Improving Automatic Summarization for Browsing Longform Spoken Dialog

Daniel Li*
daniel.li@columbia.edu
Columbia University
New York, New York, USA

Thomas Chen*
chen.thomas@microsoft.com
Microsoft
Redmond, Washington, USA

Alec Zadikian
alecz@google.com
Google
Mountain View, California, USA

Albert Tung
atung3@stanford.edu
Stanford University
Palo Alto, California, USA

Lydia B. Chilton†
chilton@cs.columbia.edu
Columbia University
New York, New York, USA

ABSTRACT

Longform spoken dialog delivers rich streams of informative content through podcasts, interviews, debates, and meetings. While production of this medium has grown tremendously, spoken dialog remains challenging to consume as listening is slower than reading and difficult to skim or navigate relative to text. Recent systems leveraging automatic speech recognition (ASR) and automatic summarization allow users to better browse speech data and forage for information of interest. However, these systems intake disfluent speech which causes automatic summarization to yield readability, adequacy, and accuracy problems. To improve navigability and browsability of speech, we present three training agnostic post-processing techniques that address dialog concerns of readability, coherence, and adequacy. We integrate these improvements with user interfaces which communicate estimated summary metrics to aid user browsing heuristics. Quantitative evaluation metrics show a 19% improvement in summary quality. We discuss how summarization technologies can help people browse longform audio in trustworthy and readable ways.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

summarization, natural language interaction, automatic speech recognition, information retrieval, machine learning applications

ACM Reference Format:
Daniel Li, Thomas Chen, Alec Zadikian, Albert Tung, and Lydia B. Chilton. 2023. Improving Automatic Summarization for Browsing Longform Spoken Dialog. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3544548.3581339

*Both authors contributed equally to this research.
†Corresponding author.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3581339

1 INTRODUCTION

Longform spoken dialog is a powerful and intuitive communication medium that delivers rich diverse streams of informative content in a variety of formats such as podcasts, interviews, debates, and meetings. They convey a multitude of important topics ranging from healthcare and equity to current events, economics, and politics. Moreover, longform audio formats are only increasing in popularity and availability; consider the explosive growth of podcasts, averaging over 150,000 new podcasts year over year with hundreds of millions of listeners worldwide. However, users who are interested in a topic may be unwilling to invest long periods of time listening to the entirety of a long audio. For example, if a colleague emails you a 30 minute YouTube video about "diversity and inclusion in the workplace", you may be interested, but not able to spend 30 minutes. Instead, you might wish to browse the content to quickly find interesting nuggets of information, then decide whether to dive deeper.

Unfortunately, longform audio is difficult to skim or browse. Typical browsing strategies are problematic: skipping around in an audio player may cause users to miss important details, reading speech transcripts may prove slow and tedious due to the lack of structure compared to written articles, and listening at higher speeds saves time but is mentally taxing and may reduce comprehension of the material. Clearly, there is a need for people to be able to quickly and easily access areas of interest in longform spoken dialog without spending a lot of time.

Recently, advanced in NLP have enabled us to build systems [37] that use abstractive summarization as a tool to help people browse and navigate longform audio. The general approach is to first use automatic speech recognition to transcribe the audio into text and subsequently use automatic summarization to summarize that text. Because of current memory limitations of automatic abstractive summarization language models, longform documents cannot be summarized at once. Instead, they can be summarized recursively, forming a hierarchy. First, the transcript is broken into 256-character chunks, then each chunk is summarized. The resulting summary is re-segmented and re-summarized. The process can be done repeatedly until the desired summary length is reached. This

results in a set of top-level "short summaries" that a reader can use to browse and navigate. For example, a short summary may say "My parents invested in my first business". A user who finds this interesting may wish to read a longer and more detailed summary to discern what the first business was, how much his parents invested, etc. Thus, automatically generated summaries may be seen as an outline, providing users information cues to navigate and browse content to meet their information needs.

Although existing systems for hierarchical navigation have shown utility in browsing and navigating content without having to listen to an entire audio file, they face several challenges [41, 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* [9] - summaries that are not immediately comprehensible due to vague pronouns or incoherent grammar impact a user's ability to quickly understand content.
(2) *Accuracy* [15, 33] - summaries that are inconsistent with the source text may provide erroneous information and mislead users.
(3) *Adequacy* [30] - 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 issues, 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 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 ways:

(1) Automatic comparisons of generated summaries with gold standard summary references. Results indicate our system yields a 19% ROUGE-2 improvement over baselines.
(2) Compared to human annotated "gold" summaries, 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 *Short Summaries* were able to adequately summarize the content, that heuristics for information gain and summary quality helped build confidence in the system despite imperfect AI summaries, and helped them manage the exploration/exploitation trade off while browsing.
(4) A quantitative user study showing that people were able to complete recall and information foraging tasks and in about half the time using the system versus using traditional audio browsing.

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

2 RELATED WORKS

2.1 Automatic Speech Recognition and Summarization

Automatic Speech Recognition systems (ASR) [16, 56] are used to transcribe audio (commonly referred to as word recognition) into source language transcripts [7]. Concretely, in speech-to-text ASR systems, an audio file is a converted text transcript. Such systems have recently made relatively significant strides in terms of practical performance and have seen widespread adoption.

State of the art (SOTA) ASR [19] is no longer constrained by vocabulary and remains relatively robust, encouragingly extending word recognition to topical domains and noisy audio. SOTA systems [19] also offer a wide suite of useful services such as automatic punctuation insertion [3, 22, 79], where raw transcribed text has punctuation, capitalization and sentence endpointers (.!?) automatically inserted using another language model. Advances in audio source separation can be used to identify speakers (i.e. speaker diarization) [71] and enhance difficult-to-hear or noisy dialog. We emphasize the focus on this work is not on ASR systems; we use Google Speech-to-Text [19] ASR as a black box and the starting point for our applications.

Text summarization techniques can be classified into two categories: *abstractive* [23] and *extractive* [8]. Abstractive summarization generates [66] a new unique summary of text given a context. Contemporary summarization language models [36, 81] are based on a transformer [69] architecture. Extractive summarization selects relevant portions of the input text and concatenates the 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.

Many summarization models [28, 53, 80] address speech-induced specific technical challenges [10, 42, 46, 51]. In tandem, research is focused on addressing the longform input aspect of speech; processing long inputs is notoriously difficult due to context and model memory constraints. Current research typically adopts a hierarchical summarization approach to either model long range context or break up inputs to more tractable sizes (discussed in Section 3.1). However, naively breaking up inputs means hierarchical summarization often suffers from repeated or lost entities and temporal fragmentation. By contrast, in our work, we track entities through coreference imputation, ensure greater summary correctness, and reorder dialog for better flow.

While the popularity of abstractive summarization datasets continues to grow, especially in the speech domain, data is still difficult and prohibitively expensive to collect. Existing longform dialog datasets (i.e. AMI [50] & ICSI [27]) are primarily focused on meetings or lack gold standard sufficient summaries [12]. Conversational datasets, SamSum [17] and DialogSum [11] have dialogs that are in the hundreds of words instead of thousands. Moreover, no dataset contains intermediate hierarchical summaries, making such datasets problematic to use for structured browsing and navigating of longform audio content.

