate the conversations and make change happen. An employee wanted to take their learnings home. An organization should think about how they approach these issues as an organization. A lot of those starts at home, and a lot of those starts with what happens if that's your dinner table, right?

ORIGINAL TRANSCRIPT -

and so of course, you know, I think we all remember a few years back Starbucks had a very public, you know issue were embroiled in that incident in Philadelphia around racial discrimination that led to Starbucks taking sort of a step back and thinking about inclusion and implicit bias and racial sensitivity and and some you know, how did that experience helped prepare you for this year both as an individual business leader and then also as an organization And how did it help you approach what we've been experiencing this country in the past few. So that was a real example of leadership. And actually where Starbucks had failed in selecting the right leadership

Figure 5.4: **System 2 User Interface**. The System UI displays the processed audio file. The left hand side gives a digest of short summaries, broken up by semantic summary segments. An estimate of how much total information is captured by these short summaries is given under "Total Information Displayed". By moving the cursor onto a summary, the user is presented with additional information such as the summary's estimated quality, an estimate of how much additional information is contained in a longer more detailed summary, said more detailed summary, and the original ASR transcript with the respective audio segment.

section, and the estimated total amount of information exposed to the user at any given time when interacting with the system.

On the left side of the interface, the shortest summary is displayed with clear borders indicating distinct summary segments. Users may hover over any sentence in this high level summary view which will dynamically populate information on the right side of the interface. This right side interface includes a dynamic visualization of the estimated summary quality (Eq. 5.2) of the sentence the user has hovered over. Below on the right side, the system presents to the user an intermediate summary with greater detail as well as the estimated amount of additional information the user would gain from spending additional time reading this more detailed summary.

As users hover and drill down into various summary segments, the total information displayed to the user counter increases to give the user a better global idea of how much information they have been exposed to relative to the entirety of the source material. A collapsible view of the original transcript and audio player for the original audio are also provided to the user on this side of the interface should they wish to view the full information for this subsection of the original content. Finally, the bottom left side includes a temporal navigation bar which visualizes summary segments temporally to provide an alternative navigation mode for the user based on time instead of linear scrolling similar to previous interfaces.

Assisting User Navigation In the interface, the user is provided three critical heuristics regarding the summaries they are reading:

1. *Total Information:* An estimation of the overall amount of content the user has read and encountered; continuously updating as the user reads.

2. *Summary Quality*: An estimation of the quality of the summary the user is currently reading.

3. *Information Gain*: An estimation of additional information a user would gain relative to all information contained in the audio file, by exploring further into the hierarchical levels of the content at hand.

These heuristics are displayed in an intuitive and automatic manner which allows users to read through high level summaries quickly while providing the user with enough information to give them an informed decision as to whether to drill more deeply into a specific section of the content either by reading a more detailed summary or even the original transcript and audio through which various summary sentences are based upon.

Users may use the estimated summary quality to decide whether they trust the quality of a high level summary sentence or section and use this as a decision point for drilling down or continuing on. Providing granular summarization quality heuristics to the user allows them to more quickly browse and navigate the source material. Users may feel more confident reading only high level summaries of significant portions of the content due to high summary quality heuristic ratings while looking into more detailed summaries or original transcripts and audio where summary quality falls below their acceptable thresholds. This provides users informed agency in how they consume summarization content while building transparency and ultimately greater trust in the system.

Additionally, users may reference the estimated information gain from reading a more detailed summary of a given section to decide whether spending the additional time reading into that particular area is a worthwhile time investment. The estimated information gain is determined based on calculating the remaining information based on the Information Retained.

Information Retained. Information theory is a popular framework for analysing the compression and completion of information in natural language processing. While information theory relates information to the number of bits needed to disambiguate probabilistic uncertainty, we recognize that such an analysis, when applied to generated text, requires constructing a global joint distribution over users' subjective interpretations of different sentences. As such we opt for a simpler analysis.

Simplifying, we define a proxy for the passage's information content to be the total sum of nouns and verbs identified in a passage. The rationale and oversimplification is as follows: the most important words within a sentence are typically proper nouns, common nouns, verbs, and other named entities. We use a state-of-the-art NLP parts of speech tagger to identify all verbs and noun occurrences in the original transcript.

To evaluate the proportion of information retained (*abbr.* IR) by a particular summarization framework over a particular input summary, we compare the original transcript ASR input S_i^{ASR} spanned by a particular summary $H_i^{j=3}$ (in other words, the amount of text that H_i is responsible for summarizing). This is done for *Short Summaries.* We run a parts-of-speech (PoS) tagger to count the overlap of (unique) nouns and verbs in the generated summary and source text to construct the amount of information retained by our system. This heuristic has the behavior of punishing the summary for omitting the previously specified grammatical objects. *This can be viewed as a modified instance of ROUGE-1 (with the grammar objects filter) recall, which computes the word overlap of a source and reference, out of all reference words.*

$$IR(H_i^{j=3}, S_i^{ASR}) = \frac{\text{Count}(w_h \in \text{PoS}(H_i^{j=3}))}{\text{Count}(w_s \in \text{PoS}(S_i^{ASR}))} \tag{5.3}$$

Note: w_h and w_s are each word in the generated summary and input source text, respectively.

Summing over $IR(\mathbf{H}^{j=3}, \mathbf{S}^{ASR})$ estimates the overall information captured by all *Short Summaries*, resulting in a 0-1 fraction. Complete Information Retained estimates typically result around 0.40 – 0.60. Due to the resemblance with ROUGE-1 recall, we expect the heuristic to correlate and perform similarly well. We experimented with a weighted version of Eq. 5.3 where proper nouns and named entities were given increased importance, but did not observe significantly material changes.

Furthermore, as users transition from reading a high level summary to a more detailed summary and potentially an original transcript section, common key topic phrases are highlighted between them and users can begin to more clearly see the progression an abstract summarization model has taken in summarizing the content while preserving key ideas.[3] Exposing the user to these intermediate data points helps build better explainability into an otherwise opaque machine learning based system.

Finally, users are provided a quantifiable heuristic representing the total amount of information which is exposed to the user. This final heuristic is based on the complexity intersection between the short summaries and the original source text. Providing this heuristic allows users a further decision point where they can decide when they are satisfied with the extent of their consumption of a given piece of content. An example walkthrough of how a user might engage with the system's aforementioned visual information cues is given below:

1. User reads *Short Summary* segments 1-10 via the left side document and is satisfied with the level of information they are consuming.
2. Upon reading *Short Summary* 11, the user becomes more interested and decides they would like to learn more.
3. The user hovers over *Short Summary* 11, surfacing summary metrics and the right side view with a more detailed summary. The user notes the summary quality is high as well as the estimated information gain (Eq. 5.3), suggesting the intermediate summary is worth reading.

[3]While not shown in the particular example in Fig. 5.4, this can be seen in the yellow highlighting in Fig. 3.1.

4. The user reads the intermediate summary that *Short Summary* 11 is based upon and is satisfied, and decides not to read the original transcript for this segment.

5. The user continues on reading *Short Summaries* 12-20.

6. The user reads *Short Summary* 21 and suspects the summary is erroneous, disfluent or confusing due to new terminology not related to what they have thus far encountered.

7. The user hovers over *Short Summary* 21 and sees the summary quality for this segment is low confirming the users suspicion.

8. The user sees the estimated information gain to be low for segment 21 (in this case due to the erroneous "new terminology" being the only meaningful information in the current segment) hinting that the intermediate summary and transcript do not add much additional information. As a result, the user opts to listen to the original audio while skimming the transcript. The user determines the ASR transcription erroneously transcribed the "new terminology" leading to an inaccurate summary and now understands this "new terminology" was not present in the audio.

9. The user notes at this point, they have been exposed to 80% of the total information contained in this content based on the global count on the left side view, and decides to quickly skim the remaining high level segments 22-30 as they are satisfied with the information they have consumed already.

10. The user concludes consuming the content, having a clear understanding of the content being conveyed by the underlying audio, despite only having listened to a minimal subset of the original audio itself.

Ultimately, this interface provides the user a highly readable summary view while simultaneously surfacing critical heuristics that assist the user in making informed decisions on where to spend their time when consuming longform spoken dialog content. These heuristics not only provide the user greater agency and more time efficient consumption but also increase the trust and explainability of the system to the user.

5.4 Evaluation

The quality of a summarization system revolves around its ability to distill key ideas from many longer passages. Evaluating a language model's text generation is essential towards understanding the model's performance and suitability for usage [141]. In this evaluation, the summary text generations $H_i^j \in \mathbf{H}^j$ are investigated in three aspects:

1. The hierarchical level (superscript ^{j}) where the *Short Summary* is the highest hierarchial summarization level.

2. The segment level (subscript i), where each segment summary pair, (S_i, H_i) is individually considered.

3. The document level ($H^c = \text{Concat}(H_i \in \mathbf{H})$, where a hierarchical level's summaries are concatenated together and considered all at once.

Due to differences in subsequent level merging and segmentation, `Baseline` and `System` will have different (pairwise misaligned) individual segment inputs to each respective summarization system. This results in different content that is summarized, making it impossible to individually directly compare corresponding `Baseline` and `System` segment summaries. In other words, $S_i^{system} \neq S_i^{baseline}$ implying H_i^{system} and $H_i^{baseline}$ cannot be fairly compared. However, taking the concatenated individual segments at a document level and then comparing `Baseline` and `System` summaries solves this problem[4]. For any given level j, document summaries $H^{system,\,c}$ and $H^{baseline,\,c}$ now holistically summarize the entirety of the same initial input, and thus can be compared to each other. The same considerations apply for comparing individual `Baseline` or `System` summaries to a reference summary, R^c. It follows that an individual segment summary

[4]This is a standard practice when evaluating longform text generation, typically seen in machine translation.

pair, (S_i, H_i), can only be assessed at a general level, and cannot be compared between `Baseline` and `System` instance. Unsurprisingly, given the number of optimizations in `System` that specifically account for speech and dialog based noises, we see a considerable improvement over the Baseline instance - all without additional training or labeled data.

