Hierarchical Summarization for 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

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

Every day we are surrounded by spoken dialog. This medium delivers rich diverse streams of information auditorily; however, systematically understanding dialog can often be non-trivial. Despite the pervasiveness of spoken dialog, automated speech understanding and quality information extraction remains markedly poor, especially when compared to written prose. Furthermore, compared to understanding text, auditory communication poses many additional challenges such as speaker disfluencies, informal prose styles, and lack of structure. These concerns all demonstrate the need for a distinctly speech tailored interactive system to help users understand and navigate the spoken language domain. While individual automatic speech recognition (ASR) and text summarization methods already exist, they are imperfect technologies; neither consider user purpose and intent nor address spoken language induced complications. Consequently, we design a two stage ASR and text summarization pipeline and propose a set of semantic segmentation and merging algorithms to resolve these speech modeling challenges. Our system enables users to easily browse and navigate content as well as recover from errors in these underlying technologies. Finally, we present an evaluation of the system which highlights user preference for hierarchical summarization as a tool to quickly skim audio and identify content of interest to the user.

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, Albert Tung, and Lydia B. Chilton. 2021. Hierarchical Summarization for Longform Spoken Dialog. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3472749.3474771

*Both authors contributed equally to this research.

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474771

1 INTRODUCTION

Spoken dialog is a rich source of information; many media platforms frequently host discussions on important topics ranging from healthcare and diversity to economics and politics. Unfortunately compared to text, spoken dialog can be challenging to consume as it is slower than reading and difficult to skim or navigate. Although people may be interested in a given topic, they may be unwilling to commit the required time necessary to consume long form auditory media given uncertainty as to whether such content will live up to their expectations. There exists a clear need to provide access to the information spoken dialog provides in a manner through which individuals can quickly and intuitively access areas of interest without investing large amounts of time.

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

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

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

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

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

This paper makes the following contributions:

(1) An end-to-end system that automatically generates hierarchical summaries of longform spoken dialog.
(2) A novel semantic segmentation algorithm that allows the reuse of existing machine summarization models rather than training a new one.
(3) A user study demonstrating:
   (a) the system is 72% accurate in producing condensed *Short Summaries*.
   (b) system hierarchical features enable users to recover their understanding of 98% of summaries despite ASR and ML summarization model errors.
   (c) the average time that users spent to reach an understanding of an audio recording was 27% of the original audio length.
(4) Qualitative findings about how people use *Short Summaries* as navigational tools to help them "skim" audio and find the content most interesting to them.

2 RELATED WORK

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

2.1 Using NLP to Generate Multimodal Interactions

Researchers have developed models and systems to easily navigate through videos and movies by navigating to the video clip and allowing users to interpret content [Barnes et al. 2010; Goldman et al. 2006; Jackson et al. 2013]. However, these videos require users to search visual information in a video they may know little about and is inapplicable to pure audio files. To solve these issues, some researchers have employed summarizing key content in text as a means of helping users easily digest long-form content [Pavel et al. 2014], [Pavel et al. 2015]. More recent work has adapted the use of hierarchical information to provide users with multiple levels of summarization and information [Truong et al. 2021]. We build atop these systems targeting multi-party audio transcripts which pose novel challenges because these transcripts necessitate proper semantic segmentation to preserve meaning across speakers while simultaneously leveraging the usefulness of hierarchical information.

Still other work utilizes NLP to generate multimodal interactions such as images for video editing or even adding visuals to existing audio files [Xia 2020; Xia et al. 2020]. However, they rely on human-created transcripts, hurting the ability for the system to scale without automatic processes. Furthermore, visual representations only represent higher level abstract topics not the summarizations needed to represent the speaker.

2.2 Summarization of Multi-Party Audio

Creating meaningful summarizations from multi-party audio has been a difficult problem for researchers, often requiring hierarchical transformers and speaker segmentations to effectively retain information. Many of these papers, however, require full end-to-end training on transformers and even custom datasets [Karlbom and Clifton 2020; Li et al. 2019; Vartakavi and Garg 2020; Zheng et al. 2020; Zhu et al. 2020]. Still others also employ graph-based summarization and coreferences to better summarize discourse [Xu et al. 2019]. Meanwhile, current unsupervised abstractive summarizations do not utilize deep learning summarization modules and require the use of word graphs and ranking algorithms [Shang et al. 2018]. These works focus on learning end-to-end summarization which is not practical across multiple domains. Instead, we focus on utilizing these summarization systems as part of a larger unsupervised abstractive system to generalize and reduce the overhead needed to deploy and scale such a solution.

2.3 Automatic Speech Recognition and Abstractive Summarization

Automatic Speech Recognition systems (ASR) are used to transcribe audio (word recognition) into a source language transcript and have recently made relatively significant strides in terms of practical performance. Additionally, state of the art ASR [Google 2021] is no longer constrained by vocabulary and remains relatively robust, encouragingly extending word recognition to topical domains and noisy audio.

Text summarization techniques can be classified into two categories: abstractive and extractive. Abstractive summarization generates a new unique summary of text given a context whereas the extractive summarization "quotes" and concatenates relevant portions to compose into a summary. Because of spoken language noise effects in ASR transcripts, extracting transcript segments verbatim often leads to poor summaries. Therefore, we opt for the current state of the art abstractive summarization model, PEGASUS [Zhang et al. 2019], which is able to achieve much higher human-quality summaries. This is achieved by innovatively changing the pre-training process from standard word level masked language modeling, where models learn language conventions and syntax by predicting individually removed words within sentences, to sentence level masked language modeling, where entire sentences are removed and then recovered. This training process gives PEGASUS a high level of document understanding and helps to distill important information. Though promising, like most language models, it is important to note that PEGASUS is tailored towards specific benchmark datasets such as news or social media and that performance does not translate across different data domains, especially when applied to speech specific noise and disfluencies.