The idea of hierarchical summarization also extends to collaborative tools. Wikum+ [67] is system that recursively interleaves forum posts for people to jointly summarize, creating a summary tree. Arkose [54] is another tool that proposed merging multiple hierarchical levels as a way to structure and process unorganized online large-scale community discussions. While the focus of these tools is on collaboration, they share the concept of hierarchical structuring of information to improve user navigation.

2.2 User Browsing Behaviors

Information foraging theory [59] 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 not search linearly. Instead, people "forage" for information, similar to how animals sniff around for food, scanning from area to area. 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 [58]. Sensemaking [72] 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. Principles in information foraging and sensemaking describing how users browse and construe information are universal and provide valuable insights that can be applied towards designing a system to help users navigate and browse speech.

Various tools already exist to assist users with navigating and browsing different forms of media, primarily concerned with videos and the audiovisual domain. Asymmetrically, systems exclusively for audio and text have primarily been left relatively untouched in older work (pre-2000s). For audio and text navigation, SpeechSkimmer [4] is an early tool that lays the foundation in addressing the challenges and difficulties of navigating and browsing speech by listening to compressed audio segments and allowing users to continue listening further into more detailed segments. SCAN [26] is a prototype speech retrieval and browsing system that aims to help users navigate poor automatic transcriptions and retrieve multiple speech transcripts. Navigation systems [39, 63] 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.

Video summarization [14, 25] as means for skimming and browsing for information [68] is a popular domain and active area of research. SceneSkim [57] 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. Other work on video browsing and navigation investigate methods to enrich existing user browsing behaviors; Swift [47] and Swifter [48] tackle the challenge of real-time seeking in video scrubbing (where a user drops and drags a the playhead on a video timeline).

3 BACKGROUND ON AUTOMATIC HIERARCHICAL SUMMARIZATION

3.1 Challenges and Approaches for Summarizing Longform Text

Summarizing longform text [1, 6, 78] is an outstanding challenge in natural language processing. While massive language models are extremely promising, they require encoding the entire input simultaneously into memory. Unfortuately, longform text, especially audio transcripts, exceed their memory and size constraints. To circumvent processing large inputs directly, recent works have adopted recursive and hierarchical text processing approaches [37, 76, 77, 82] which are generally concerned with segmenting a longform input into smaller and more tractable inputs and recursively using a summarization model's output as another input to obtain increasingly shorter summaries.

While the hierarchical processing method is not only necessary for longform dialog, it also provides usability benefits. From a user's perspective, being able to understand and control the degree of summarization is immensely valuable as it empowers users to individually decide the trade-off between time spent consuming information and the thoroughness of each summary. Hierarchical summarization affords this by generating multiple summaries of varying levels of abstraction, allowing a user to read a summary with a level of detail suiting their needs. Level refers to a subsequent shorter recursive summary; the highest set of summaries are referred to as the *Short Summaries*. We adopt the hierarchical approach as the basis of our automatic summarization system.

3.2 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; stand-alone 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 [9]. 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.* ⇒ *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 [15, 33] . 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.* ⇒ *The weather will be cloudy.*

Adequacy [30]. 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 [30]. Poor adequacy example: *There will be high winds and heavy traffic on the freeway due to storm congestion.* ⇒ *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.

3.3 Dataset

To our knowledge, there are currently no readily available gold label longform spoken dialog summarization datasets, let alone any hierarchical summarization datasets containing intermediate summaries. To properly evaluate our system, we prepared a dataset of 25 transcripts consisting of audio recordings to use as test data. Transcripts were transcribed with speaker diarization using `Google Speech-to-Text`.

Table 1: Aggregate Statistics from ASR Transcribed Dataset

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

Table 1 shows the three sources of interviews, how many minutes of audio was processed, and the total word count. We carefully selected our recordings to evaluate how our complete system performs when summarizing a diverse range of speakers and topics such as finance, social issues, and medicine (A.1). 10 proportional transcripts[1] from each transcript source have been hierarchically summarized with human annotators in order to provide intermediate summaries. Transcripts that were not annotated were used subsequently in the annotation study (Section 5.2) [21] and the qualitative study (Section 6).

The procedures mimic the hierarchical summarization process of the System in Section 4. Starting with a longform transcript as the source input, the hierarchical recursive annotation procedure (A.2), is as follows:

(1) The source text is segmented using a fixed procedure, 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 process terminates.

All audio recordings consist 1, 2, or 3 speakers engaging in a dialog about different topics ranging from business to geopolitics to social sciences. All audio files are in English, although not all speakers are native English speakers.

4 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 3.2. 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 [31]. In a similar fashion with regards to leveraging external models (i.e. transformer models trained for different language tasks such as entailment [73]), we propose a hierarchical automatic summarization framework which emphasizes improving system robustness with a specific longform dialog domain focus. We utilize multiple different transformers including `BART-L` for dialog summarization, `PEGASUS` for short abstractive summarization and `T5` for grammar correction. These transformers were primarily chosen based on the pre-trained models that were available to the public on HuggingFace [75], which is important in ensuring that our research is reproducible and maintainable.

[1]We intend on releasing this dataset as a separate contribution in a separate work.

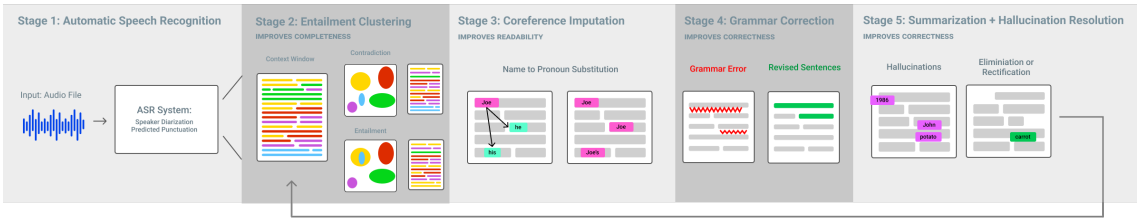

Figure 1: Dialog Improvement Framework. The overall contribution of the system are a series of training agnostic speech processing steps which improve overall summary text generation quality. Step 2 involves reordering input dialog to increase coherence. Step 3 and 4 imputes vague pronoun references 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.

4.1 Notation Walk-through

Input source texts (i.e. initially the ASR transcript) containing n sentences are denoted by $S = (s_1, ..., s_n)$. The segmented instance of S is defined as $S_i \in \mathbf{S}$, where each S_i contains 1 or more sentences and can be thought of as a portion of the original ASR transcript. Bold faced capital variables indicate a collection of a list of sentences. Each S_i is iteratively and sequentially ingested by the hierarchical summarization system. Concretely, input segment S_i is first processed and summarized, before S_{i+1} is processed.

The summary outputs from each level is given as $H_i \in \mathbf{H}$; each S_i and H_i correspond exactly, referred to as a segment-summary pair $(\mathbf{S}, \mathbf{H})$. At this point in processing, the outputs $\mathbf{H}^j$ become the new inputs S^{j+1} at the $j+1$-th hierarchical level. The superscript j indicates the current hierarchical level (e.g. S^j), the j-th recursive summarization iteration. References, or annotations, are gold label human authored summaries and similarly denoted $R_i^j \in \mathbf{R}^j$. For simplicity, the superscript j is dropped unless otherwise noted; the algorithm explanations assume an arbitrary recursive level j. Please refer to A.4 for more details.