5.4.1 Automatic Evaluation

We evaluate the `Baseline` and our `System` on the 10 (out of 25) transcripts containing gold standard reference summaries using automatic metrics. We use a standard automatic metric ROUGE [7] as cheap and inexpensive methods of evaluating our method and baseline. Concretely we evaluate $H^{system,c}$ and $H^{baseline,c}$ against R^c for all hierarchical levels[5] $j \in \{1, 2, 3\}$, though the most weight should be given at the final $j = 3$ *Short Summary* level. Per Chapter 4, each transcript had 2 different annotators write hierarchical summaries; as such the final score for a transcript is the average ROUGE F_1 scores of both of the 2 annotated references. We use the following ROUGE (F_1) score variants: ROUGE-1 (unigram), ROUGE-2 (bigram), and ROUGE-L (longest common sequence). ROUGE-L can be seen as a measure for fluency[6] while ROUGE-1 and ROUGE-2 are used as proxies for adequacy. As noted in [7], ROUGE-2 is better suited for evaluating document summarization; as a result we place specific **emphasis on ROUGE-2**; in general ROUGE-2 scores are considerably lower than ROUGE-1 scores and is especially true for longer sequences. Lastly, we run ablation studies to determine which of the steps in our pipeline are most effective Summaries are evaluated at the *document* level.

Automatic Evaluation Results.

Digging into the results, we can observe three interesting phenomena:

1. A steady and drastic decrease in ROUGE score in both `Baseline` and `System` as a function of the hierarchical level, which is explained by the increased difficulty of content selection due to input length.

[5]Some transcripts have more than 3 levels due to their substantially longer run times and initial word counts. In those instances, level $j = 3$ skips directly to the final *Short Summary* (last level) in the results.

[6]The intuition is that if a generated summary more closely follows the ordering of words in the reference, then the generated summary is more fluent [150].

Table 5.2: ROUGE-N comparisons of the `Baseline` and `System`. The F_1 score for each ROUGE instance is reported and particular emphasis is placed on ROUGE-2 for document (long-form) level summary evaluation. While `System` outperforms `Baseline` on all experiments, the ROUGE-2 *Short Summary* performance is distinctively better. `System` is run with all steps.

Model & Level-j	ROUGE−1	**ROUGE-2**	ROUGE−L
`Baseline`-3 (*Short Summary*)	0.434	0.097	0.164
`System-3` (*Short Summary*)	**0.470**	**0.115**	**0.191**
Short Summary % Improvement	8.0%	**19%**	23%
`Baseline`-2 (*Intermediate Summary*)	0.595	0.198	0.234
`System`-2 (*Intermediate Summary*)	0.641	0.214	0.260
Intermediate Summary % Improvement	7.7%	8.5%	11%
`Baseline`-1 (*Long Summary*)	0.670	0.405	0.432
`System`-1 (*Long Summary*)	0.711	0.463	0.471
Long Summary % Improvement	6.0%	14%	9.0%

2. A more pronounced improvement in the `System`'s generated summaries at the *Short Summary* level. Clearly addressing speech errors at each level before allowing them to compound further is critical. This is further supported by the ablation studies (Table **??**) demonstrating how individual preprocessing steps contribute to only a subset of total performance gains.

3. The overall low performance of summarization systems at the *Short Summary* level is apparent and highlights the intrinsic difficulty of this dataset, and in particular, the prospect of adequately and concisely summarizing 20-45 minute audio files into only 500-850 words.

Ablation Experiments. We run separate instances of our `System`'s pre and post processing steps: coreference imputation 5.3.3, cohesion clustering 5.3.2, and guided decoding 5.3.4. In comparison to the full method's (`System`) results in Table 5.2, individual aforementioned components usually improve, with the exception of Level-2's Guided Decoding. The offending score is italicized in red in Table 5.3, which under-performs the baseline by 5%. Scores that are close to Baseline's performance are colored blue.

Level-2's Guided Decoding's behavior is difficult to precisely ascertain; however, we hypothesize that the restrictions placed by guided decoding on the text generation adversely affect ROUGE evaluation due to the rejection of inconsistent candidate summaries. It is important to note that

ROUGE cannot evaluate the consistency (accuracy) of a text generation and must be done with a human; unfortunately running a comparison study dedicated to ablations is far too costly.

Moreover, these performance gains are not purely additive in their improvement nature. That is to say, these are not *strictly* exclusive in performance gains; total performance gains likely share overlap in all 3 steps and is difficult to truly disentangle which pipeline steps improved specific dimensions from 5.2.1.

Table 5.3: Ablation study of individual steps from the `System` framework. Scores that are close to `Baseline`'s performance are colored in blue and scores that underperform `Baseline` are colored in red.

Model & Level-j	ROUGE−1	ROUGE−2	ROUGE−L
`System-3` (Reference)	**0.470**	**0.115**	**0.191**
`System`-3 Clustering Only	0.445	0.108	0.187
`System`-3 Imputation Only	0.434	0.110	0.187
`System`-3 Guided Decoding Only	0.432	0.108	0.184
`System`-2 (Reference)	0.641	0.214	0.260
`System`-2 Clustering Only	0.602	0.206	0.251
`System`-2 Imputation Only	0.613	0.201	0.247
`System`-3 Guided Decoding Only	0.595	0.199	*0.221*
`System`-1 (Reference)	0.711	0.463	0.471
`System`-1 Clustering Only	0.690	0.446	0.448
`System`-1 Imputation Only	0.692	0.449	0.451
`System`-1 Guided Decoding Only	0.691	0.441	0.450

Still, it is important to reiterate that ROUGE simply measures lexical overlap and is not a substitute for human evaluation. The reliance on n-gram matching can be an issue for long-text generation [210] as its evaluation does not contain coherence, flow, grammar, and factual correctness.

5.4.2 Comparison Study Methodology

For human annotators, we evaluate on 8 transcripts that do not contain gold labels. With a directed human annotated study we are able to measure more explicit dimensions dimensions of readability, adequacy and accuracy (Section 5.2.1). We hired 4 annotators that were paid $20 per hour. All annotators are native English speakers and were instructed to assess 10 of the the transcripts from Table 4.1 (2 Ted Talks, 4 NPR Podcasts, 2 Bloomberg Wealth). Each annotator reported a total of $8 - 10$ hours spent performing the evaluation tasks. Note, this section is solely concerned with evaluating the *Short Summary*, or the highest hierarchical level's quality in specific dimensions.

Readability, Accuracy, Adequacy Evaluation

Annotators were instructed (in accordance with 5.2.1) to assess segment summary pairs for:

1. *Readability* on a 1 to 5 scale; is the individual summary fluent and unambiguous, leading to easy reading comprehension?

2. *Accuracy* in a binary `Y/N` fashion; is the segment's summary meaning consistent with the original source text?

3. *Adequacy* in a binary `Y/N` fashion; does the segment's summary sufficiently cover the main details contained in the source text?

Each transcript was evaluated by 2 different annotators[7] in order to obtain an inter-annotator agreement score (Krippendorff's Alpha) [211]. A total of $N = 791$ unique segment summary pairs given as randomized rows each containing: a `Baseline` generated summary $H_i^{baseline}$ with the original portion of the ASR transcript $S_i^{baseline}$ that summary $H_i^{baseline}$ covers, and a `System` summary H_i^{system} with the original portion of the ASR transcript that S_i^{system} that H_i^{system} covers; this is done only at the *Short Summary* ($j = 3$ or the last level). Summaries are evaluated at the *segment* level.

Coherence. Lastly, annotators assessed the overall document's readability and coherence; in other words, how well do $H^{baseline,\,c}$ and $H^{system,\,c}$ logically flow or "hang together" [212]? Annotators were presented with all *Short Summaries* consecutively and were instructed to count each

[7]The lower (than expected) IA scores for Readability stem from the latitude evaluators were afforded; for example a 3 or a 4 out of 5 could be quite similar but still counted as agreement. The lower IA scores for Adequacy are italicized, emphasizing the inherent subjectivity in evaluating summary adequacy (and to some extent, accuracy); what one annotator rates "adequate" has far more subjectivity than readability and accuracy.

Table 5.4: Readability, Accuracy, Adequacy human evaluation of the `Baseline` and `System` generated *Short Summaries*. `System` is run with all steps.

Model & Level-j (*Short Summary*)	Readability	Accuracy	Adequacy
`Baseline`-3	3.55	0.78	0.63
`System`-3	**4.10**	**0.87**	**0.74**
% Improvement	15%	12%	17%
`Baseline`-3 Inter-Annotator (IA) Agreement	0.26	0.28	*0.18*
`System`-3 Inter-Annotator (IA) Agreement	0.24	0.27	*0.16*

time they deemed there was confusion regarding logical consistency [201] or continuity [202] between 2 sentences. Summaries are evaluated at the *document* level.

To calculate the overall coherence of a list of *Short Summaries*, we assess the overall document holistically. For this evaluation, sentences are treated pairwise and individually iterated (by one). Annotators were told to count the pairwise dissonances between sentences, regardless of summary boundaries. In the instance where the underlying content had a shift in content, such as a break in between two separate summaries, was not counted. However, in the opposite case, when there was a break between summaries but the pairwise summaries were clearly related and separated, the imperfect split was punished and counted in the dissonance tally. This is denoted by the function $BREAK()$. The *coherence* score, Eq. 5.4, is computed by the fraction of incoherent sentences out of all pairwise sentence pairs.

$$COHERENCE(\mathbf{H}^{j=3}) = 1 - \frac{\sum_{i=1}^{s\in S} \mathbb{1}(BREAK(s_{i-1}, s_i))}{|\mathbf{H}^{j=3}| - 1} \tag{5.4}$$

A perfect score of 1 would result when there are no issues with sequential ordering and a minimum score of 0 would result when every possible pairwise ordering is problematic. Note that this is ultimately a subjective task to the reader to deem what is coherent and what is not. Annotators were told to use their best judgment.

5.5 Qualitative User Study

In our user studies, we investigate the following:

Table 5.5: Overall coherence of *Short Summaries* in `Baseline` and `System` instances. A score of 1 indicates a perfectly coherent summary and a score of 0 indicates total dissonance between sentences, computed by Eq. 5.4. `System` is run with all steps.