2.4 Recursive and Hierarchical Summarization

Summarization of long complex material into recursively shorter and more tractable artifacts has been previously explored and found to provide an effective avenue for gaining useful comprehension of content [Zhang et al. 2017]. Notably, this work showcased an interface displaying multiple summaries with varying levels of detail resulting in users having superior substantive recall and enabling non-linear exploration of the source material. However, this prior work employed crowd-sourced techniques to generate summaries and targeted solely threaded discussions typically found in forums. We build off these findings by developing a novel system employing automatic summarization and speech recognition techniques to spoken dialogue in order to generate a similar hierarchical exploration of content without requiring human-in-the-loop summary generation.

The utility of hierarchical summarization has also been shown for multimodel instructional videos that use audio and video to demonstrate each instructional step [Truong et al. 2021]. By using computer vision, ASR, and domain-specific heuristics they automatically group fine-level actions into coarse-level events (with summary text) that users can navigate at their own pace. We build on these ideas by using machine summarization to provide multiple levels of summarization detail and allow users not only better navigation but also time savings in consuming media.

3 FORMATIVE STUDY

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

(1) ASR and summarization models are far from perfect and have inherent pre-existing challenges.
(2) Summarization models are almost always trained on text rather than speech data. If a text trained summarization model is deployed on speech data, there would be a data domain mismatch, leading to considerably degraded model performance.

Compared to text, speech is far less structured - there are no topic sentences to rely on, speakers can stop mid sentence and backtrack their thought or never complete it, and coherency is challenging when multiple speakers are making different points simultaneously. Additionally, speech contains informal language and disfluencies such as hesitation and vocal fillers. These reasons heavily indicate that existing text-trained summarization models will perform very poorly on speech dialog.

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

(1) *Coherency*, are the final output summaries coherent? If this constraint is not met, the model is not usable. Aside from re-training and adapting a model towards speech data, we have no tractable strategies for compelling model coherence.
(2) *Information retention*, because output summaries are shorter and lossy, we check if they still retain salient information from the original passage. If a shortened summary does not contain useful or relevant information, it has no value.

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

3.1 Evaluation Data

We evaluate seven recordings of longform spoken dialog that span different topics, domains, and speech styles (Table 1). The average length of the recordings is 32.5 minutes and the average word count output from ASR is 5622. Of these seven recordings, four are edited interviews from the NPR podcast "How I Built This", and 3 are unedited recordings from live events. Two are Bloomberg interviews regarding finance and one is a conversation about "How to foster true diversity and inclusion at work (and in your community)." These recordings were selected based on being content rich and of reasonable length. Information rich dialogue serves as a useful medium for this experiment by providing a sufficient density of information to showcase summarization. Additionally, choosing sources from the same producer reduces variance and provides a consistent structure for experiments. Finally, our experiment included both edited and unedited recordings of dialog to expose our system to both more coherent and structured conversations as well as free form dialogue.

3.2 Automatic Speech Recognition Model

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

3.3 Summarization Models

For summarization, we investigate the current abstractive state of the art language model PEGASUS. While PEGASUS is noticeably improved over other summarization methods in terms of producing human level quality summaries, it requires fine-tuning onto domain specific summarization data. It is also important to note that a pre-trained only instance of PEGASUS is not normally used without modification; the pre-training procedure is different from summarizing and the authors focus solely on fine-tuned downstream summarization datasets. Appropriately, we select fine-tuned instances from `huggingface.co` [Wolf et al. 2019] that generate complete and grammatically correct passages (i.e. not a few keywords) and are still in considerably general domains (i.e. not a

Table 1: Dataset metadata used in formative study and final evaluation

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

Table 2: Model nomenclature where `M`*i* indicates Model *i*, training data descriptions, and model maximum input and typical output sizes.

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

medical field model instance) to assess PEGASUS coherence and information retention. Model details are given in Table 2.

We begin by processing audio files to obtain raw ASR transcripts. However, because of the nature of longform dialog, the number of words per transcript greatly exceeds the maximum input length that **M1, M2, M3** can accept. Transcripts must be processed and split into manageable lengths. We naively segment the transcript in fixed 60 word length segments set to **M3**'s maximum input length[1]. For example, if an input transcript segment had a total of 154 words, it would be broken into a list of 3 individual segments, each containing $[60, 60, 34]$ words. To maintain evaluation consistency across all models, any evaluation involving naive fixed segmentation is set to 60 words. These are then summarized by **M1**, **M2**, and **M3**, which are set to output summaries containing at most half of the original passage's words.