4.2 Baseline Dialog Summarization System

The baseline hierarchical summarization system is adapted from [37] 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 [43, 60]. The initial semantic segmentation algorithm is already well suited towards chunking dialog as it fundamentally incorporates concepts that leverage coreferences and other speech cues [38, 44, 65] 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 fine-tuned on the SamSum Corpus [18] 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 1, Section 4.3-4.5). Finally the `Baseline` also adopts the hierarchical procedure from [37] as its hierarchical merging procedure. Additional details can be found in A.3.

4.3 Improving Adequacy: Entailment Clustering for Temporal Dialog 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 [13] and continuity [5] of text) or continuous. For example, a speaker discussing an idea could trail off and revisit that same idea a few sentences later. This leads to fragmented context and incoherently ordered thoughts [20] 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. See Figure 2.

4.3.1 Implementation. The operation starts with the previous hierarchical level's summary outputs $\mathbf{H}^{j-1} = H_{1...n}^{j-1}$ as $\mathbf{S}^j = S_{1...n}^j$ inputs where each individual summary S_i^j is iteratively processed in a streaming fashion. We consider three criteria to group summaries into semantic clusters. Pseudocode and full procedure is given in A.4.1.

(1) A well-formed semantic cluster should consist of summaries that do not contradict one another [49]. Specifically a high entailment beyond only a high semantic overlap (i.e. evaluated by the cosine similarity of `SBERT` embeddings [62]) 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 [74]. This is a key distinction from prior work as entailment allows dynamic assignment of cluster

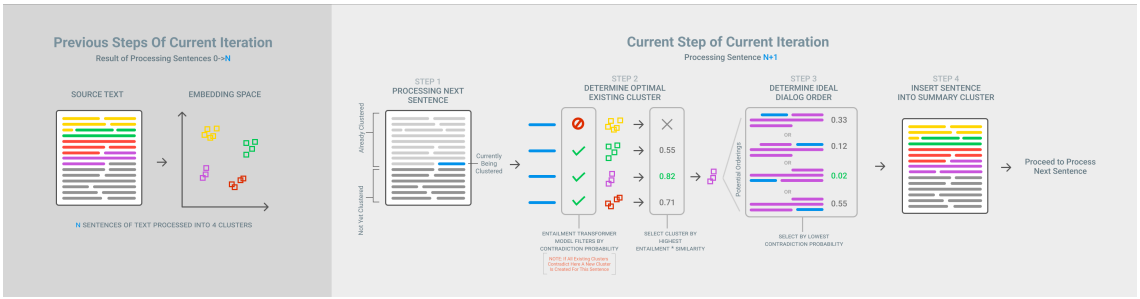

Figure 2: Entailment Clustering Process. To increase the summarization language model's semantic comprehension of the input text, the temporal ordering of dialog is rearranged and reordered to improve cohesion.

centers, in turn allowing for the procedure to reorient the temporal order of $\mathbf{S}^j = S^j_{1...n}$ 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.

4.4 Improving Readability: Coreference Imputation and Grammar Correction

Unlike formally written prose, conversation often contains ambiguous references. After an entity (such as a noun or other object) is introduced into a conversation, it is typically 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.

Moreover, the summarizer often introduces pronouns or references in place of named entities – likely as an attempt to make the text shorter. However, this adds additional vague or even hallucinated references to the summaries. Thus, the coreference imputation and grammar correction must be done at every hierarchical pass of the summarizer to ensure references are concrete in the all the summaries, including the shortest summaries.

4.4.1 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.

Table 2: 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 ...

Such imputations may induce syntactic grammatical errors (such as subject verb agreements) due to imperfect insertions. Accordingly we use a T5 [61] based neural grammar rewriter trained on a fluency corpus [55] to correct for small grammatical mistakes. A full procedure is given in A.4.2.

4.5 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 succinct categorization. An example of a positive hallucination would be replacing references to cars, buses, and boats with the label *vehicles*. The goal is to catch when summaries contain text that is semantically far from

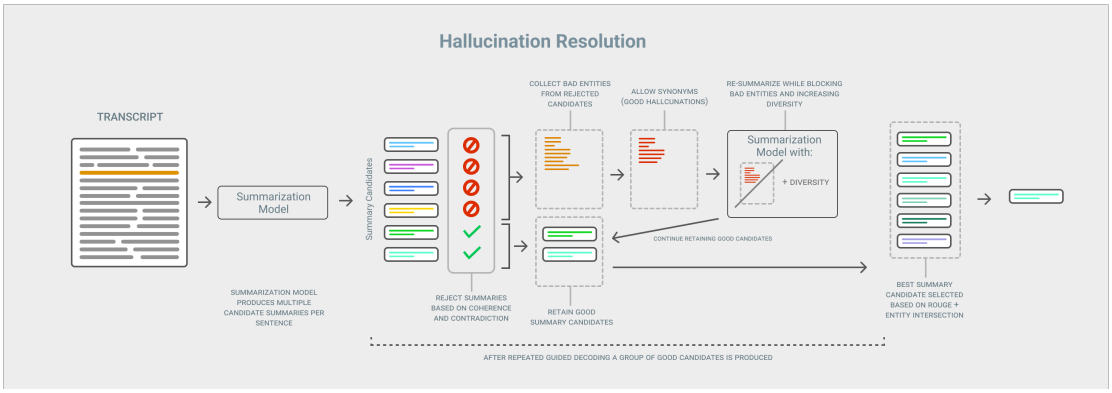

Figure 3: Hallucination Resolution Process. Through guided text decoding, the system re-ranks and selects the most plausibly accurate summaries for subsequent recursive summarization.

the context that was summarized. "Vehicles" is a positive hallucination because it is semantically close to the input ("cars, buses, and boats"). However, if the summarizer produced text based on its training data, that would likely produce negative hallucinations which would not be semantically close to the input.

4.5.1 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 [2] and Named Entity Recognition [34] 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 [52], 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. 1) with the original input S_i as the premise and H_i as the hypothesis and check for logical consistency [49].

We adapt BEAM search to decode multiple candidate summaries $H_i^{k \in K}$ where K = NUM_BEAMS [70] 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[2].
(2) *Increasing BEAM Search Space.* [70] 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 1. For readability, we use a language model LM_{wfd} trained on sentence well-formedness dataset[3] to score the syntax quality (0-1) of each summary. For accuracy, we use ROUGE-2 [40] to ensure that the summary and the input passage are similar in content. Eq 1 gives (0-1) an estimate encompassing both of these attributes.

$$\text{Quality}(H_i, S_i) = \alpha * \text{LM}_{wfd}(S_i) + \beta * \text{ROUGE}_2(H_i, S_i) \quad (1)$$

4.6 User Interface

4.6.1 User Interface Overview. Similar to previous hierarchical browsing systems [37], we present an interface where users can see all the *Short Summaries* on the left, then select any *Short Summary* to read more detailed information on the right (Figure 4). The detail information here consists of two options: 1) reading the corresponding long summary and 2) listening to the corresponding clip from the original audio. Previous systems also included medium-length summaries, but they were not seen as helpful to readers – if users wanted more details on a *Short Summary*, it was best to jump straight to the long summary which is often just a cleaned up version of the audio with speech disfluencies removed, but almost all details on the content preserved.