Model	Coherence (*Short Summary*)
`Baseline`-3	0.764
`System`-3	**0.836**
% Improvement	**9.4%**

- *RQ1:* Do the heuristics driving the information cues match what users intuitively and internally expect?
- *RQ2:* What drives audio browsing strategies? How do users incorporate visual information cues in their audio browsing strategies?

5.5.1 Methodology

We recruited 10 participants[8] (6 male, 4 female, average age of 25) from mailing lists of graduate and undergraduate students and working professionals from a diverse range of technical backgrounds; users were compensated at \$20 per hour. Each study lasted from 1 to 1.5 hours, averaging 1.2 hours, using the remaining 7 unevaluated transcripts. For this study we evaluate two instances of our longform speech browsing system (5.4), an instance with visual information cues $\texttt{UI}_{cues}$, and an instance without, $\texttt{UI}_{base}$. For $\texttt{UI}_{base}$, we remove the *"Total Information Displayed"* and the top right box containing *"Summary Quality"* and *"Information Gain"*. The steps of this user study are order sensitive and conducted in a semi-structured interview format.

Participants were first introduced to the concept of an AI-driven summarization tool and given a brief tour of the $\texttt{UI}_{base}$ instance of the summarization interface using the same control transcript, "Diversity and Inclusion". After a supervised tutorial stepping through the hierarchical nature of the interface, participants were instructed to choose a different file from the audio library that they found interesting and use $\texttt{UI}_{base}$ to browse the transcript. Users had no time constraints browsing $\texttt{UI}_{base}$ and were told they could stop either once they felt satisfied with the amount of content

[8]These are different individuals from the human annotators in Section 5.4 and 5.2.2.

consumed or at their leisure. This concludes $\texttt{UI}_{base}$ operations.

To acquaint participants with $\texttt{UI}_{cues}$'s visual information cues, we again used the same control transcript to introduce and abstractly explain estimated summary quality, information gain, and total information displayed. Specifically, users were told to note the total initial information displayed percentage on the user interface; this is the starting estimate for how much information can be gained by solely reading *Short Summaries*. Participants were then instructed to explore the control transcript and to fully understand the information cues' associated behaviors and intended purpose, and ask any questions. Once done, using $\texttt{UI}_{cues}$, participants chose a new unseen transcript to browse; participants were told to use the visual information cues as they deemed fit. They were also informed that it was perfectly acceptable to ignore some or all information cues completely if they found them to not be beneficial.

After users finished browsing their selected audio file with $\texttt{UI}_{cues}$, participants were told to listen to the complete audio to obtain a complete understanding of the audio file's underlying content in order to properly reflect on how well the summaries and heuristics (information cues) captured the information from the audio. Participants studied and compared the *Short Summary*'s individual information gain and summary quality components. Users were then asked for their thoughts on $\texttt{UI}_{cues}$ and if they matched what they intuitively expected.

Once finished, we asked participants a set of holistic questions (SUS: System Usability Scale) [213] about their impressions on both $\texttt{UI}_{cues}$, *specifically addressing the functionality of the visual indicators*, with $\texttt{UI}_{base}$ as a frame of reference. We score the SUS survey with standard conventions and investigate the individual categories. Typically, a SUS score of 68 is considered the margin for usability[9].

System Usability Scores (SUS) are computed by:

1. Odd categories (X): sum all odd numbered questions and then subtract 5 from the total

2. Even categories (Y): sum all even numbered questions and subtract the total from 25

[9]`https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/`

3. Final score: $(X + Y) \cdot 2.5$, where a score below 68 indicates design issues and scores higher than 68 indicate minor improvements required. Rough thresholds are as follows:

   - 35: Poor
   - 55: Okay
   - 75: Good
   - 85.5: Excellent
   - 100: Best imaginable

5.5.2 Findings

RQ1: User Interpretations of Speech Information Cues

In general, participants felt that the summaries adequately supported their understanding of the podcasts and that the compression of the audio was accurate. 9 out of 10 participants reported that the percentage of initial information displayed approximately matched what they expected after listening to the podcast. 8 out of 10 participants were highly positive about the "information gain" displayed for each *Short Summary* which estimates how much additional information could be gained from exploring more detailed hierarchical levels. P1 stated that their expectation of information gain was accurately reflected by the system and that it was a useful indicator for browsing summaries: *"for a system to be aware of how much information it's leaving out...it makes it all that much more powerful. An informed user knows what he does not know."*

However, the information gain heuristic is not perfect. P1 also mentioned that this heuristic could at times be misleading and rationalized why, *"here, I was surprised because it seemed like [the speaker] rephrased the same idea several times, but the system tagged it as having a lot of information left out. My guess is that a lot of synonyms (Primary Health Care, Ministries, Government) were used interchangeably but all ended up meaning the same thing"*. Notably, this reflects the drawbacks of a simple heuristic since it technically followed its designed and intended behaviors of Eq. 5.3 used in estimating information gain. P2 found the concept of information gain confusing at first, requiring *"clicking through one [Short Summary] with high error to understand what it meant"*, and approached concept of total information displayed with skepticism since *"it is hard to approximate what a certain percentage of the entire podcast represents"*. The information gain figure abstracts out a lot of information. Although a useful heuristic, it was also not immediately obvious how to interpret it and required users to see multiple examples before they were able to familiarize themselves with the concept and gain an intuitive understanding for its use.

RQ2: Audio Browsing Strategies and Browsing with Information Cues

We observed that participants shifted between two distinct patterns of browsing, linearly and sporadically. Most participants started by browsing summaries linearly, occasionally stopping to investigate more detailed summaries or play audio. Some participants who tried to progress through the summaries more efficiently tended to keep with the linear trajectory but sometimes skipped some *Short Summaries*. They noted how the hierarchical summarization was able to adequately summarize content, allowing them to quickly decide if a *Short Summary* was interesting or relevant. P5 in particular was impressed by the "compression" that they observed when they were listening to a Navy SEAL's Ted Talk. *"In that one, there was a detailed account of a back and forth chase, but it was compressed into 'they shot at the wrong people', which was cool."* P7 likewise found this compression fascinating and was impressed by the way the model was able to filter out certain anecdotes and figures of speech that they considered speech filler. Conversely, P6 described instances were the summarization was insufficient but was able to leverage the hierarchical nature of the system to recover comprehension by *"clicking on a few of the summaries that were less clear... to see the initial summary"*.

The other browsing pattern was scrolling up and down between *Short Summaries*. This behavior was exhibited by all users at some point during the user study. Often users scrolled to see if they topic they were currently reading about was discussed later, or if it was just a passing comment. Other times when some users first began using the system and encountered a *Short Summary* they deemed to be suspicious, they jumped ahead to see if additional summaries were also suspect. This behavior demonstrated the consequence of early mistrust in the AI when automatically generating summaries.

Participants' browsing behaviors can be characterized into two decisions: (i) is there enough evidence to continue exploring along their current hierarchical path? and (ii) is there enough evidence to stop reading further hierarchical levels and move on to other content. For (i), P8 noted that, *"if a sentence in the summary was indicated as contributing a lot to the information of the podcast, I'd read it a little more carefully, or if it didn't make sense I tried a little harder to understand it."* Specifically, P9 referenced an instance where the information gain signified when it was *"clear that data has been lost. For example [in a Ted Talk (topic: Ukraine War)], the summary stated: 'Sweden and Finland did not send arms to Ukraine in the Cold War' while the longer summary stated: 'You even see countries like Finland and Sweden sending arms to Ukraine and closing their airspace, They didn't even do it in the Cold War, It's really amazing to see it.' providing me [P9] with additional relevant content."* Interestingly, P7 arrived at a similar conclusion but with a different information cue; P7 noted that if *"if the quality of a summary is supposed low, [they] would listen more carefully [to the corresponding original audio portion]"*. Meanwhile, for (ii), P9 interpreted a low information gain for a *Short Summary* as a cue to stop exploring, noting that *the estimated information gain [was] a useful tool for helping me decide if [I] wanted to read the original transcript or not. It's like a circuit breaker that saves me time from reading redundant information.*

5.5.3 User Survey Feedback

In the SUS usability survey, users stated that $\texttt{UI}_{base}$ narrowly met the usability threshold with a score of 69.2 while $\texttt{UI}_{cues}$ improved the SUS score to 75 and into the approximate "Good" territory. Table 5.6 gives the individual breakdowns of the survey. Encouragingly, users noted that the addition of visual information cues resulted in a higher likelihood of usage (Q1, $p = 0.047$) and that they were well integrated (Q5, $p = 0.018$). Moreover, participant feedback indicates $\texttt{UI}_{cues}$ either remained constant or did **not** necessarily result in a statistically significant increases in negative (even) questions for the following: unnecessary complexity (+0.28, Q2), inconsistency (-0.14, Q6), or additional learning curves (+0.42, Q4; +0.28, Q10).

Table 5.6: System Usability Score (SUS) breakdown of $\texttt{UI}_{cues}$ and $\texttt{UI}_{baseline}$; scores range from 1 (Strong Disagree) to 5 (Strong Agree). **Note: when the questions refer to "system" in $\texttt{UI}_{baseline}$, it is explicitly referring to the condition of adding visual information cues.** Participant feedback indicates increased enthusiasm for $\texttt{UI}_{cues}$ without additional confusion and complexity.

Index	Question	$\texttt{UI}_{baseline}$ Score	$\texttt{UI}_{cues}$ Score	p-value
1	I would use the system frequently	2.43	3.29	**0.047**
2	System is unnecessarily complex	1.57	1.85	0.56
3	System is easy to use	3.85	4.28	0.522
4	I would need technical support to use the system	1.14	1.57	0.61
5	System functions were well integrated	3.28	4.43	**0.018**
6	System has too much inconsistency	2.14	2	0.56
7	Ease of learning for most people	3.71	3.42	0.56
8	System is very cumbersome to use	2.28	1.71	0.48
9	Felt confident in using the system	3.28	3.71	0.44
10	Needed to learn a lot before system use	1.71	2	0.52

5.6 Discussion

Explainability and Trust in Summarization. Abstractive summarization language models, while powerful, are still opaque black boxes when presented alone to the user. From reading a summary, users cannot interpret a summarization model's rationale or decisions on how it distilled information. This lack of understanding may lead to potentially catastrophic scenarios: a reader could be unaware that important information was omitted as an abstractive summary can misrepresent the content from the original source passage. This is an important consideration since the ability to decipher a language model's decisions promotes transparency and accountability of the system, ultimately driving users to trust the outputs they are consuming.