3.4 Heuristic Score

We evaluate a summarization model's coherency and information retention using a heuristic score consisting of state of the art automated metrics in natural language processing. For coherence evaluation, we use a `BERTScore` [Zhang et al. 2020] between a reference ASR segment and a model generated summary (candidate input). This method correlates well with human evaluation and uses word level contextualized embeddings to capture dependencies and word ordering. For retained information, we use the cosine similarity between `Sentence Transformer` [Reimers and Gurevych 2019] embeddings of a reference ASR segment and a model generated output summary. A higher cosine similarity between the reference ASR segment and output summary suggests the summary captures the reference ASR segment's semantic content. The final heuristic is the simple average of the two and has a range of $[-1, 1]$. In practice, cosine distance based metrics used to determine similarities between word embeddings are positive, with a general range of $0 - 0.5$ for a weak correlation, $0.5 - 0.8$ for a moderate correlation, $0.8 - 1$ for a strong correlation, and 1 for a perfect correlation [Jatnika et al. 2019]. As a sanity check, we observe a correlation of 1.0 when we set the reference and candidate text inputs to be the same. Intuitively, as **M1**, **M2**, and **M3** outputs are still summaries, they will contain at least some semantic similarity to the reference ASR segment; therefore we expect to observe a somewhat moderate correlation $(0.5 - 0.6)$ with our heuristic. After determining which model can be feasibly re-purposed, we use the heuristic score again to evaluate our method's impact towards improving summarization (Section 4.2.1). A more exhaustive discussion on heuristic score motivations, details, and limitations is given in the appendix, section B.

3.5 Formative Study Findings

We discuss Table 3 throughout this section. It contains an example of the ASR transcript segment of one speaker in the "Diversity and Inclusion" recording and the corresponding summaries generated by the three models. Text is color coded to indicate shared regions between the ASR transcript and the summary.

3.5.1 Google Speech-to-Text Automatic Speech Recognition Quality. We quickly and quantitatively evaluate the word error rate (WER) of the ASR system. Because Table 1 only consists of audio data and perfect transcripts (human transcribed) are non-existent, we benchmark ASR performance on a random subset of Ted Talks as they are somewhat similar in terms of speech and data structure to Table 1 and thus would likely be indicative of ASR performance. We find an average WER of 10%[2], slightly above the reported 6.7% WER [Kim et al. 2019], and far below a usability constraint of 30% [Gaur et al. 2016].

As seen in the provided ASR Transcript example in Table 3, the ASR Speech-to-Text makes very few errors. However, rare words, unfamiliar phrases, or new words not yet encountered still degrade performance. For example, in the NPR: Chipotle dialog, *"mise-en-place"* was mistakenly transcribed as *"knees in place"*. In the "Diversity and Inclusion" dialog, *"rectangle. Opening"* was mistakenly transcribed from *"reckoning"*. Additionally, performance can fluctuate due to a variety of noising factors such as speech disfluencies,

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

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

Table 3: Example summary generations on the "Diversity and Inclusion" dialog showing an instance of the ASR transcript for one speaker turn and errors frequently made by the three summarization models. Text is color coded to indicate shared regions between the input and output.

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

foreign accents, and audio recording quality. Although ASR makes few errors, they will propagate to downstream tasks and create challenges for generating a practical audio summarization system.

Table 4: Automatic evaluation heuristic scores for various segmentation strategies.

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

3.5.2 Summarization Model Quantitative Analysis. Table 4 gives the automatic evaluation heuristic scores for **M1**, **M2**, and **M3** ranging from 0.61 – 0.70. Despite generating summaries on out of domain speech data, we can conclude that all the baseline language models can still reasonably function and retain a moderate amount of information with a summary containing at most half the words as the input ASR segment. Nonetheless, the the tight spread of the heuristic score range indicates a moderate correlation and merits further investigation into the re-usability of **M1**, **M2**, and **M3** to fully understand model behaviors. While the heuristic score is telling, it is not a replacement for human level evaluation; it provides only a limited perspective into performance that is subject to intrinsic methodology constraints enumerated in Appendix B.2. To get a sense of what types of errors the automatic summarization models are making and whether they could potentially be addressed, we studied various segments by hand.

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

M1 produced summaries that contain frequent hallucinations [Maynez et al. 2020] – phrases or entities that appear to be semi-relevant but are not actually present in the underlying text. This can be attributed to its news based training data.

For example, in Table 3 **M1's** summary contains the text "African-American journalists" and "Richard Lawson." Neither of these entities are mentioned in the input (or entire audio file). However, these entities are in **M1's** training data. This is a typical problem seen in language models when deployed on new data that is not encountered in training. Only recently, an attempt at fixing hallucinations has resulted in improved ROUGE precision and increased human preference [Zhao et al. 2020], but still requires additional dataset generation. These errors are in almost every summary produced by **M1**. Thus, fixing **M1**'s hallucinations would be nontrivial and require a new training dataset.

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

M3 has more fluent text with no hallucinations. However, it makes an egregious error of misrepresenting the content. The transcript clearly states that ' *"The [Black and Latin] talent is out there,"* but the summary introduces a negation to say that the talent is not there. The root of this problem is that **M3** coerces two different segments into a semantically incorrect summary. These errors occur when multiple non-sequitur or different topics are provided as a single input. Because abstractive summarization generates words that are not necessarily present in the source input text, they require a high degree of content understanding of the underlying semantic information in the passage [Gliwa et al. 2019] to successfully

generate a semantically faithful summary; a poorly segmented input containing multiple different concepts would be exceedingly detrimental towards a model's semantic comprehension. Thus, **M3**'s resulting coherent and abstract summaries (albeit with contextual misrepresentation errors) signal that:

(1) A successful semantically accurate segmentation that can group similar topics and ideas together, while splitting dissimilar sentences into a separate chunk can improve a model's semantic comprehension, and transitively improve summary generation accuracy.
(2) The summary context's input accuracy issue is now reframed as a processing challenge that does not require changes to the model's architecture, re-training, or additional annotated training data.
(3) **M3** is able to maintain its abstractive nature, which is essential to summarizing dialog due to speech disfluencies and other noise artifacts.