Although improving the readability, adequacy and accurate of summaries is the core features that will improve hierarchical summarization systems, we also introduce information cues to aid users in browsing the summaries. Because of the black-box nature of abstraction AI summarization, users might not have an intuition of how much information is being compressed in a *Short Summary*, the quality of the summaries, or how much of the total information they have seen so far. This information is critical to browsing an navigating behavior as users decide where to keep browsing in their current location or to move on.

[2]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.

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

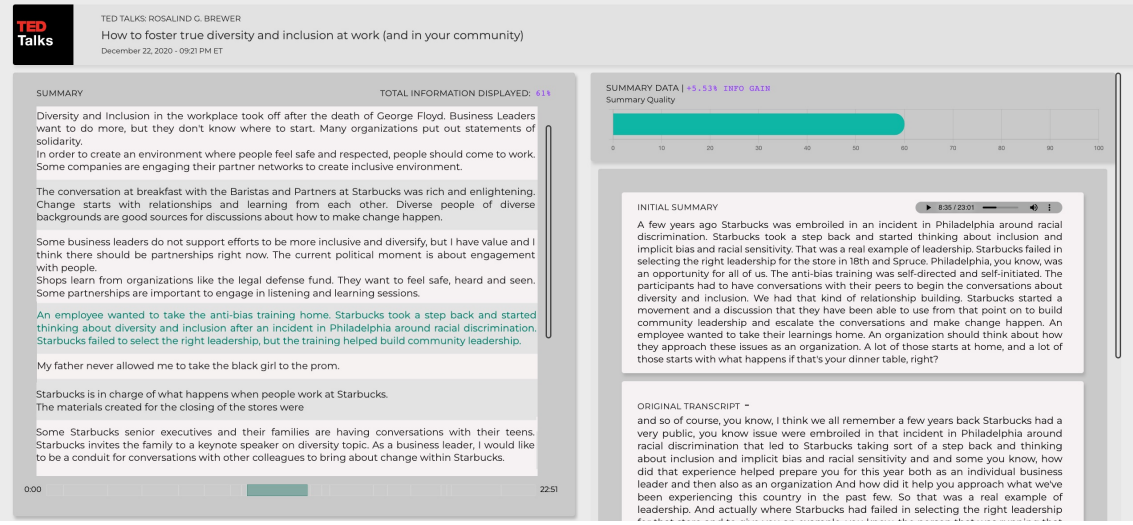

Figure 4: System User Interface. The left hand side gives a digest of short summaries, broken up by semantic summary segments. An estimate of how much total information is captured by these short summaries is given under "Total Information Displayed". By moving the cursor onto a summary, the user is 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.

To explore whether information cues would assist with navigation and improve user experience, we introduce three heuristics displayed near the top of the interface.

(1) *Total Information Displayed:* An estimation of the overall amount of content the user has read and encountered; continuously updating as the user reads.
(2) *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 current *Short Summary*.
(3) *Summary Quality*: An estimation of the quality of the summary the user is currently reading.

The total information displayed heuristic show the percentage of all the information (nouns and verbs) from the original transcript that the user has seen thus far. When a user first enters the system and sees only *Short Summaries*, the total information displayed is typically 50%-70% – meaning if the users reads all the *Short Summaries* they will have read a decent amount of information. As the user clicks on a *Short Summary* to read more, the total information displayed will increase to reflect the amount of new information seen. If he user clicks on every *Short Summary*, total information displayed will be 100%. This can help users keep track on how much they have read.

The information gain heuristic is based on the proportion of the nouns and verbs in a *Short Summary* which also appear in the part of the transcript the *Short Summary* covers. (See information retained defined in A.5) This is divided by the total information (nouns and verbs) in the audio file to express the global amount of information that can be gained by reading the transcript. It typically ranges between +0% and +6%. Thus, if a user sees that a *Short Summary* has "+5% information gain" it means that if the read the original transcript they will recover 5 percentage points of the total information (nouns and verbs) in the entire audio file. 5 percentage points is quite high, so it may be worth doing – quite a bit of information has been left out. However, if the *Short Summary* has less than +1% information gain, there probably isn't much more information worth reading behind the current *Short Summary*.

The summary quality heuristic is based on a weighted sum of readability (syntax quality) and accuracy (ROUGE-2) (see eq. 1). It typically ranges between 40% and 90%. Low summary quality can be a sign of ungrammatical trailing sentences such as *"The materials created for the closing of the stores were"*. From reading the initial summary, it is clear that key phrases were missing. *"The [diversity and inclusion] materials created for the closing of the stores were [used by many other companies.]"* High summary quality either indicates that 1) the *Short Summary* did not compress anything out (which is typical for ideas that are expressed in a single sentence, or 2) when the material that was compressed was highly redundant. For example, the *Short Summary "We don't know how to build lasting relationships and key partnerships."* has a relatively high summary quality (70%). Almost everything that was compressed was filler words on redundancy: *"What we don't know how to do is to build strong relationships that are lasting that are valued and I think that's where we need to start is relationship building and key partnerships."* This *Short Summary* also has low information gain (+0.19%) because almost no new entities are mentioned.

5 TECHNICAL 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

Table 3: 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 (longform) 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%

the model's performance and suitability for usage [45]. 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 = Concat($H_i \in \mathbf{H}$), also discussed in A.3.1), 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 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.

5.1 Automatic Evaluation

We evaluate the `Baseline` and our `System` on the 10 (out of 25) transcripts containing gold standard reference summaries using automatic metrics (Table 3). We use a standard automatic metric ROUGE [40] 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 Section 3.3, 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 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 [40], 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 (A.6). Summaries are evaluated at the *document* level.

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

(1) A steady and drastic decrease in ROUGE score in `Baseline` and `System` as a function of the hierarchical level, which is explained by the increased difficulty of content selection due to input length.
(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 8) 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.

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

[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 [15].

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 [29] as its evaluation does not contain coherence, flow, grammar, and factual correctness.

5.2 Annotation 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 3.2). 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 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.

5.2.1 Readability, Accuracy, Adequacy Evaluation. Annotators were instructed (in accordance with 3.2) 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 in order to obtain an inter-annotator agreement score (Krippendorff's Alpha) [32]. 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.

Table 4: Readability, Accuracy, Adequacy human evaluation of the `Baseline` and `System` generated *Short Summaries* (Level-3). The lower inter-annotator agreement (IAA) scores for Adequacy are italicized, emphasizing the inherent subjectivity in evaluating summary adequacy; what one annotator rates "adequate" has far more subjectivity than readability and accuracy. `System` is run with all steps.

Model & Level-j	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` IAA	0.26	0.28	*0.18*
`System-3` IAA	0.24	0.27	*0.16*

5.2.2 Aggregate Short Summary 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" [35]? Annotators were presented with all *Short Summaries* consecutively and were instructed to count each time they deemed there was confusion regarding logical consistency [13] or continuity [5] between 2 sentences. Additional details on computation and procedure are given in A.7. Summaries are evaluated at the *document* level.

Table 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. 4. `System` is run with all steps.

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

6 QUALITATIVE USER STUDY

Although the technical evaluation shows that progress was made in improving summary quality, the summaries are still imperfect. Thus, we perform a qualitative evaluation to understand how people use the system, what they find helpful and not helpful, and how they perceived and used the information cues in the system.

We investigate the following questions:

- How do users perceive the quality of summaries?
- Do the heuristics driving the information cues match what users expect?
- How do visual information cues impact users' browsing behavior?