With visual heuristic cues providing *some* information scent for summarization quality and information compression, the user is able to obtain a working intuition of the underlying summarization models performance. As seen in Section 5.5.2, participants utilized these cues to determine when to drill more deeply into their current content exploration path or when to "circuit break" their information foraging and move on to subsequent segments. Participants also leveraged the estimates of a summary's quality to know when to be more vigilant of possible AI gaffes. However, it is important to note the intrinsic design bias in the algorithms and heuristics behind our system's information cues. Assumptions and simplifications can lead to unintended consequences and potentially misleading users, as noted by P1's rationalization in Section 5.5.2. Lastly, the navigable hierarchy of summary segments presented alongside original transcripts and audio segments further provides insight towards a summarization system's thought process by allowing users the ability to see the intermediate steps, akin to "showing your work" for a problem.

The visual heuristic cues employed in this system codify basic properties of AI summarization: summary quality and information compression. Our interface and evaluation therefore serve as a proof of concept that such heuristics embedded alongside hierarchical summaries can provide utility in this medium, suggesting future work may be successful in discovering more effective heuristics and visualizations for automatic summarization transparency.

User-Tailored Summarization. A notable consideration when discussing abstractive summarization is its user-dependant nature and how different users may prefer not only different content but also varying levels of detail in their own summaries. For example, consider how a subject matter expert's ideal summary may prefer far greater detail and a different content selection than that of a lay person's. In order to create an effective summarization system for content dense long-form audio content, we must address the subjective nature of user's expectations in this context; no single summary is a one size fits all.

Previous work has explored improving the flexibility of information selected by state of the art summarization models by introducing hybrid language frameworks which pair customizable extractors with abstractors in an effort to offer more granular and explainable control of the extracted information [214]. Such approaches attempt to address the problem from the model design front as opposed to an interface and processing side. Our solution takes the latter approach providing an interface which allows users agency to select their preferred level of detail and consume the information accordingly. This approach, coupled with training agnostic post processing solutions, decouples achieving user tailored summarization from specific and complex language model instances. It should be noted, however, that solutions addressing this issue from either side are not mutually exclusive and future work should explore the efficacy of employing both approaches simultaneously.

5.7 Conclusion

This Chapter presents a novel system for efficient navigation and consumption of longform spoken dialogue by incorporating a series of training agnostic language post processing steps with an explainable and navigable hierarchical summary interface that surfaces a text representation of audio content alongside visual summary heuristic cues to the user.

Although, previous works have also leveraged hierarchical and abstractive summarization models to tackle this medium they are susceptible to three key challenges of abstractive summarization: readability, accuracy, and adequacy. Intrinsic challenges ranging from ASR errors to model hallucinations all work to lower overall summary quality. The proposed system addresses these issues providing a better foundation for leveraging hierarchical summarization as an improved medium for consuming long form audio. Critically, our system showcases the ability of these training agnostic post-processing solutions to take an off-the shelf state of the art abstractive summarization model and apply them effectively to the audio domain without additional training or custom datasets. Furthermore, the system showcases how hierarchical summaries in particular coupled with visual heuristic cues provides a novel level of browsability and explainability in an AI based system targeting this domain. Both qualitative and quantitative evaluations show our system achieves statistically significant improvement over previous hierarchical summarization interfaces as well as state-of-the-art baseline summarization models.

Chapter 6: Conclusion and Future Work

6.1 Restatement of of Thesis and Contributions

To review, this dissertation proposed using HCI principles of information foraging to design a system that leverages automatic summarization to structure longform audio transcripts and create information cues to enable navigating and browsing of longform spoken dialog. The contributions of this dissertation are as follows:

1. *Chapter 2 (Theory and Background).* We apply the HCI concept of information foraging to longform speech, enabling people to browse and navigate information in podcasts, interviews, panels, and meetings.

2. *Chapter 3 (System 1).* We accordingly introduce an approach that recursively applies automatic summarization to create hierarchical summaries, thereby condensing longform dialog to help users to 1) skim (browse) audio and 2) navigate and drill down into interesting sections to read full details.

3. *Chapter 4 (Dataset Collection).* We created a human annotated hierarchical dataset to quantitatively evaluate the effectiveness of our system's performance and find that users prefer our system in navigating and browsing content.

4. *Chapter 5 (System 2).* We introduced a suite of dialog oriented processing optimizations to improve the user experience of summaries: enhanced readability and fluency of short summaries through better topic chunking and pronoun imputation, and reliable indication of semantic coverage within short summaries to help direct navigation towards interesting information.

6.2 Limitations

As with all systems, our speech browsing and navigation tool has limitations in both its technical design and usage.

6.2.1 Technical Limitations

1. *Automatic Summarization Performance Concerns.* While abstractive summarization has markedly improved through deep neural architectures and increased summarization data availability, it is still important to emphasize the abstractive component of such summarization systems. Because the model generates an entirely new output (compare this against extractive summarization where direct verbatim passage components are taken), it is possible to generate a summary where its content is not consistent with the original source text. Combined with a challenging and noisy domain of speech, this is even more problematic. Though System 2 remedies some of these concerns, it is impossible to ensure perfect summaries; as such, systems should design around such imperfections and facilitate user recovery.

2. *Text to Speech (TTS) Performance Concerns.* Similar to language modeling concerns, ASR TTS model performance also present challenges towards our longform speech navigation system(s). Recall how ASR is the initial starting point of the entire system pipeline; it follows how initial errors such as word recognition and improper segmentation bounds already detrimentally bias the subsequent downstream summarization model. While outside the scope of our work, we do note that improvements to TTS will be similarly mirrored through higher quality summaries.

3. *Speech Domain.* Speech summarization is particularly challenging when speakers reference ambiguous objects without proper introduction. Such instances include referencing world knowledge (i.e. current events and facts), local knowledge (speaker dependent context), and deictic references (entity a speaker is directly pointing at). When summarizing a text with such incomplete background, it is possible to still generate a grammatically correct, consis-

tent, and coherent summary. However, it will still be confusing to a reader. For clarity, we say *background* here as opposed to *context*. Background refers to worldly knowledge while context refers to relevant information still contained in the source text. Issues stemming from context are from language modeling imperfections while issues from background cannot be blamed on the language model.

6.2.2 Usability Limitations

1. *Users May Miss Information While Browsing.* The goal of the visual information cues in System 2 is to help users navigate longform audio and find interesting information quickly which creates user comprehension trade-offs. Concretely, what the longform speech browsing system does is allow users to jump around, bit by bit, and in turn, makes it probable that users will gloss over information and still be unaware of doing so. While we have designed indicators to try to estimate the amount of information users have already encountered in System 2 to provide more informed browsing, this still remains a challenging problem balancing user value propositions of time spent and knowledge gained.

2. *Incorrect User Comprehension of Underlying Content.* In addition to overlooking information, users may be misled by the system if output summaries contain errors that users do not detect. Typically errors are often obvious due to the surrounding context or the reader's prior knowledge of the topic. However, subtle non-obvious errors may go unnoticed, obscuring system inaccuracies to users. This can lead to misunderstandings as users may not realize a correction is needed to properly understand the underlying information. Our evaluation did not address this limitation. An important next step is to study whether or not the summarization system contains these subtle errors, and if so, how frequent they are. Depending on the nature and severity of such errors, systems could contextually reason entire dialog transcripts (as opposed to individual local semantic segment) or query outside information to check summary adequacy.

3. *Loss of Vocal Information Cues.* By converting audio into a transcript, there is a loss of emotion, tone, and prosody found in speech. Intonation, pauses, and inflections often provide additional information and meaning beyond the underlying spoken words. For example, sarcasm may be immediately apparent in audio but it is difficult to detect, much less preserve, in a text form, especially when summarized hierarchically. Although our interface still provides the original audio, most users opted to only consume the summary text representation as reading is more efficient than listening. As is currently in the system, if the user does not elect to listen to the audio file, they would lose these information cues.

6.3 Future Work

1. *Domain extension.* The system is designed for two-person audio dialogues and performs particularly well for interviews where one person is creating a majority of the information content and narrative. However, there are many other types of speech and audio that future work could extend to: political debates, city council meetings, project discussions, doctors appointments, quarterly earnings calls, sporting events, documentaries, and instructions as they all typically share an information dense characteristic. Extending this approach to these areas has many technical challenges including connecting audio to video, increasing ASR accuracy for multi-person audio, and tracing multiple threads of conversation across people.

   Currently, these audio sources remain relatively free of external references. Compared to an audio source such as a calculus lecture that contains many references to an external source such as a whiteboard with equations, it is extremely likely that the system would not work well here. We note that that the system has fundamental limitations of being unimodal and that it cannot incorporate visual information. As a result, proposed domains have to contain knowledge that is primarily present in the audio file.

   In future work, the existing system could explore how multimodal models can jointly process image and other such references to provide adequate context to users. Naively, some forms of data such as images and video can be aligned with the longform audio's high level summaries and presented as a reference (similar to [36]) to provide the user with a more complete context. In a more involved approach, future work could explore ways to *encode* such information and provide additional context to language models. VisualBERT [215] provides an example as to how a text and language model can be employed to also gain an understanding of aligned images and text. Technical modeling can adapt this as an encoder basis with a subsequent possibility of improved contextual text generation.

2. *External knowledge.* Clearly in speech, background is often missing in a conversation, as it is often assumed. This future work direction is similar in the previous point of providing additional useful context. However, the distinction for this aspect of future work is to provide models with missing *worldly external knowledge*. Consider an audio transcript that is discussing a recent event such as the "War in Ukraine". A language model that was trained on crawled web data from a year prior to this event would not have enough context to properly comprehend specific news references; a speaker referencing the "Bridge Attack" is referring to a specific event, whereas a language model can construe "bridge attack" simply at face value. Future work can expand on leveraging external information in a search and retrieve manner; this is remniscient if open domain question answering [216] where language models attempt to retrieve arbitrary documents from an index corpus.