3.5.4 Formative Study Key Takeaways. Based on this exploration of the three models, we hypothesize that **M3** is the best one to build on top of and reduces the challenge of practical dialog summarization to a tractable problem. **M1** and **M2** errors are exceedingly difficult to correct without significant amounts of specialized speech training data. **M2**'s marginally higher score over **M3** is immaterial given **M2**'s disfluency and incoherence[3]. Although incoherent topic grouping is rarely the case in written language where ideas are well-formed and presented in a manner that is optimized for ease of understanding, it is usually the norm in spoken language where topics shift over time as speakers react to the last thing that was said. Concretely, if we can segment transcripts into semantically cohesive segments, creating easier inputs and facilitating improved summarization performance, **M3** may be an effective summarization tool. When errors do remain, there is the fallback of the user using the hierarchical browsing features to investigate surprising or suspicious claims to see if the summary is consistent with the text.

4 SYSTEM

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

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

[3]Refer to Appendix B.2 for an explanation to why **M3** still achieves a comparable score to the other models.

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

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

The inclusion of these procedures to the two-stage framework enables not only grammatical and semantic cohesiveness but also facilitates various levels of summarization detail:

(1) *Long Summary*: Cleaned ASR Transcript. At this stage, a transcript's disfluencies and noises are cleaned and presented according to conversational order or speaker turns.
(2) *Medium Summary*: Moderately Detailed Summarization. Similar *Long Summaries* are merged and further paraphrased, providing key concepts along with essential details.
(3) *Short Summary*: High level Summarization. Similar *Medium Summaries* are further merged to provide the transcript's salient ideas in more concise language.

4.1 Interface

The interface (Figure 1) consists of three main sections: *high level summary* column on the left, the *segment data view* in the middle and the *timeline of segments* at the top of the interface. Users explore the content by first browsing the *Short Summary* column to get a high level overview of the content.

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

The interface was designed with two goals in mind:

4.1.1 Design Goal 1: Enable users to quickly identify useful information to them. Presenting high level summaries to the reader allows them to quickly grasp a general idea of what is being said. However, simply reading *Short Summaries* may not entirely satisfy the reader. By nature of being summaries, they may omit details that may be of interest. Additionally, the automatic summarization algorithms are imperfect and sometimes present summaries that are more vague than a user would prefer. However, the purpose of the *Short Summaries* are not necessarily to fully summarize, but to allow the user enough information scent [Pirolli 2007] to decide if they want more detail. If they want more detail, they can use the hierarchy of summaries to read *Medium* or *Long Summaries*, read the ASR transcript, or listen to the underlying audio.

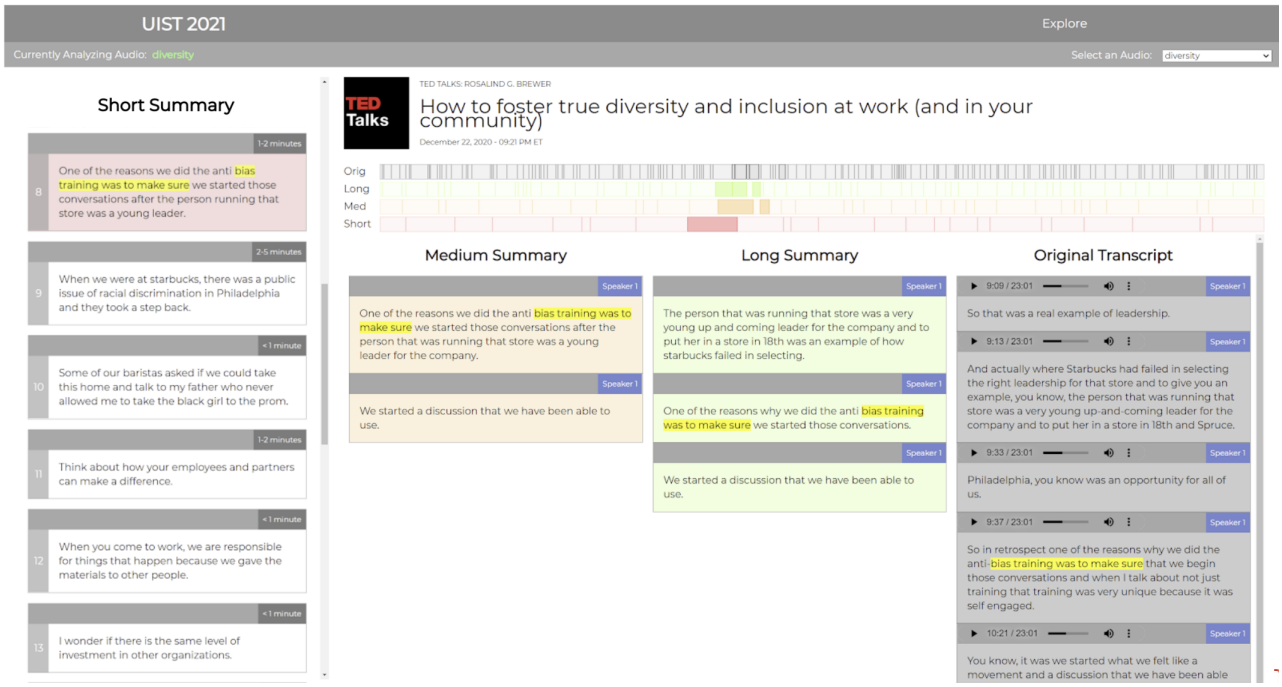

Figure 1: System User Interface. Example of the system's summarization display for each unique audio file. The left part of the interface contains several short summaries which, when clicked, displays the medium and long summaries along with the corresponding original transcripts and audio clips. The top part of the interface shows what part of the transcript each summary encapsulates information about. When the top part is clicked, users can navigate to any part of the summaries or transcripts.