6.1 Methodology

We recruited 10 participants[7] (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 (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

[7]These are different individuals from the human annotators in Section 5 and 3.3.

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 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.

6.2 Findings

6.2.1 Summary Quality Perceptions. All 10 participants felt as though the hierarchical summaries presented a complete digest of the podcasts. Ultimately, all users were able to construct a coherent and complete understanding of the transcript that they chose. Users all 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. *"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."* Likewise, P7 was impressed by the way the model was able to compress anecdotes and filter out speech filler.

As expected, the *Short Summaries* were imperfect, and using the hierarchy was necessary to understand some material. 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."* Similarly, P3 said, *"any summary I found inadequate, I listened to the audio for more information."* Both the initial summaries and the audio segments were found useful to regain comprehension.

Consistent with previous findings [37], there were times when user read a *Short Summary* deemed it to be "suspicious". This could be trailing sentences that were clearly missing information, or contradictory *Short Summaries* with potentially misleading information. when users encountered these, they often jumped ahead to see if additional *Short Summaries* were also suspect. The users were all aware that the summaries were automatically generated with AI, and this behavior demonstrated the consequence of mistrust in the AI. Thus, even with the improvements in summarization accuracy, adequacy and readability, **user trust and confidence in the system is still a concern for users**.

6.2.2 Perceptions of Information Cues. In general, participants felt that information cues matched their expectations. 9 out of 10 participants reported that the percentage of *total information displayed* approximately matched what they expected after listening to the entire podcast. 8 out of 10 participants agreement with the information gain associated with each *Short Summary* - 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 *"I think [amount of information gain] does match what I expected. For example, the section where [the speaker] shares an anecdote about her husband supporting their family while she works seemed difficult to approximate as she doesn't really say it explicitly, and rather just shares idioms about taking care of the children and house. In that section I expected a lot of information to be left out, and it looks like that was correctly identified."*

Moreover, having information cues like information gain were comforting to users and helped build their confidence in a system. P1 explained *"for a system to be aware of how much information it's leaving out...it makes it all that much more powerful."*

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. 3 used in estimating information gain.

Users who were less positive about information gain, found that it had an upfront learning curve. For example, P2 found information gain numbers confusing at first, so he *"click[ed] through one [Short Summary] with high error to understand what it meant."* 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.

Overall, despite some learnability challenges and occasional errors, information cues were deemed accurate enough to be useful to users and played an important role in building confidence in a system that uses AI that is sometimes flawed. We discuss ways of improving these metrics and their usability in future work.

6.2.3 Browsing Behavior with Information Cues. When browsing for information, users continually make the decision to either read more where they are or to move on to a new "patch" of information. Information cues in the system were designed to help users make these decisions. Indeed, users reported that both information gain

and summary quality help with the decision. Users did not mention total information display to be helpful or harmful in this area.

High information gain was a cue to readers to keep reading more in their current area. 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.' providing me with additional relevant content."* Although the *Short Summary* captured the main idea, the longer summary was worth reading to explain the historical context that was omitted by the summarizer, which this user found valuable. The high information gain estimate indicated this might be the case and cued him to keep reading.

Interestingly, other users relied on summary quality to decide when to read more deeply or carefully in their current area. P7 noted that if *"if the quality of a summary is supposed low, [I] would listen more carefully [to the corresponding original audio portion]"*. Thus, quality estimates for the AI generated summaries have the potential not just to build users' confidence in the system but also to **help users decide how to manage their attention while browsing.**

Low information gain was a signal for users to stop exploring their current section and move on. P9 noted that *the estimated information gain [was] a useful tool for helping me decide if [I] wanted to read the original transcript or not... that saves me time from reading redundant information.* Thus, information cues could help with **time saving from looking for more information when there isn't anymore to be found.**

7 QUANTITATIVE USER STUDY

To understand the time savings posed by the system, we conduct a controlled study measuring the time required to complete recall and information foraging tasks with and without the system.

7.1 Methodology

We recruited 12 additional participants[8] (8 male, 4 female, average age of 28) 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 45 minutes to 1.5 hours, averaging 1.2 hours, and utilized 4 source audio sources (from Table 7) and there associated hierarchical summary data and heuristics represented through the system interface described in 4.6. We designed a user study with two experiments each including a control task and an experimental task. Participants were randomly assigned 1 of the 4 audio sources or its corresponding summary data for each task.

Authors of the paper, prior to seeing summary data for the 4 specific audio sources selected for this study, listened to the material and extracted a list of salient points discussed in the audio files. These points were extracted from the complete duration of the audio. Next, we identified a set of points based on media of similar format discussing the identical topic to create a set of points which may have been in each selected audio source but were not.

The task order, experiment order, and assignment of audio to each task was counterbalanced. Participants were never given the same source audio or its associated summary data for multiple tasks.

7.2 Experiment 1 Procedure: Recall

The participant was presented a source audio media and asked to consume it using their regular listening/browsing habits in order to gain an understanding of the material. Once the participant had concluded consuming the media, they were presented with a list of points (half of which were present in the source audio and half of which were not). Participants were asked to select the points which were present, but were not informed how many points in total were correct or incorrect. In the experimental task, participants were presented with the system interface described in 4.6 including summary and heuristic data from a different source audio media than their control task and asked to consume the material. They were subsequently presented with a list of points with the same format, distribution, and procedure as the control task and asked to select the points they recalled from using the system interface. For both the control and experimental task, participants were not allowed to refer back to either the source audio or the system while answering the evaluation questions. The duration of time participants spent consuming material for each task was recorded.

7.3 Experiment 2 Procedure: Information Foraging

The participant was presented with both the source audio media and an associated list of points similar to those used in Experiment 1. The participant was then asked to utilize the source audio media to identify the correct points in the list. For the experimental task, the participant was presented with the system interface described in 4.6 including data from a different source audio and an associated list of points to evaluate and asked to perform the same task as the control. The duration of time participants spent identifying the correct points was recorded.

Throughout the experiment and each task, authors observed and noted the participants browsing behavior. Additionally, at the end of the experiment participants completed a survey regarding their experience consuming the material through different mediums and for different tasks.

7.4 Results

When using the system interface, participants on average performed marginally better on both the Recall and Information Foraging tasks, and notably spent nearly half as much time to achieve this. Participant "Scores" were calculated as the percentage of correct selection/non-selections of the list of points for each task.

In the control tasks, participants had a variety of consumption styles based on their typical browsing behavior. The majority (8/12) elected to listen to the media at normal speed while the remainder (4/12) chose to listen at accelerated speeds. With regards to usage

[8]These are different individuals from the human annotators in Section 5, 3.3, and the participants from Study 1.

Table 6: The results of the User Study for Experiment 1 and Experiment 2, averaged across all participants

Task	Control Score	Exp. Score	Score Diff.	Control Duration	Exp. Duration	Duration Diff.
1. Recall	0.847	0.896	+5.73%	22m42s	11m55s	-47.50%
2. Info. Foraging	0.905	0.972	+7.37%	13m34s	7m18s	-46.17%

behavior on the system interface, all participants elected to utilize the more detailed summaries, audio transcript, audio snippets, and heuristic views with differing frequency. Most users (11/12) elected for a depth first approach expanding many summaries as they read through linearly as opposed to reading through the high level summary first and going back to explore more deeply. Despite this common depth-first based traversal, all participants typically finished well ahead of the duration it took to listen to a comparable length audio.