3. *Dataset and Training Based Methods.* As newer abstractive dialog summarization datasets become more widely available, we could further explore technical training based approaches in form of novel objective functions for speech specific hallucination robustness and architectures designed for longform text generation. In a similar vein, future work could explore augmenting training samples with poor ASR examples to improve robustness towards speech recognition errors. Currently, the dataset provided in Chapter 4 is only sufficient as a test set, and does not yet have enough data to be used as a training dataset; direct subsequent work involves obtaining more summary annotations. Other direct subsequent work include training baselines and data validation (quality assessment). Specifically, future work will involve

evaluating both lexical quality and content selection between annotators. In a broader sense, future work can leverage this dataset as a way of providing additional intermediate text level supervision for longform summarization models.

4. *Incorporating Higher Text Summary Levels for Hierarchical Summarization Refinement.* Future work can explore how different summary levels can leverage other summary levels to improve summarization quality. Consider an example containing a Long, Medium, and Short summary. After the hierarchy of summaries is initially generated, how might a Medium summary leverage the Short summary's existence to improve its summarization quality? One important aspect to consider is the logical content coherence between summary levels; in other words, if a Short summary is discussing a topic, then Medium summaries should elaborate on said topic in more detail. Sometimes, similar phrases on a topic are regurgitated in the text generation process across different levels, providing no useful additional amounts of detail, despite increasing in length.

   Future work can explore ways to improve distinctions between levels in the text generation process. For example, Medium summaries now have access (a new prior) to the topics discussed in the Short summary. In this sense, it would be possible to re-generate or rewrite (where needed) Medium summaries to include relevant details of topics in a Short summary. Some relevant works have proposed models "grounding" the abstractive summarization process [107] by extracting portions that may be more relevant and increasing their importance in subsequent text generation. With this in mind, one proposed future work direction can investigate how to encode higher level summaries as priors (along with the input text) to rewrite a given (current level) summary. Theoretically, improving the summarization language model in this facet would lead to improved utility and cohesiveness of intermediate level summaries.

5. *Encoding Auditory Information.* In conversation, intonation, pauses, and inflections give additional information and meaning towards the underlying content. Currently, our system

does not address retaining this form of information. This behavior is in line with our expectations as the audio component is seen as a last resort intended for recovering from ASR errors or model hallucinations. However, future work should explore incorporating models [217, 218] for translating these forms of vocal information and conveying them to readers. Similar to how visual information cues are currently presented in the system, future work could explore how to convey this information to users. By adding a separate language model or service[1] that is trained on a task such as emotion detection, it may be possible to identify relevant parts of a transcript that may be said with different intents. For example, a part of audio file may be said very quickly and enthusiastically, revealing the speaker's personal bias on the subject matter.

6. *Tailored Summarization and User Profiles.* A more difficult challenge would be to make a system that helps users find *all* interesting information if a user so desires. In a sense, this would measure the precision of the system. By design of the longform speech browsing system, users (ideally and typically) will not encounter all the content in an audio file, and by definition would "miss" information - either relevant (bad) or irrelevant (good). Thus, a more tractable reformulation of the problem would be to optimize how content is presented and to *tailor it at the user level.*

   First it is important to note that summarization has a user-dependant nature; different users may prefer not only different content but also varying levels of detail in their own summaries. Consider how a subject matter expert's ideal summary may not require having introductory material while a lay person's preferred summary would include such content in order to be useful. By developing user profiles that model user interests and expertise, content can be filtered and restructured in a manner that suits an individual's specific browsing needs. Such user profiles can explore the thresholds of user behavior with regards to information need. In another tangential vein but in a language modeling perspective, future work can incorporate advances summary language modeling allowing for tailored text generation [214] as the new

[1]`https://developer.ibm.com/patterns/use-advanced-nlp-and-tone-analyser-to-extract-insigh`

backbone to our system. In tandem to user profile modeling, the system's core language model can take into consider a user's priors (such as their interests) at the summary text generation stage.

7. *Visual Information Cues.* Though users found the system's visual information cues useful, they required additional work to understand. Users found there was a learning curve in order to build an intuition and mental model of the provided indicators; in our experiments, we found that individuals first had to build a conceptual understanding for quantifying both information compression and summary quality. For example, a person first had to gain a sense of the ranges of information compression and summary quality (what is a 60% on a 0% to 100% scale?) by working through examples.

   With more extensive experimentation and understanding different users' sensemaking, we could further iterate and refine how the system presents additional visual information cues towards users. Instead of a user needing to interpret 60%, the system could bucket percentages into meaningful categories which reflect people's existing notions of "quality" and "information retained". 60% could be “much information retained” while 20% could be "little information retained". Another possible visual representation worth exploring could be with "stars"; users innately know "5-stars" is the maximum a metric could achieve in terms of "goodness" whereas "3 out of 5 stars" is still somewhat decent. Regardless of the representation chosen, future work can iterate on finding the most effective visual encoding that naturally maps to people's intuitive notions of quality and can still be easily perceived and interpreted.

8. *Prolonged User Study.* The next logical step is evaluating our system in a longitudinal study in an extended practice studying user engagement with information cues and the system's performance on different domains. As with all systems, some of the longform speech browsing system's quirks and unforeseen bugs can only emerge after extended use. With the deployed instance of the system, future work can collect user engagement data, providing

insight as to how users spend time with the system. Some questions that can be addressed include: (i) Where do users spend most of their time reading (and compared to the estimated information estimation heuristics)? (ii) When do users elect to listen the original audio? Such data invariably will help future iterations on the system.

In another prolonged study, future work can technically evaluate how well the heuristics driving the visual information cues match what users intuitively expect. Does the choice of counting language objects (ROUGE-1) in 5.3 (as opposed to a different N-gram) recall affect the heuristic enough to alter how users interpret the information cue? Another question to investigate is the differences between the thresholds users have when quantifying and interpreting information cues. By having insights towards these questions, future work can refine the system accordingly.

6.4 Conclusion

6.4.1 High-Level Takeaways for HCI from NLP

Technologies in aiding 'human' understanding of language have drastically improved in recent years and have seen widespread adoption in various domains and tasks in the form of large scale transformer language models. One important consideration for HCI practitioners looking to develop new text based systems and applications is to recognize the break-neck pace at which language models are advancing. These language models require little to no additional modification nor training data in order to achieve decent off the shelf performance. Moreover, existing popular frameworks make them easy to adapt and use. As a result, these new and powerful models present a unique and modular opportunity to impact user interactions.

Natural language modeling can be viewed as providing a technical toolbox while HCI concepts and theories compose them into useful and meaningful applications for people. Typically language models are trained to solve a targeted and specific natural language task (such as sentiment classification, translation, or summarization); these tasks frequently exist as a subproblem within a more involved HCI application which these models can then by directly applied to. Some of these subproblems that were once extremely human intensive and prohibitive, can now be solved at a degree that matches or exceeds a human's capability in a cheap and efficient manner. For example, Soylent [219] which required Mechanical Turkers to edits and shorten parts of an input text, can now be automated with summarization and grammar quality classification language models. This in turn eliminates the dependency on crowdsourcing and allow the system to be deployed on a previously thought impossible scale. Though past HCI systems were lacking in raw computational power, they have already presciently addressed how such simple subproblems can be composed together within a system to solve a more complex challenge.

The powerful language modeling capabilities now possible from these language models enable novel end user applications and use cases. For example, recent work [220] has explored applying the powerful generative ability of large language models in a constrained manner to aid users in creative writing. Whereas past language models simply did not have the semantic understanding to scale to the degree required to make useful creative suggestions to users, current advances in modeling now allow for useful AI creativity tools.

6.4.2 High-Level Takeaways for NLP from HCI

Current research is heavily focused on metric performance measures such as task accuracy or text generation fidelity, and little work is done on assessing these models and their utility in different interaction scenarios. Obviously, improving a language model's performance is important as it translates to an improved experience towards users. However, understanding the different needs of how different end-users will use such language models can better help refine how different language tasks are posed. In the example of neural machine text quality classification [221], different users can tolerate different levels of quality in a translation. An individual requiring a textbook to be translated can tolerate no errors while an individual reading a Facebook post's translation can tolerate some, so long as the individual can still understand the content. From this example, we can observe how different user needs dictate the requirements for a model's (or two models') sensitivities and subsequent training.

Another HCI perspective to motivate future work in NLP language model development is to consider an interaction angle. Current AI systems that claim "human interaction" do not afford users the ability to truly dictate what their desired result is. Consider an example with Dall-E; a user only has two points of interaction with the AI system: (i) specify what they wish to have generated and (ii) decide whether or not the generation is acceptable. As is, such large scale systems do not provide users with fine-grained control to influence the end result. In the previous example, the user is relegated to "guessimating" how to change the input text prompt. Now from an interaction perspective, allowing the user to specify to the model what dimensions (such as style or color) were good and that the model should keep in subsequent generations, would result in a more effective interactions. On this technical note, by developing models that allow users to provide input and feedback at multiple points *specifically* at inference time, AI models can be tailored and adapted towards different users on demand.

6.4.3 Closing Remarks

One of the key themes this dissertation explores how we applied information foraging towards structuring longform spoken dialog and thus allowing users to subsequently browse and navigate content. It is important to emphasize that the systems presented in this dissertation are an abstract framework: while the technical procedures we have employed are currently state of the art, such as the abstractive summarization models and coreference resolution methods used, they can easily be substituted out for newer (and not yet developed) methods and applied in the same manner. In that regard, the resulting system would still remain identical, albeit with improved performance.

Given the mass adoption of video and audio for communication, developing tools for automated abstractive summarization from audio represents a valuable opportunity to re-invent how we consume such information-rich mediums. In particular, with new technologies, it is possible to browse and skim audio content on the web, similar to how we forage for visual information clues on web-pages. This thesis developed an application that utilizes automatic summarization and speech modeling to structure longform dialog to present information in a manner that is both intuitive and flexible towards different user browsing needs.

References

[1] J. Foote, “An overview of audio information retrieval,” *Multimedia systems*, vol. 7, no. 1, pp. 2–10, 1999.

[2] M.-M. Bouamrane and S. Luz, “Meeting browsing,” *Multimedia Systems*, vol. 12, no. 4, pp. 439–457, 2007.