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

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

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

4.2 Summarization Algorithm and Implementation

Figure 2 shows the steps through which an audio file is recurrently processed to obtain different levels of summarization (*Short, Medium, Long*). At a high level, the system segments an ASR transcript and iteratively summarizes previously combined conceptually similar segments to obtain increasingly abstract summaries while preserving semantic meaning.

Stage 1 (Fig. 2) of the system employs ASR to create a speaker diarized transcript of the input audio file. These speaker turns are further processed by a semantic segmentation algorithm which divides a given speaker turn into chunks of semantically related sentences. The now refined speaker turns are iteratively given as inputs into stage 2 (Fig. 2) of the system where processing and hierarchical automatic summarization occurs. After each speaker turn is individually summarized, its summaries are embedded which are then used to cluster sentences of the summary. Clusters are concatenated (merged) and shorter summaries, which generally contain little salient information, are stemmed. The first resulting summary of this iteration through the pipeline represents a *Long Summary*. This *Long Summary* is fed back into the the automatic summarization model and follows the same embedding, hierarchical clustering, and stemming steps once more to generate a *Medium Summary*. One further cycle using the *Medium Summary* yields a *Short Summary*.

Details of each system component are as follows:

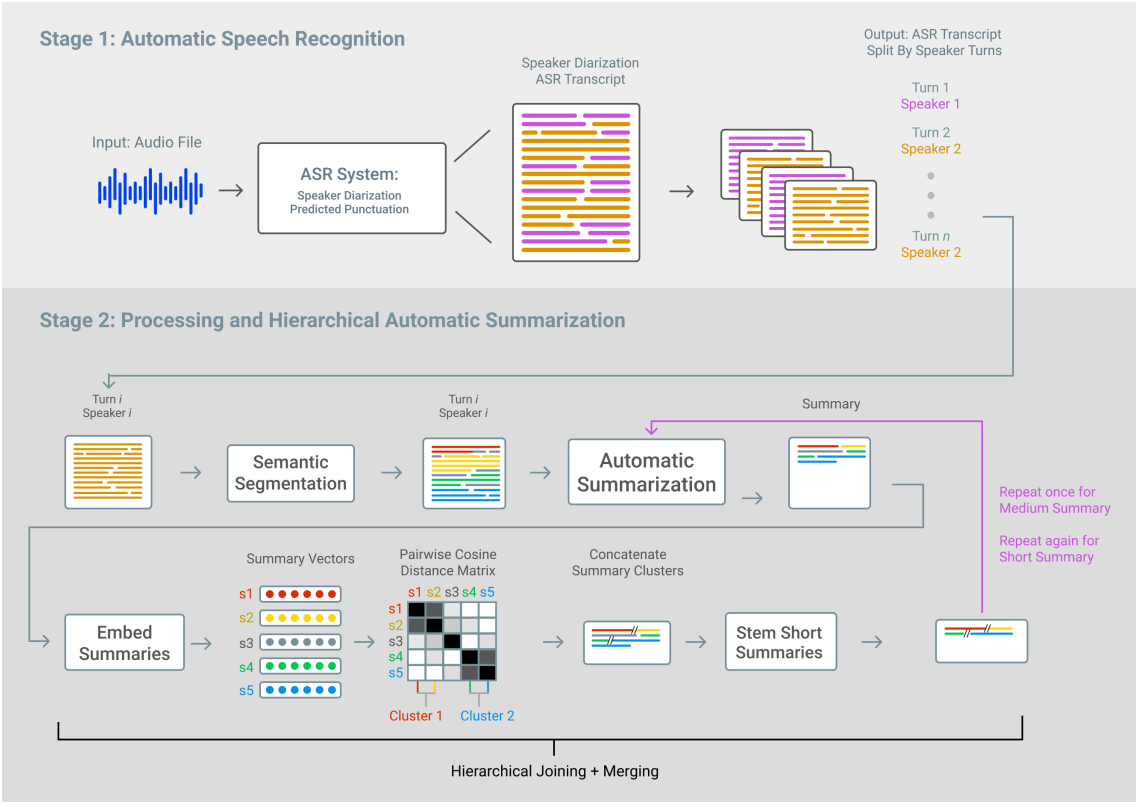

Figure 2: Summarization Generation Pipeline. Our system enables the conversion of audio files to multiple tiers of summarization. In the first stage, we convert the audio file into a speaker-segmented and punctuated transcript and process the transcript, split by speaker turns. In the second stage, we take each speaker turn and cluster conceptually similar summaries via semantic segmentation. Each cluster's summaries are joined (concatenated) based off of semantic similarity. We remove small summarizations and then repeat the summarization and merging process to obtain the *Medium* and *Short* summaries.