Users recalled material and found information as accurately using the system interface and took only half the time. Participants performed on average 5.73% more accurately while using the System Interface for the recall experiment and 7.37% more accurately for the information foraging task. However, in both experiments no statistically significant difference was found between the control and treatment groups for accuracy scores (Recall: t=2.037, p=0.066; Info. Foraging: t=1.829, p=0.095) at $p < 0.5$. This may be due to the small sample size, variations in material duration, and variations in browsing and listening behavior. Despite the negligible difference in accuracy performance, note the duration participants spent to achieve comparable accuracy was statistically significant between control and experimental tasks (Recall: t=-4.855, p=0.00051; Info Foraging: t=1.829, p=0.00547) at $p < 0.05$. Users spent 47.5% less time consuming material using the interface for the recall experiment and 46.17% less time in the information foraging experiment highlighting the significant time savings utilization of the System Interface confers with minimal impact on information accuracy for both recall and foraging tasks.

Users reported using information cues to inform their browsing decisions. In both the recall and information foraging experiments, participants were frequently observed expanding certain summaries to access more detailed summaries, transcripts, or underlying audio for some sentences but not others. Participants were selective about which summaries they expanded. Multiple users expressed using the Summary Quality indicator as a gauge for whether to trust the high level summary sentence, P3 stated *"The summary quality [metric] was useful to help me decide whether to invest time expanding a summary to read further."* While users utilized the quality indicator as a decision point for diving deeper into specific summary levels, others utilized the information gain heuristic after expanding a given summary to decide whether to read further with P5 stating: *"I would click open a sentence and if the information gain was low, I would move on."*

Users found the system especially useful in the Information Foraging task. Participants found that generally the *Short Summaries* provided a clear and comprehensible outline from which to navigate. P12 stated *"I could glean 80% of the information from the left side view"*. Authors observed that participant's experience with the information cues aligned with their expectation of the heuristics behavior, although there was a clear learning curve. P12 noted *"After a few iterations, I began to trust the summaries and quality indicators more and could work faster."* P8 further noted *"When transcription was incorrect, the summaries were incorrect and I had to use the audio"*. This highlights another aspect of the interface which allowed for users to error correct utilizing audio snippets when needed. Multiple participants noted performing the same routine to correct for transcription errors.

8 DISCUSSION

8.1 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 6.2, 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 6.2.2. 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.

8.2 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 longform 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 [24]. 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.

8.3 System Limitations and Considerations

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. Currently, our system does not address retaining this form of information. 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. 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 methods for translating these forms of vocal information and conveying them to readers.

Additionally, users found there was a learning curve in order to build an intuition and mental model of the provided indicators. Though users found the system's visual information cues useful, they required additional work to understand. For example, users 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. Many participants also found it compelling to try to understand AI as a layer of abstraction between the audio file's speaker(s) and them as a listener. Users attempted to leverage this mental model to identify why the system was occasionally creating summary errors. Ultimately, although these indicators provided information scent and some transparency into the system's summarization process, the learning curve highlights the hurdles in the adoption of AI assisted tools to a wider audience.

Tackling longer form discussions also presents many challenges. While the algorithm presented would be able to recursively summarize to any depth, the `BART-L` model can only summarize a specific finite length of text. Future work can expand on trying different models such as `Longformer` which is designed to take thousands of tokens allowing us to increase the factor of summarization. Another aspect of arbitrarily longer discussions is that the user interface would have to showcase deeper layers of the summarization, but this would be possible by presenting a collapsible scroll-able list of intermediate summaries on the right side of the view of the existing system. On the other hand, with respect to handling significantly deeper summary hierarchies than this, future work may consider instead exploring novel interface designs distinct from the evaluated system which allow for more intuitive exploration, collapsing, and parallel browsing of very high numbers of layers at once to handle such scenarios.

8.4 Technical Limitations

Word recognition (transcription) of audio content has intrinsic accuracy challenges within automated speech recognition systems. 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. Although speech to text technology has grown more accurate over time, audio especially in noisy formats and uncommon domains contain many disfluencies and recognition errors which still remain a challenge to filter, translate, or accurately model. While our system attempts to address intrinsic speech issues, future work can be dedicated explicitly towards addressing word recognition errors. Additionally, imperfect automatically generated summaries may cause misinterpretations of audio content. Though abstractive summarization models have improved over time and addressed in this work, they still suffer from common language model challenges. Model generated summaries may hallucinate statements with no basis in the source audio leading to contradictory statements (accuracy), fail to retain important concepts in the original content (adequacy), or simply result in difficult to read outputs (readability).

8.5 Future Work & Implications

The system and interface evaluated in this work presents a novel method for automatically processing abstractive AI transcribed and generated summaries and presenting it in a navigable and browsable format. Much of the work has focused on tackling this summarization problem for audio, as this domain has additional challenges and complexities not present in structured written text and therefore served as a rigorous litmus test for the viability of such an approach. However, many of the processes and optimizations utilized in this system could be applied directly to long-form written text with similar effects therefore future work may consider evaluating performance of these methods directly on such mediums. Furthermore, as discussed in the limitations sections above, future work should seek to explore evaluating integration with a wider variety of long form language models as well as more versatile interfaces which can intuitive and easily accept deeper levels of hierarchical summarization.

9 CONCLUSION

This paper 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.

REFERENCES

[1] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding Long and Structured Inputs in Transformers. arXiv:2004.08483 [cs.LG]

[2] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. 1638–1649.

[3] Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. 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)*. Association for Computational Linguistics, Online, 132–142. https://doi.org/10.18653/v1/2020.wnut-1.18

[4] Barry Arons. 1997. SpeechSkimmer: a system for interactively skimming recorded speech. *ACM Transactions on Computer-Human Interaction (TOCHI)* 4, 1 (1997), 3–38.

[5] Frederic Charles Bartlett and Frederic C Bartlett. 1995. *Remembering: A study in experimental and social psychology*. Cambridge university press.

[6] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[7] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. 2007. Automatic speech recognition and speech variability: A review. *Speech communication* 49, 10-11 (2007), 763–786.

[8] Giuseppe Carenini and Jackie Chi Kit Cheung. 2008. Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the Fifth International Natural Language Generation Conference*. 33–41.

[9] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep Communicating Agents for Abstractive Summarization. https://doi.org/10.48550/ARXIV.1803.10357

[10] Jiaao Chen and Diyi Yang. 2021. Simple conversational data augmentation for semi-supervised abstractive dialogue summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6605–6616.

[11] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A Real-Life Scenario Dialogue Summarization Dataset. https://doi.org/10.48550/ARXIV.2105.06762

[12] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020. 100,000 Podcasts: A Spoken English Document Corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5903–5917. https://doi.org/10.18653/v1/2020.coling-main.519

[13] Baiyun Cui, Yingming Li, Yaqing Zhang, and Zhongfei Zhang. 2017. Text coherence analysis based on deep neural network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2027–2030.

[14] Zaynab Elkhattabi, Youness Tabii, and Abdelhamid Benkaddour. 2015. Video summarization: techniques and applications. *International Journal of Computer and Information Engineering* 9, 4 (2015), 928–933.

[15] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409.