[3] P. Lison and R. Meena, “Spoken dialogue systems: The new frontier in human-computer interaction,” *XRDS: Crossroads, The ACM Magazine for Students*, vol. 21, no. 1, pp. 46–51, 2014.

[4] P. Pirolli and S. Card, “Information foraging.,” *Psychological review*, vol. 106, no. 4, p. 643, 1999.

[5] P. Pirolli and W.-T. Fu, “Snif-act: A model of information foraging on the world wide web,” in *International Conference on User Modeling*, Springer, 2003, pp. 45–54.

[6] P. Pirolli, S. K. Card, and M. M. Van Der Wege, “Visual information foraging in a focus+ context visualization,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2001, pp. 506–513.

[7] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out*, 2004, pp. 74–81.

[8] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[9] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *arXiv preprint arXiv:1808.06226*, 2018.

[10] M. Purver, “Topic segmentation,” *Spoken language understanding: systems for extracting semantic information from speech*, pp. 291–317, 2011.

[11] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[12] A. Kittur, A. M. Peters, A. Diriye, T. Telang, and M. R. Bove, “Costs and benefits of structured information foraging,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2989–2998.

[13] P. Pirolli, "A theory of information scent," *Human-computer interaction*, vol. 1, pp. 213–217, 2003.

[14] K. E. Weick, K. M. Sutcliffe, and D. Obstfeld, "Organizing and the process of sensemaking," *Organization science*, vol. 16, no. 4, pp. 409–421, 2005.

[15] P. Pirolli, "Rational analyses of information foraging on the web," in *Cognitive science*, Routledge, 2013, pp. 343–373.

[16] P. Pirolli and S. Card, "Information foraging in information access environments," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 51–58.

[17] S. McDonald and R. J. Stevenson, "Disorientation in hypertext: The effects of three text structures on navigation performance," *Applied ergonomics*, vol. 27, no. 1, pp. 61–68, 1996.

[18] B. Arons, "Speechskimmer: A system for interactively skimming recorded speech," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 4, no. 1, pp. 3–38, 1997.

[19] J. Hirschberg, S. Whittaker, D. Hindle, F. Pereira, and A. Singhal, "Finding information in audio: A new paradigm for audio browsing/retrival," in *ESCA Tutorial and Research Workshop (ETRW) on Accessing Information in Spoken Audio*, 1999.

[20] J. Besser, K. Hofmann, M. A. Larson, *et al.*, "An exploratory study of user goals and strategies in podcast search.," in *LWA*, 2008, pp. 27–34.

[21] M. Tsagkias, M. Larson, and M. d. Rijke, "Exploiting surface features for the prediction of podcast preference," in *European Conference on Information Retrieval*, Springer, 2009, pp. 473–484.

[22] M. Tsagkias, M. Larson, and M. De Rijke, "Predicting podcast preference: An analysis framework and its application," *Journal of the American Society for information Science and Technology*, vol. 61, no. 2, pp. 374–391, 2010.

[23] H. Li, B. Jou, J. G. Ellis, D. Morozoff, and S.-F. Chang, "News rover: Exploring topical structures and serendipity in heterogeneous multimedia news," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 449–450.

[24] Y. Shi, C. Bryan, S. Bhamidipati, Y. Zhao, Y. Zhang, and K.-L. Ma, "Meetingvis: Visual narratives to assist in recalling meeting context and content," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 1918–1929, 2018.

[25] S. Lim *et al.*, "A narrative topic map visualization to summarize and recall a meeting," in *2019 IEEE Visualization Conference (VIS)*, 2019.

[26] K. Kusama and T. Itoh, “Abstract picture generation and zooming user interface for intuitive music browsing,” *Multimedia tools and applications*, vol. 73, no. 2, pp. 995–1010, 2014.

[27] M. G. Christel, “Evaluation and user studies with respect to video summarization and browsing,” *Multimedia Content Analysis, Management, and Retrieval 2006*, vol. 6073, pp. 196–210, 2006.

[28] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein, “Video tapestries with continuous temporal zoom,” in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–9.

[29] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz, “Schematic storyboarding for video visualization and editing,” *Acm transactions on graphics (tog)*, vol. 25, no. 3, pp. 862–871, 2006.

[30] D. Jackson, J. Nicholson, G. Stoeckigt, R. Wrobel, A. Thieme, and P. Olivier, “Panopticon: A parallel video overview system,” in *proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 123–130.

[31] L. He, E. Sanocki, A. Gupta, and J. Grudin, “Auto-summarization of audio-video presentations,” in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, 1999, pp. 489–498.

[32] Z. Elkhattabi, Y. Tabii, and A. Benkaddour, “Video summarization: Techniques and applications,” *International Journal of Computer and Information Engineering*, vol. 9, no. 4, pp. 928–933, 2015.

[33] N. Vasconcelos and A. Lippman, “Bayesian modeling of video editing and structure: Semantic features for video summarization and browsing,” in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, IEEE, 1998, pp. 153–157.

[34] B. Yu, W.-Y. Ma, K. Nahrstedt, and H.-J. Zhang, “Video summarization based on user log enhanced link analysis,” in *Proceedings of the eleventh ACM international conference on Multimedia*, 2003, pp. 382–391.

[35] S. Srinivasan, D. Ponceleon, A. Amir, and D. Petkovic, “" what is in that video anyway?": In search of better browsing,” in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, IEEE, vol. 1, 1999, pp. 388–393.

[36] A. Pavel, D. B. Goldman, B. Hartmann, and M. Agrawala, “Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries,” in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST ’15, Charlotte, NC, USA: Association for Computing Machinery, 2015, 181–190, ISBN: 9781450337793.

[37] D. Rotman, Y. Yaroker, E. Amrani, U. Barzelay, and R. Ben-Ari, “Learnable optimal sequential grouping for video scene detection,” in *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020, 1958–1966, ISBN: 9781450379885.

[38] A. Truong, P. Chi, D. Salesin, I. Essa, and M. Agrawala, “Automatic generation of two-level hierarchical tutorials from instructional makeup videos,” 2021.

[39] J. Matejka, T. Grossman, and G. Fitzmaurice, “Swift: Reducing the effects of latency in online video scrubbing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 637–646.

[40] J. Matejka, T. Grossman, and G. Fitzmaurice, “Swifter: Improved online video scrubbing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 1159–1168.

[41] K. Krishna, S. Khosla, J. P. Bigham, and Z. C. Lipton, “Generating soap notes from doctor-patient conversations using modular summarization techniques,” *arXiv preprint arXiv:2005.01795*, 2020.

[42] S. Tian, “Integrating discussion and summarization in collaborative writing,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA ’20, Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–6, ISBN: 9781450368193.

[43] K. K. Nam and M. S. Ackerman, “Arkose: Reusing informal information from online discussions,” in *GROUP ’07*, 2007.

[44] S. Gehrmann, Y. Deng, and A. M. Rush, “Bottom-up abstractive summarization,” *arXiv preprint arXiv:1808.10792*, 2018.

[45] A. Gatt and E. Krahmer, “Survey of the state of the art in natural language generation: Core tasks, applications and evaluation,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[46] J. Zhu, H. Li, T. Liu, Y. Zhou, J. Zhang, and C. Zong, “Msmo: Multimodal summarization with multimodal output,” in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 4154–4164.

[47] B. Wang, G. Li, X. Zhou, Z. Chen, T. Grossman, and Y. Li, “Screen2words: Automatic mobile ui summarization with multimodal learning,” in *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021, pp. 498–510.

[48] I. Avellino, S. Nozari, G. Canlorbe, and Y. Jansen, “Surgical video summarization: Multifarious uses, summarization process and ad-hoc coordination,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–23, 2021.

[49] N. UzZaman, J. P. Bigham, and J. F. Allen, “Multimodal summarization of complex sentences,” in *Proceedings of the 16th international conference on Intelligent user interfaces*, 2011, pp. 43–52.

[50] D. O’Shaughnessy, “Automatic speech recognition: History, methods and challenges,” *Pattern Recognition*, vol. 41, no. 10, pp. 2965–2979, 2008.

[51] W. Ghai and N. Singh, “Literature review on automatic speech recognition,” *International Journal of Computer Applications*, vol. 41, no. 8, 2012.

[52] M. Benzeghiba *et al.*, “Automatic speech recognition and speech variability: A review,” *Speech communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[53] Google, *Speech-to-text*, 2021.

[54] K. Zechner, “Automatic summarization of open-domain multiparty dialogues in diverse genres,” *Computational Linguistics*, vol. 28, no. 4, pp. 447–485, 2002.

[55] A. Gravano, M. Jansche, and M. Bacchiani, “Restoring punctuation and capitalization in transcribed speech,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 4741–4744.

[56] T. Alam, A. Khan, and F. Alam, “Punctuation restoration using transformer models for high-and low-resource languages,” in *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 132–142.

[57] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, “Speaker diarization with lstm,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5239–5243.

[58] Z. Chen, L. Chen, Z. Xu, Y. Zhao, S. Zhu, and K. Yu, “Credit: Coarse-to-fine sequence generation for dialogue state tracking,” *arXiv preprint arXiv:2009.10435*, 2020.

[59] M. Bacchiani *et al.*, “Scanmail: Audio navigation in the voicemail domain,” in *Proceedings of the first international conference on Human language technology research*, 2001.

[60] I. Lopatovska *et al.*, “Talk to me: Exploring user interactions with the amazon alexa,” *Journal of Librarianship and Information Science*, vol. 51, no. 4, pp. 984–997, 2019.

[61] H. Liao, E. McDermott, and A. Senior, “Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription,” in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, IEEE, 2013, pp. 368–373.

[62] P. Donnelly, N. Blanchard, A. Olney, S. Kelly, M. Nystrand, and S. D’Mello, “Words matter: Automatic detection of teacher questions in live classroom discourse using linguistics, acoustics, and context,” Mar. 2017, pp. 218–227.

[63] B. M. L. Srivastava and S. Sitaram, “Homophone identification and merging for code-switched speech recognition.,” in *Interspeech*, 2018, pp. 1943–1947.

[64] J. Liu *et al.*, “Robustness testing of language understanding in task-oriented dialog,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 2467–2480.