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

Coreferenced Semantic Segmentation. To understand why we employ coreference resolution [Soon et al. 2001] and speaker shifts to semantically link sentences together and correct poor segmentation, we must recognize two linguistic tendencies:

First, unlike written prose, conversation can be far more vague; nouns and objects, herein referred to as entities, are only initially mentioned and then sporadically referenced, while all other mentions are pronouns (it, s/he, they, etc.). Generic topic modeling of dialogue performs poorly due to the nature of conversations, since conversations have both local and global topic structures that have weak signals in conversation [Takanobu et al. 2018]. Individuals can talk about a topic using specific references, but a model not trained to recognize these topics could fail to recognize the boundaries of the topic effectively. To eliminate a dependency on custom training data, we instead choose to identify expressions that refer to the same entity using coreference resolution. To employ this technique, our algorithm seeks to group sentences spanned by an entity, which is defined as the sentences contained by the start and end of the entity. This method of using coreferenced entities to model text has historically been shown to be successful [Stoyanov and Cardie 2006] [Witte and Bergler 2003] and is still used in current state of the art models [Xu et al. 2019]. We use the publicly available Allen NLP API [Gardner et al. 2018] for state of the art coreference resolution. We directly refer to variables in the pseudocode for Algorithm 1, given in appendix section A.1, in the following walk through of the process.

Second, speaker shifts may begin relatively different concepts [Maynard 1980] and repeated references to the same entity indicate the same concept is still being discussed. Sudden changes in speakers can be correlated with topic boundaries [Galley et al. 2003] and this concept serves as a common segmentation approach in NLP [Zhu et al. 2020]. All iterative instances of the algorithm on each speaker turn segment S_i are therefore independent of each other.

Here we walk through a single speaker turn pass of Algorithm 1. Speaker turn S_i contains sentences s that were previously divided by ASR predicted punctuation. For each sentence s, we first obtain the coreferenced entity c_{min} (`line 13`) that maximally spans the

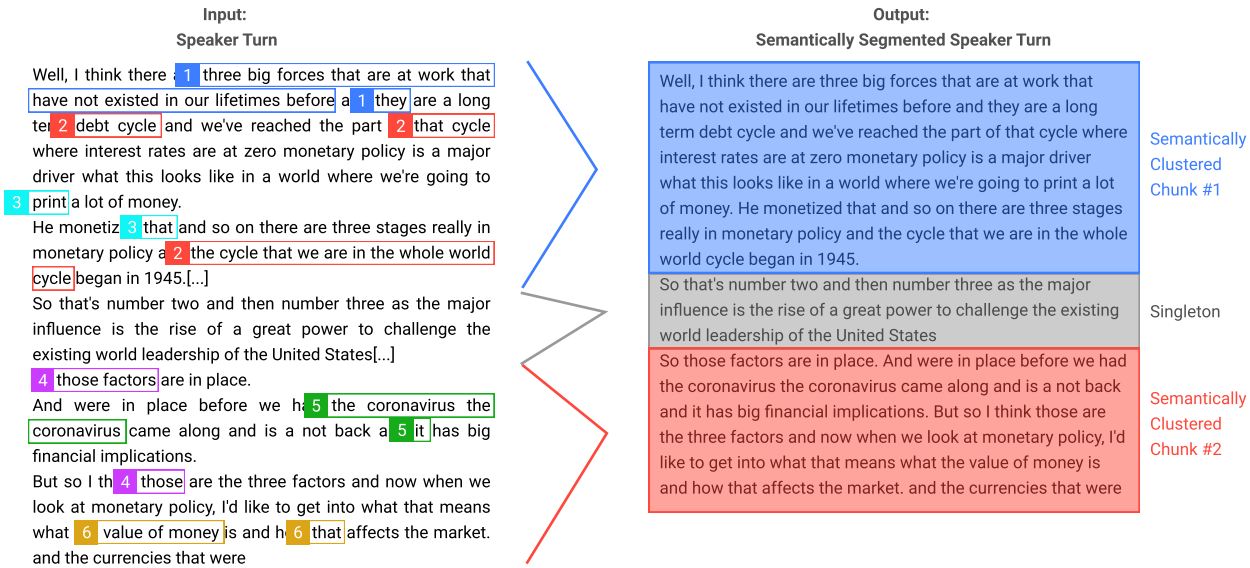

Figure 3: Alg 1. Semantic Segmentation Example of one speaker turn input into our coreference resolution algorithm. On the left, the coreference tags generated from the AllenNLP coreference resolution module are shown with six different references highlighted. Our algorithm groups sentences with references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

current sentence s and future consecutive sentences $s \in S_i$ (`lines 4:10`), subject to constraints. This denotes the start and end pointers (e_0, e_f) of the current semantically similar chunk, **P** (herein referred to as cluster, `line 13`). Sentences contained by **P**'s span e_0, e_f, are assigned to **P** (`lines 19:20`). If sentence s contains another entity that spans further than **P**'s current end pointer, e_f is updated to the sequentially higher indexed sentence (`lines 21:23`). When **P**'s span is exhausted and cannot be further extended, the algorithm begins a new cluster **P**. Sentences that contain no entities are singleton clusters.

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

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

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

The tradeoff for increased robustness towards speech noise and artifacts is also inherently found in **`M3`**'s paraphrasing nature; **`M3`** struggles to reason out semantically different ideas and suffers substantially from contextual errors (Table 3, **`M3`**). However, when ASR transcripts are preprocessed with Algorithm 1, our framework is able to generate not only cohesive and semantically logical summaries, but also achieve practical accuracy.