[16] Wiqas Ghai and Navdeep Singh. 2012. Literature review on automatic speech recognition. *International Journal of Computer Applications* 41, 8 (2012).

[17] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. *ArXiv* abs/1911.12237 (2019).

[18] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237* (2019).

[19] Google. 2021. Speech-to-Text. https://cloud.google.com/speech-to-text/

[20] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. SNaC: Coherence Error Detection for Narrative Summarization. *arXiv preprint arXiv:2205.09641* (2022).

[21] Yvette Graham. 2015. Improving Evaluation of Machine Translation Quality Estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1804–1813. https://doi.org/10.3115/v1/P15-1174

[22] Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 4741–4744.

[23] Som Gupta and Sanjai Kumar Gupta. 2019. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications* 121 (2019), 49–65.

[24] Wang Haonan, Gao Yang, Bai Yu, Mirella Lapata, and Huang Heyan. 2020. Exploring Explainable Selection to Control Abstractive Summarization. *arXiv preprint arXiv:2004.11779* (2020).

[25] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. 1999. Auto-summarization of audio-video presentations. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. 489–498.

[26] Julia Hirschberg, Steve Whittaker, Don Hindle, Fernando Pereira, and Amit Singhal. 1999. Finding Information in Audio: A New Paradigm for Audio Browsing/Retrival. In *ESCA Tutorial and Research Workshop (ETRW) on Accessing Information in Spoken Audio*.

[27] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI Meeting Corpus. 364–367.

[28] Muhammad Khalifa, Miguel Ballesteros, and Kathleen McKeown. 2021. A bag of tricks for dialogue summarization. *arXiv preprint arXiv:2109.08232* (2021).

[29] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. 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*. Association for Computational Linguistics, Valencia, Spain, 199–209. https://aclanthology.org/E17-1019

[30] Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

[31] Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. 2021. Abstractive Text Summarization: Enhancing Sequence-to-Sequence Models Using Word Sense Disambiguation and Semantic Content Generalization. *Computational Linguistics* 47, 4 (2021), 813–859.

[32] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).

[33] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840* (2019).

[34] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* (2016).

[35] Mirella Lapata, Regina Barzilay, et al. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, Vol. 5. Citeseer, 1085–1090.

[36] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[37] Daniel Li, Thomas Chen, Albert Tung, and Lydia B. Chilton. 2021. Hierarchical Summarization for Longform Spoken Dialog. *The 34th Annual ACM Symposium on User Interface Software and Technology* (2021).

[38] Jiaqi Li, Ming Liu, Zihao Zheng, Heng Zhang, Bing Qin, Min-Yen Kan, and Ting Liu. 2021. DADgraph: A Discourse-aware Dialogue Graph Neural Network for Multiparty Dialogue Machine Reading Comprehension. In *2021 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN52387.

2021.9533364
[39] Suhyun Lim, Chanhee Park, Hyunwoo Han, Jaejong Ho, Junyup Hong, Soojung Lee, and Kyungwon Lee. 2019. A Narrative Topic Map Visualization to Summarize and Recall a Meeting. In *2019 IEEE Visualization Conference (VIS)*.
[40] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
[41] Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. 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)*. Association for Computational Linguistics, Online, 2467–2480. https://doi.org/10.18653/v1/2021.acl-long.192
[42] Yang Liu and Dilek Hakkani-Tür. 2011. Speech summarization. *Spoken language understanding: Systems for extracting semantic information from speech* (2011), 357–396.
[43] Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164* (2019).
[44] Zhengyuan Liu, Ke Shi, and Nancy F Chen. 2021. Coreference-aware dialogue summarization. *arXiv preprint arXiv:2106.08556* (2021).
[45] Elena Lloret, Laura Plaza, and Ahmet Aker. 2018. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation* 52, 1 (2018), 101–148.
[46] Sameer Maskey and Julia Hirschberg. 2005. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Ninth European Conference on Speech Communication and Technology*.
[47] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2012. Swift: reducing the effects of latency in online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 637–646.
[48] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Swifter: improved online video scrubbing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1159–1168.
[49] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. arXiv:2005.00661 [cs.CL]
[50] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. 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*.
[51] Kathleen McKeown, Julia Hirschberg, Michel Galley, and Sameer Maskey. 2005. From text to speech summarization. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, Vol. 5. IEEE, v–997.
[52] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
[53] Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. (2005).
[54] Kevin Kyung Nam and Mark S. Ackerman. 2007. Arkose: reusing informal information from online discussions. In *GROUP '07*.
[55] Courtney Napoles, Keisuke Sakaguchi, and Joel R. Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. *CoRR* abs/1702.04066 (2017). arXiv:1702.04066 http://arxiv.org/abs/1702.04066
[56] Douglas O'Shaughnessy. 2008. Automatic speech recognition: History, methods and challenges. *Pattern Recognition* 41, 10 (2008), 2965–2979.
[57] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. 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* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2807442.2807502
[58] Peter Pirolli. 2003. A theory of information scent. *Human-computer interaction* 1 (2003), 213–217.
[59] Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106, 4 (1999), 643.
[60] MengNan Qi, Hao Liu, YuZhuo Fu, and Ting Liu. 2021. Improving Abstractive Dialogue Summarization with Hierarchical Pretraining and Topic Segment. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 1121–1130.
[61] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).
[62] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *CoRR* abs/1908.10084 (2019). arXiv:1908.10084 http://arxiv.org/abs/1908.10084
[63] Yang Shi, Chris Bryan, Sridatt Bhamidipati, Ying Zhao, Yaoxue Zhang, and Kwan-Liu Ma. 2018. Meetingvis: Visual narratives to assist in recalling meeting context and content. *IEEE Transactions on Visualization and Computer Graphics* 24, 6 (2018), 1918–1929.
[64] Brij Mohan Lal Srivastava and Sunayana Sitaram. 2018. Homophone Identification and Merging for Code-switched Speech Recognition.. In *Interspeech*. 1943–1947.
[65] Nikolaos Stylianou and Ioannis Vlahavas. 2021. A neural entity coreference resolution review. *Expert Systems with Applications* 168 (2021), 114466.
[66] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014).
[67] Sunny Tian. 2020. Integrating Discussion and Summarization in Collaborative Writing. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3334480.3381437
[68] Nuno Vasconcelos and Andrew Lippman. 1998. 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, 153–157.
[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[70] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424* (2016).
[71] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. 2018. Speaker diarization with LSTM. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5239–5243.
[72] Karl E Weick, Kathleen M Sutcliffe, and David Obstfeld. 2005. Organizing and the process of sensemaking. *Organization science* 16, 4 (2005), 409–421.
[73] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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, 1112–1122. http://aclweb.org/anthology/N18-1101
[74] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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, 1112–1122. http://aclweb.org/anthology/N18-1101
[75] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771
[76] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862* (2021).
[77] Min Yang, Qiang Qu, Wenting Tu, Ying Shen, Zhou Zhao, and Xiaojun Chen. 2019. Exploring human-like reading strategy for abstractive text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7362–7369.
[78] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for Longer Sequences.. In *NeurIPS*.
[79] Klaus Zechner. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics* 28, 4 (2002), 447–485.
[80] Klaus Zechner and Alex Waibel. 2000. DIASUMM: Flexible summarization of spontaneous dialogues in unrestricted domains. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.
[81] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *CoRR* abs/1912.08777 (2019). arXiv:1912.08777 http://arxiv.org/abs/1912.08777
[82] Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H Awadallah, Dragomir Radev, and Rui Zhang. 2021. Summˆ N: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents. *arXiv preprint arXiv:2110.10150* (2021).