[65] L. Mošner *et al.*, “Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6475–6479.

[66] I. Pak and P. L. Teh, “Text segmentation techniques: A critical review,” *Innovative Computing, Optimization and Its Applications*, pp. 167–181, 2018.

[67] K. M. Hermann *et al.*, “Teaching machines to read and comprehend,” *Advances in neural information processing systems*, vol. 28, 2015.

[68] B. Kim, H. Kim, and G. Kim, “Abstractive summarization of reddit posts with multi-level memory networks,” *arXiv preprint arXiv:1811.00783*, 2018.

[69] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, *On faithfulness and factuality in abstractive summarization*, 2020. arXiv: `2005.00661 [cs.CL]`.

[70] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, “Samsum corpus: A human-annotated dialogue dataset for abstractive summarization,” *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[71] J. Ainslie *et al.*, *Etc: Encoding long and structured inputs in transformers*, 2020. arXiv: `2004.08483 [cs.LG]`.

[72] M. Zaheer *et al.*, “Big bird: Transformers for longer sequences.,” in *NeurIPS*, 2020.

[73] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[74] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[75] S. Gupta and S. K. Gupta, "Abstractive summarization: An overview of the state of the art," *Expert Systems with Applications*, vol. 121, pp. 49–65, 2019.

[76] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[77] Z. Liu, K. Shi, and N. F. Chen, "Coreference-aware dialogue summarization," *arXiv preprint arXiv:2106.08556*, 2021.

[78] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[79] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. arXiv: 1910.10683.

[80] T. B. Brown *et al.*, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. arXiv: 2005.14165.

[81] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[82] P. Koehn, "Neural machine translation," *arXiv preprint arXiv:1709.07809*, 2017.

[83] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[84] C.-Z. A. Huang *et al.*, "Music transformer," *arXiv preprint arXiv:1809.04281*, 2018.

[85] G. Carenini and J. C. K. Cheung, "Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality," in *Proceedings of the Fifth International Natural Language Generation Conference*, 2008, pp. 33–41.

[86] R. Nallapati, B. Zhou, and M. Ma, "Classify or select: Neural architectures for extractive document summarization," *arXiv preprint arXiv:1611.04244*, 2016.

[87] T. Cohn and M. Lapata, "Sentence compression beyond word deletion," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 137–144.

[88] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," *arXiv preprint arXiv:1805.11080*, 2018.

[89] H. Behnke, M. Fomicheva, and L. Specia, “Bias mitigation in machine translation quality estimation,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1475–1487.

[90] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, “Neural text summarization: A critical evaluation,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 540–551.

[91] E. Sharma, L. Huang, Z. Hu, and L. Wang, *An entity-driven framework for abstractive summarization*, 2019.

[92] L. Huang, L. Wu, and L. Wang, “Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward,” *arXiv preprint arXiv:2005.01159*, 2020.

[93] M. Cao, Y. Dong, and J. C. K. Cheung, “Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 3340–3354.

[94] F. Nan *et al.*, *Improving factual consistency of abstractive summarization via question answering*, 2021. arXiv: `2105.04623 [cs.CL]`.

[95] M. Eyal, T. Baumel, and M. Elhadad, “Question answering as an automatic evaluation metric for news article summarization,” *arXiv preprint arXiv:1906.00318*, 2019.

[96] T. Goyal and G. Durrett, “Evaluating factuality in generation with dependency-level entailment,” *arXiv preprint arXiv:2010.05478*, 2020.

[97] A. Wang, K. Cho, and M. Lewis, “Asking and answering questions to evaluate the factual consistency of summaries,” *arXiv preprint arXiv:2004.04228*, 2020.

[98] M. Cao, Y. Dong, J. Wu, and J. C. K. Cheung, “Factual error correction for abstractive summarization models,” *arXiv preprint arXiv:2010.08712*, 2020.

[99] Y. Dong, S. Wang, Z. Gan, Y. Cheng, J. C. K. Cheung, and J. Liu, “Multi-fact correction in abstractive text summarization,” *arXiv preprint arXiv:2010.02443*, 2020.

[100] K. Zechner and A. Waibel, “Diasumm: Flexible summarization of spontaneous dialogues in unrestricted domains,” in *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, 2000.

[101] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings.," 2005.

[102] M. Khalifa, M. Ballesteros, and K. McKeown, "A bag of tricks for dialogue summarization," *arXiv preprint arXiv:2109.08232*, 2021.

[103] K. McKeown, J. Hirschberg, M. Galley, and S. Maskey, "From text to speech summarization," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, IEEE, vol. 5, 2005, pp. v–997.

[104] Y. Liu and D. Hakkani-Tür, "Speech summarization," *Spoken language understanding: Systems for extracting semantic information from speech*, pp. 357–396, 2011.

[105] S. Maskey and J. Hirschberg, "Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[106] J. Chen and D. Yang, "Simple conversational data augmentation for semi-supervised abstractive dialogue summarization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6605–6616.

[107] K. Song, C. Li, X. Wang, D. Yu, and F. Liu, "Towards abstractive grounded summarization of podcast transcripts," *arXiv preprint arXiv:2203.11425*, 2022.

[108] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[109] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: pre-training with extracted gap-sentences for abstractive summarization," *CoRR*, vol. abs/1912.08777, 2019. arXiv: `1912.08777`.

[110] Y. Zhang *et al.*, "An exploratory study on long dialogue summarization: What works and what's next," *arXiv preprint arXiv:2109.04609*, 2021.

[111] L. Bing, P. Li, Y. Liao, W. Lam, W. Guo, and R. J. Passonneau, "Abstractive multi-document summarization via phrase selection and merging," *arXiv preprint arXiv:1506.01597*, 2015.

[112] A. Gidiotis and G. Tsoumakas, "A divide-and-conquer approach to the summarization of long documents," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 3029–3040, 2020.

[113] Y. Zhang *et al.*, "Summˆ n: A multi-stage summarization framework for long input dialogues and documents," *arXiv preprint arXiv:2110.10150*, 2021.

[114] J. Wu *et al.*, “Recursively summarizing books with human feedback,” *arXiv preprint arXiv:2109.10862*, 2021.

[115] T. Brown *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[116] D. Li, T. Chen, A. Tung, and L. B. Chilton, “Hierarchical summarization for longform spoken dialog,” *CoRR*, vol. abs/2108.09597, 2021. arXiv: `2108.09597`.

[117] T. Rohde, X. Wu, and Y. Liu, “Hierarchical learning for generation with long source sequences,” *arXiv preprint arXiv:2104.07545*, 2021.

[118] S. Cao and L. Wang, “Hibrids: Attention with hierarchical biases for structure-aware long document summarization,” *arXiv preprint arXiv:2203.10741*, 2022.

[119] M. Henaff, J. Weston, A. Szlam, A. Bordes, and Y. LeCun, “Tracking the world state with recurrent entity networks,” *arXiv preprint arXiv:1612.03969*, 2016.

[120] A. Graves *et al.*, “Hybrid computing using a neural network with dynamic external memory,” *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[121] Q. Grail, J. Perez, and E. Gaussier, “Globalizing bert-based transformer architectures for long document summarization,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 1792–1810.

[122] B. Pang, E. Nijkamp, W. Kryściński, S. Savarese, Y. Zhou, and C. Xiong, “Long document summarization with top-down and bottom-up inference,” *arXiv preprint arXiv:2203.07586*, 2022.

[123] Y. Liu and M. Lapata, “Hierarchical transformers for multi-document summarization,” *arXiv preprint arXiv:1905.13164*, 2019.

[124] A. Cohan *et al.*, “A discourse-aware attention model for abstractive summarization of long documents,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 615–621.

[125] C. Zhu, R. Xu, M. Zeng, and X. Huang, “A hierarchical network for abstractive meeting summarization with cross-domain pretraining,” *arXiv preprint arXiv:2004.02016*, 2020.

[126] I. Mccowan *et al.*, “The ami meeting corpus,” in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research.*

L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology, 2005.

[127] A. Janin *et al.*, "The icsi meeting corpus," 2003, pp. 364–367.

[128] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," *CoRR*, vol. abs/1506.03340, 2015. arXiv: `1506.03340`.

[129] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," *ArXiv*, vol. abs/1808.08745, 2018.

[130] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[131] A. Cohan *et al.*, "A discourse-aware attention model for abstractive summarization of long documents," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

[132] M. Maddela, M. Kulkarni, and D. Preotiuc-Pietro, *Entsum: A data set for entity-centric summarization*, 2022.

[133] L. Huang, S. Cao, N. Parulian, H. Ji, and L. Wang, "Efficient attentions for long document summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 1419–1436.

[134] M. Zhong *et al.*, "Qmsum: A new benchmark for query-based multi-domain meeting summarization," *arXiv preprint arXiv:2104.05938*, 2021.

[135] C. Zhu, Y. Liu, J. Mei, and M. Zeng, "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*, Online: Association for Computational Linguistics, Jun. 2021, pp. 5927–5934.

[136] M. Chen, Z. Chu, S. Wiseman, and K. Gimpel, *Summscreen: A dataset for abstractive screenplay summarization*, 2021.

[137] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue dataset for abstractive summarization," *ArXiv*, vol. abs/1911.12237, 2019.

[138] Y. Chen, Y. Liu, L. Chen, and Y. Zhang, *Dialogsum: A real-life scenario dialogue summarization dataset*, 2021.

[139] M. Khalman, Y. Zhao, and M. Saleh, "ForumSum: A multi-speaker conversation summarization dataset," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4592–4599.

[140] A. Clifton *et al.*, "100,000 podcasts: A spoken English document corpus," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5903–5917.

[141] E. Lloret, L. Plaza, and A. Aker, "The challenging task of summary evaluation: An overview," *Language Resources and Evaluation*, vol. 52, no. 1, pp. 101–148, 2018.

[142] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: learning robust metrics for text generation," *CoRR*, vol. abs/2004.04696, 2020. arXiv: `2004.04696`.

[143] C. Chelba, J. Zhou, Y. Li, H. Kazawa, J. Klingner, and M. Niu, "Data troubles in sentence level confidence estimation for machine translation," *CoRR*, vol. abs/2010.13856, 2020. arXiv: `2010.13856`.