Hierarchical Concept Clustering and Merging. The next challenge is to determine which of the previous level's summaries to concatenate for further abstract summarization (appendix section A.2). Recall that semantically similar summaries must be joined or the model output can be factually incorrect (Table 3, **`M3`**) as abstractive summarization requires a high degree of semantic understanding of the underlying input passage. As a means to compare summary content similarity, we first use `Sentence Transformer` [Reimers and Gurevych 2019] to individually embed summaries (still contained in their own speaker turns S) $s \in S_{i=0}^{n}$, into vectors in a semantic space. By transforming the text segments into vector representations, we can now quantitatively compare their similarities via cosine distance. Summaries within each speaker turn S_i are then merged through usage of a pairwise cosine distance matrix for hierarchical (agglomerative) clustering. Merges are done within speaker turns to enforce a proximity constraint of only merging local summaries due to the long lengths of ASR transcripts.

Next, identified summary clusters are sequentially concatenated and concatenated summaries containing 5 or fewer words are stemmed. We observed that summaries which are not merged and contain few words are very frequently speech artifacts that contribute no value. Pseudocode is given in Algorithm 2.

Table 5: Automatic evaluation heuristic scores for various segmentation strategies on *Long Summaries* compared to `M3` using Coreferenced Semantic Segmentation.

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

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

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

5 EVALUATION

We performed user studies to evaluate the following:

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

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

5.1 Methodology

We recruited 10 recent university graduates

from diverse professions (5 women, average age = 26) for our study. Each study lasted 1.2 to 2 hours and averaged 1.5 hours; subjects were paid $20 per hour for their time. To begin, participants were provided with a scenario where a dialog summarization tool would be potentially useful: *imagine you get an email from a friend or colleague about an exciting interview on YouTube about "Diversity and Inclusion in the Workplace." It's 23 minutes long, you're not sure if you want to commit to watching the whole thing, but you want to know if there's anything new or interesting in it. We're trying to help people explore audio clips to find key takeaways.* They were also informed that the summaries were generated by an AI and might be imperfect.

Participants were then given a link to the interface with the audio for "Diversity and Inclusion" loaded in. During the warm-up, we explained the different UI components, *Short, Medium,* and *Long* levels of summarization, the original transcript, and the media button to play and scan the corresponding original audio section. Figure 1 is an example of what the user would see. To familiarize users with the system, we instructed them to read the first *Short Summary*, its corresponding *Medium Summary*, *Long Summary*, and ASR transcript segment, as well as to play the audio segment.

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

(1) `Short Summary Quality Assessment`: assess *Short Summary* quality for two audio files ("Diversity and Inclusion" and one of their choice). For `Short Summary Assessment`, participants were asked to think aloud so we could understand how participants built an intuition and what their interpretation of the system was like. We asked participants to rate each *Short Summary* for two things: 1) grammatical correctness and 2) semantic comprehensibility. Users answered "yes" or "no" to each question. For semantic comprehensibility, we ask them whether they were able to understand the *Short Summary* and if it matched the corresponding audio segment's content. Users were able to check for semantic meaning by comparing against *Medium, Long Summaries*, original ASR transcripts, and audio.
(2) `Short Summary Error Recovery`: if participants were confused on a *Short Summary*'s meaning, they were asked if 1) they could regain comprehension of the *Short Summary*'s meaning and 2) what system features they used to recover the meaning. Participants were allowed to spend as much time as they needed to rate all the *Short Summaries* for the two audio files and were encouraged to think aloud as much as possible.
(3) `Practical Usage Assessment`: use the system as they would in an every day situation. Participants were asked to choose the audio file that interested them most from the 5 remaining audio files in Table 1 and to use the system to find interesting information in that dialog quickly. They used the system without any restrictions and without thinking aloud. We timed participants' usage and observed their browsing strategies.

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

5.2 Results

5.2.1 Short Summary Accuracy. During the study participants rated a total of 556 *Short Summaries*. The overall average accuracy across all users and recordings was 71.4% (see Table 6). The overall accuracy includes both grammatical correctness (84.9%) and semantic comprehensibility (75.9%). Overall accuracy is a measure of how many *Short Summaries* had any kind of error (either grammatical or semantic).

Table 6: Average *Short Summary* accuracy and standard deviations.

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

An overall accuracy of 71.4% means that many *Short Summaries* can be read and understood without any issues. The system's grammatical correctness is reasonably high (84.9%), but the system's semantic comprehensibility is lower (75.9%). Generally, grammatical errors are not detrimental to user experience because most grammatical errors do not distort the meaning of the sentence. However, poor semantics often requires users to investigate further to comprehend the meaning [Kaschak and Glenberg 2000]. In this study, to fully comprehend every *Short Summary*, users would need to investigate semantic errors for 1 in 4 *Short Summaries*.

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

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

The hierarchical summarization features of the interface were designed to help users recover from errors quickly and easily. We wanted to know to what degree participants used these features during recovery. We found that participants used all the hierarchical summarization features to some degree. Participants had two main styles of using the summarization: either they traversed down the hierarchy in order (from most summarized to least summarized forms of the semantic chunk or skipped to their preferred source of information. Table 7 shows the breakdown of how often each feature was used. Participants used *Medium Summaries* a small amount (in 11.5% of recoveries) and used *Long Summaries* more (in nearly 20% of recoveries). However, participants used the ASR Transcript Segment the most (for 40.8% of recoveries.) This behavior is explained by users noting how *Medium, Long Summaries* were either too similar or contained the same semantic errors as *Short Summaries*. As a result, users defaulted to reading the ASR Transcript Segment more often.