A APPENDIX

A.1 Dataset Audio Titles

Table 7: List of the Media in Dataset

Title
Bloomberg Wealth with David Rubenstein: Ron Baron
Bloomberg Wealth with David Rubenstein: Ray Dahlio
Bloomberg Wealth with David Rubenstein: Bill Ackman
Bloomberg Wealth with David Rubenstein: John Paulson
Bloomberg Wealth with David Rubenstein: Marc Andreessen
Maria Van Kerkhove: How to end the pandemic – and prepare for the next
The War in Ukraine Could Change Everything | Yuval Noah Harari
Extreme Ownership | Jocko Willink | TEDxUniversity-ofNevada
Elon Musk: The future we're building – and boring
How to foster true diversity and inclusion at work (and in your community)
How I Built Resilience: Kara Goldin of Hint
Lyft: John Zimmer (2017) How I Built This with Guy Raz
Leatherman Tool Group: Tim Leatherman
How I Built Resilience: Dr. Iman Abuzeid of Incredible Health
Burt's Bees: Roxanne Quimby (2019)
Springfree Trampoline: Keith Alexander & Steve Holmes (2019)
Coinbase: Brian Armstrong
Tate's Bake Shop: Kathleen King (2019)
How I Built Resilience: M. Night Shyamalan
How I Built Resilience: Elisa Villanueva Beard of Teach For America
Health-Ade Kombucha: Daina Trout
Chipotle: Steve Ells (2017)
How I Built Resilience: Lindsay Peoples Wagner of The Cut
Maverick Carter on Building the LeBron James Empire | The Limits

A.2 Hierarchical Dataset Annotation Details

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 $H_i^j \in \mathbf{H}^j$. The concatenated summary is given by $H^{j,c}$; the superscript c (for concatenated) is indicates the individual summary outputs have been concatenated. Outputs can be used as individual segment-summary pairs $(\mathbf{S}^j, \mathbf{H}^j)$ 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 $(S^j, H^{j,c})$. Since we are annotating, the $\mathbf{H}$ for "hypothesis" is dropped and now referred to as $\mathbf{R}$ for "reference".

A.2.1 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 $R_i \in \mathbf{R}$ 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[9]. $S_i^{'}$ is not used for future levels. This concludes the process for one level and is repeated identically until a termination condition is reached (final summary length).

A.2.2 Annotators. 10 annotators were chosen, all English proficient, having an undergraduate degree in the United States. Annotators were assigned 2 different transcripts, ensuring each transcript was annotated by 2 different individuals. Annotators noted this was an extremely time intensive process, resulting in each transcript requiring 5-6 hours for a total of 10-12 hours.

A.3 Hierarchical Summarization System Details

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 1), 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.

A.3.1 Compression Ratio. 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:

$$\text{Compression\_Ratio}(\mathbf{S}, \mathbf{H}) = \frac{|\text{Concat}(H_i \in \mathbf{H})|}{|\text{Concat}(S_i \in \mathbf{S})|} \quad (2)$$

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.

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

A.4 System Pseudocode Procedures

A.4.1 Entailment Clustering Details. Additional details on the entailment clustering procedure are given in Alg A.4.1.

Algorithm 1: 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 $\mathbf{S}^j = S^j_{1\ldots n}$. 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), 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

$L_{out} \leftarrow$ list();
$L_{clusters} \leftarrow$ list();
for $S_i \in$ S **do**
  if $|L_{clusters}| \leq 3$ **then**
    $L_{clusters}$.append(S_i);
    continue;
  if $|L_{clusters}| > p_r$ **then**
    Remove $L_{clusters}[0]$ and append to L_{out}
  $c_{candidates} \leftarrow$ list();
  for $S_k \in L_{clusters} : |S_k| \leq p_{len}$ **do**
    $sim \leftarrow M_{ST}(S_i, S_k)$;
    $ent \leftarrow M_{NLI}(S_i, S_k)$;
    $overlap \leftarrow M_{ENT}(S_i, S_k)$;
    $weighted\_score \leftarrow$ NORM($sim \cdot ent$);
    if $overlap > 1$ **then**
      $c_{candidates}$.append($[S_k, weighted\_score]$)
  $c_{chosen} \leftarrow$ MAX($c_{candidates}$);
  $order \leftarrow$ Higher Entailment Between $M_{NLI}(S_i, c_{chosen})$ and $M_{NLI}(c_{chosen}, S_i)$;
  $L_{clusters}$.append(concatenate($order$));
return L_{out}

A.4.2 Coreferenced Imputation and Grammar Correction. Additional details on the coreference imputation procedure. S_{impute} is the summary to have all coreferences imputed from a previous context $L_{context}$. The function AllenNLPCoref obtains all coreferences found in the concatenation between $L_{context}$ and S_{impute}; $c_j[1] \in S_{impute}$ is the ending coreference ending in S_{vague}.

Algorithm 2: Coreferenced Imputation and Grammar Correction Procedure.

Input: This procedure uses the following inputs and models.

(1) *Input*: List of summaries $\mathbf{S}^j = S^j_{1\ldots n}$. 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.

$L_{out} \leftarrow$ list();
for $i \in$ |S| **do**
  if $i < p_w$ **then**
    L_{out}.append(S_i);
    continue;
  $L_{context} \leftarrow [S_k \in \mathbf{S} : i - p_w \leq k < i]$;
  $S_{impute} \leftarrow \mathbf{S}_i$;
  $coreferences \leftarrow M_{CRF}(L_{context}, S_{impute})$;
  for $c_j \in coreferences : c_j[1] \in S_{impute}$ **do**
    Impute $c_j[0]$ word reference into $S_{impute}[c_j[1]]$;
    $S_{impute} \leftarrow M_{GRM}(S_{impute})$;
  L_{out}.append(S_{impute});
return L_{out}

A.5 Information Retained Rationale

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

Table 8: 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

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.*

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

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. 3 where proper nouns and named entities were given increased importance, but did not observe significantly material changes.

A.5.1 Sample User Flow. Here we provide a sample user workflow of the interface detailed above. Assume a piece of 30 minute longform audio content yielded 30 *Short Summary* segments out of the ASR, hierarchical summarization, and post processing pipeline outlined through Section 4.3-4.5.

(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. A.5), suggesting the intermediate summary is worth reading.
(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.

A.6 Ablation Studies

We run separate instances of our `System`'s pre and post processing steps: coreference imputation 4.4, cohesion clustering 4.3, and guided decoding 4.5. In comparison to the full method's (`System`) results in Table 3, individual aforementioned components usually improve, with the exception of Level-2's Guided Decoding. The offending score is italicized in red in Table 8, 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 3.2.

A.7 Aggregate Coherence Details

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. 4, is computed by the fraction of incoherent sentences out of all pairwise sentence pairs.

$$COHERENCE(\mathbf{H}^{j=3}) = 1 - \frac{\sum_{i=1}^{s \in S} 1(BREAK(s_{i-1}, s_i))}{|\mathbf{H}^{j=3}| - 1} \quad (4)$$

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.