[144] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," *CoRR*, vol. abs/2006.14799, 2020. arXiv: `2006.14799`.

[145] A. Nenkova and R. J. Passonneau, "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*, 2004, pp. 145–152.

[146] A. Nenkova, R. Passonneau, and K. McKeown, "The pyramid method: Incorporating human content selection variation in summarization evaluation," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 4, no. 2, 4–es, 2007.

[147] C. Greenbacker, "Towards a framework for abstractive summarization of multimodal documents," in *Proceedings of the ACL 2011 Student Session*, 2011, pp. 75–80.

[148] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, *Deep communicating agents for abstractive summarization*, 2018.

[149] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," *arXiv preprint arXiv:1910.12840*, 2019.

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

[151] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[152] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," *arXiv preprint arXiv:1908.08960*, 2019.

[153] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[154] S. Banerjee and A. Lavie, "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*, 2005, pp. 65–72.

[155] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: `1904.09675 [cs.CL]`.

[156] W. Yuan, G. Neubig, and P. Liu, "Bartscore: Evaluating generated text as text generation," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 27 263–27 277.

[157] Y. Gao, W. Zhao, and S. Eger, "SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1347–1354.

[158] H. Wu, T. Ma, L. Wu, T. Manyumwa, and S. Ji, "Unsupervised reference-free summary quality evaluation via contrastive learning," *CoRR*, vol. abs/2010.01781, 2020. arXiv: `2010.01781`.

[159] H. Shimanaka, T. Kajiwara, and M. Komachi, "RUSE: Regressor using sentence embeddings for automatic machine translation evaluation," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 751–758.

[160] S. Agrawal, G. F. Foster, M. Freitag, and C. Cherry, "Assessing reference-free peer evaluation for machine translation," *CoRR*, vol. abs/2104.05146, 2021. arXiv: `2104.05146`.

[161] A. Pavel, C. Reed, B. Hartmann, and M. Agrawala, “Video digests: A browsable, skimmable format for informational lecture videos.,” in *UIST*, Citeseer, vol. 10, 2014, pp. 2 642 918–2 647 400.

[162] A. Pavel, D. B. Goldman, B. Hartmann, and M. Agrawala, “Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries,” in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 2015, pp. 181–190.

[163] H. Xia, “Crosspower: Bridging graphics and linguistics,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 722–734.

[164] H. Xia, J. Jacobs, and M. Agrawala, “Crosscast: Adding visuals to audio travel podcasts,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 735–746.

[165] M. Li, L. Zhang, H. Ji, and R. J. Radke, “Keep meeting summaries on topic: Abstractive multi-modal meeting summarization,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2190–2196.

[166] A. Vartakavi and A. Garg, “Podsumm–podcast audio summarization,” *arXiv preprint arXiv:2009.10315*, 2020.

[167] H. Karlbom and A. Clifton, “Abstractive podcast summarization using bart with longformer attention,” 2020.

[168] C. Zheng, K. Zhang, H. J. Wang, and L. Fan, *A two-phase approach for abstractive podcast summarization*, 2020. arXiv: `2011.08291 [cs.CL]`.

[169] J. Xu, Z. Gan, Y. Cheng, and J. Liu, “Discourse-aware neural extractive text summarization,” *arXiv preprint arXiv:1910.14142*, 2019.

[170] G. Shang *et al.*, “Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization,” *arXiv preprint arXiv:1805.05271*, 2018.

[171] A. X. Zhang, L. Verou, and D. Karger, “Wikum: Bridging discussion forums and wikis using recursive summarization,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 2082–2096.

[172] T. Wolf *et al.*, “Huggingface’s transformers: State-of-the-art natural language processing,” *CoRR*, vol. abs/1910.03771, 2019. arXiv: `1910.03771`.

[173] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *CoRR*, vol. abs/1908.10084, 2019. arXiv: `1908.10084`.

[174] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in english words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019.

[175] J. Y. Kim *et al.*, *A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech*, 2019. arXiv: `1904.12403 [cs.SD]`.

[176] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th Web for All Conference*, 2016, pp. 1–8.

[177] Z. Zhao, S. B. Cohen, and B. Webber, "Reducing quantity hallucinations in abstractive summarization," *arXiv preprint arXiv:2009.13312*, 2020.

[178] P. Pirolli, *Information foraging theory : adaptive interaction with information / Peter Pirolli.* Oxford University Press New York, 2007, 204 p. : ISBN: 9780195173321 0195173325.

[179] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[180] R. Takanobu *et al.*, "A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning.," in *IJCAI*, 2018, pp. 4403–4410.

[181] V. Stoyanov and C. Cardie, "Partially supervised coreference resolution for opinion summarization through structured rule learning," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 336–344.

[182] R. Witte and S. Bergler, "Fuzzy coreference resolution for summarization," in *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, 2003, pp. 43–50.

[183] M. Gardner *et al.*, "Allennlp: A deep semantic natural language processing platform," *arXiv preprint arXiv:1803.07640*, 2018.

[184] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *NAACL-HLT*, 2018.

[185] D. W. Maynard, "Placement of topic changes in conversation," vol. 30, no. 3-4, pp. 263–290, 1980.

[186] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 562–569.

[187] M. P. Kaschak and A. M. Glenberg, "Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension," *Journal of memory and language*, vol. 43, no. 3, pp. 508–529, 2000.

[188] J. Christensen, S. Soderland, G. Bansal, *et al.*, "Hierarchical summarization: Scaling up multi-document summarization," in *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, 2014, pp. 902–912.

[189] S. Zhang, C. Gong, and E. Choi, "Capturing label distribution: A case study in nli," *arXiv preprint arXiv:2102.06859*, 2021.

[190] L. Qiu, J. Li, W. Bi, D. Zhao, and R. Yan, "Are training samples correlated? learning to generate dialogue responses with multiple references," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3826–3835.

[191] J. Kasai *et al.*, "Bidimensional leaderboards: Generate and evaluate language hand in hand," *arXiv preprint arXiv:2112.04139*, 2021.

[192] A. Wang, R. Y. Pang, A. Chen, J. Phang, and S. R. Bowman, "Squality: Building a long-document summarization dataset the hard way," *arXiv preprint arXiv:2205.11465*, 2022.

[193] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 121–128.

[194] T. Vodolazova and E. Lloret, "Towards adaptive text summarization: How does compression rate affect summary readability of L2 texts?" In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, Varna, Bulgaria: INCOMA Ltd., Sep. 2019, pp. 1265–1274.

[195] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Abstractive text summarization: Enhancing sequence-to-sequence models using word sense disambiguation and semantic content generalization," *Computational Linguistics*, vol. 47, no. 4, pp. 813–859, 2021.

[196] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1112–1122.

[197] M. Qi, H. Liu, Y. Fu, and T. Liu, "Improving abstractive dialogue summarization with hierarchical pretraining and topic segment," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 1121–1130.

[198] N. Stylianou and I. Vlahavas, “A neural entity coreference resolution review,” *Expert Systems with Applications*, vol. 168, p. 114 466, 2021.

[199] J. Li *et al.*, “Dadgraph: A discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[200] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, “Samsum corpus: A human-annotated dialogue dataset for abstractive summarization,” *arXiv preprint arXiv:1911.12237*, 2019.

[201] B. Cui, Y. Li, Y. Zhang, and Z. Zhang, “Text coherence analysis based on deep neural network,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2027–2030.

[202] F. C. Bartlett and F. C. Bartlett, *Remembering: A study in experimental and social psychology*. Cambridge university press, 1995.

[203] T. Goyal, J. J. Li, and G. Durrett, “Snac: Coherence error detection for narrative summarization,” *arXiv preprint arXiv:2205.09641*, 2022.

[204] A. Williams, N. Nangia, and S. Bowman, “A broad-coverage challenge corpus for sentence understanding through inference,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1112–1122.

[205] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in *COLING 2018, 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.

[206] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *arXiv preprint arXiv:1910.10683*, 2019.

[207] C. Napoles, K. Sakaguchi, and J. R. Tetreault, “JFLEG: A fluency corpus and benchmark for grammatical error correction,” *CoRR*, vol. abs/1702.04066, 2017. arXiv: `1702.04066`.

[208] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[209] A. K. Vijayakumar *et al.*, “Diverse beam search: Decoding diverse solutions from neural sequence models,” *arXiv preprint arXiv:1610.02424*, 2016.

[210] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem, “Re-evaluating automatic metrics for image captioning,” in *Proceedings of the 15th Conference of the European Chapter*

of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 199–209.

[211] K. Krippendorff, “Computing krippendorff’s alpha-reliability,” 2011.

[212] M. Lapata, R. Barzilay, *et al.*, “Automatic evaluation of text coherence: Models and representations,” in *IJCAI*, Citeseer, vol. 5, 2005, pp. 1085–1090.

[213] J. Brooke *et al.*, “Sus-a quick and dirty usability scale,” *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[214] W. Haonan, G. Yang, B. Yu, M. Lapata, and H. Heyan, “Exploring explainable selection to control abstractive summarization,” *arXiv preprint arXiv:2004.11779*, 2020.

[215] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, “Visualbert: A simple and performant baseline for vision and language,” *arXiv preprint arXiv:1908.03557*, 2019.

[216] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, *Realm: Retrieval-augmented language model pre-training*, 2020.

[217] M. R. Amer, B. Siddiquie, C. Richey, and A. Divakaran, “Emotion detection in speech using deep networks,” in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2014, pp. 3724–3728.

[218] M. S. M. Suhaimin, M. H. A. Hijazi, R. Alfred, and F. Coenen, “Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts,” in *2017 8th International conference on information technology (ICIT)*, IEEE, 2017, pp. 703–709.

[219] M. S. Bernstein *et al.*, “Soylent: A word processor with a crowd inside,” in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010, pp. 313–322.

[220] K. I. Gero, V. Liu, and L. Chilton, “Sparks: Inspiration for science writing using language models,” in *Designing Interactive Systems Conference*, 2022, pp. 1002–1019.

[221] J. Libovickỳ and J. Helcl, “End-to-end non-autoregressive neural machine translation with connectionist temporal classification,” *arXiv preprint arXiv:1811.04719*, 2018.