Table 7: Distribution of the hierarchical levels users explored in order to recover from an inaccurate *Short Summary*.

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

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

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

5.2.3 Time Savings. In the `Practical Usage Assessment`, when participants used the system at their own pace (without thinking aloud or rating tasks), we found that on average, the time participants spent was **27.1**% of the original audio time to reach a level of understanding that they were satisfied with (Table 8). The fastest user spent only 6.9% of the original audio time, and the slowest spent 75.9% of the original audio time, but most spent between 20% and 30% of the original audio time. We believe the system presents a sizable time savings. Although users do have strategies for processing audio faster (such as listening at 1.25x speed or skipping the first minute of the audio), these strategies are unlikely to provide dramatic speed ups and often lack the control and freedom that our system provides.

How much time users spent varied with what level of detail they wanted to understand. The two users with the highest time percentage (75.9% and 45.5%) were looking for details. The other users took much less time and wanted a cursory understanding of the material (Table 8) such as getting a broad gist of the conversation, wanting a few key takeways, or wanting only a specific piece of information. See Section 5.2.4 for more details.

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

While browsing for interesting nuggets of information, users sometimes found the *Short Summary* sufficient, but often leveraged

Table 8: Individual participant times using the system as percentage of original audio length alongside intended browsing style.

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

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

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

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

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

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

6 LIMITATIONS AND FUTURE WORK

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

6.1 ASR Limitations

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

Currently, the system is designed for two-person audio dialogues and performs particularly well for interviews where one person is creating a majority of the information content and narrative. There are many other types of speech and audio that future work could extend to: political debates, city council meetings, project discussions, doctors appointments, quarterly earnings calls, sporting events, documentaries, and instructions as they all typically share an information dense characteristic. Extending this approach to these areas has many technical challenges including connecting audio to video, increasing ASR accuracy for multi-person audio, and tracing multiple threads of conversation across people. Lastly, there are also design and privacy considerations around summarizing personal audio. Future work should explore the trade-offs between the benefits of reviewing personal audio and the privacy risks.

6.2 Summarization Language Model Limitations

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

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

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

7 CONCLUSION

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

While prior work has established the benefits of hierarchical browsing of information [Zhang et al. 2017] [Truong et al. 2021], we demonstrate that automatic summarization is now feasible through large scale language models. Moreover, hierarchical browsing can help people skim information and recover from errors made by

automatic tools. Interestingly, we also observe that users can apply *Short Summaries* as a navigational tool to identify interesting parts of audio recordings and drill deeper. From a technical perspective, our system achieves practical summarization accuracy of spoken dialog without custom data sets [Karlbom and Clifton 2020] or subsequent retraining of large language models. Rather, by using an improved segmentation approach we were able to employ out of the box industry standard models to produce readable and meaningful summaries in a novel traversable interface.

REFERENCES

Connelly Barnes, Dan B Goldman, Eli Shechtman, and Adam Finkelstein. 2010. Video tapestries with continuous temporal zoom. In *ACM SIGGRAPH 2010 papers*. 1–9.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. 562–569.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640* (2018).

Yashesh Gaur, Walter S Lasecki, Florian Metze, and Jeffrey P Bigham. 2016. The effects of automatic speech recognition quality on human transcription latency. In *Proceedings of the 13th Web for All Conference*. 1–8.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. *Proceedings of the 2nd Workshop on New Frontiers in Summarization* (2019). https://doi.org/10.18653/v1/d19-5409

Dan B Goldman, Brian Curless, David Salesin, and Steven M Seitz. 2006. Schematic storyboarding for video visualization and editing. *Acm transactions on graphics (tog)* 25, 3 (2006), 862–871.

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

Dan Jackson, James Nicholson, Gerrit Stoeckigt, Rebecca Wrobel, Anja Thieme, and Patrick Olivier. 2013. Panopticon: A parallel video overview system. In *proceedings of the 26th annual ACM symposium on User interface software and technology*. 123–130.

Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. 2019. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science* 157 (2019), 160–167.

Hannes Karlbom and Ann Clifton. 2020. Abstractive Podcast Summarization using BART with Longformer attention. (2020).

Michael P Kaschak and Arthur M Glenberg. 2000. Constructing meaning: The role of affordances and grammatical constructions in sentence comprehension. *Journal of memory and language* 43, 3 (2000), 508–529.

Joshua Y. Kim, Chunfeng Liu, Rafael A. Calvo, Kathryn McCabe, Silas C. R. Taylor, Björn W. Schuller, and Kaihang Wu. 2019. A Comparison of Online Automatic Speech Recognition Systems and the Nonverbal Responses to Unintelligible Speech. arXiv:1904.12403 [cs.SD]

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

Douglas W. Maynard. 1980. Placement of topic changes in conversation. 30, 3-4 (1980), 263–290. https://doi.org/doi:10.1515/semi.1980.30.3-4.263

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. arXiv:2005.00661 [cs.CL]

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*. 181–190.

Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video digests: a browsable, skimmable format for informational lecture videos.. In *UIST*, Vol. 10. Citeseer, 2642918–2647400.

Peter. Pirolli. 2007. *Information foraging theory : adaptive interaction with information / Peter Pirolli.* Oxford University Press New York. 204 p. : pages. http://www.loc.gov/catdir/toc/ecip0617/2006021795.html

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

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

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics* 27, 4 (2001), 521–544.

Veselin Stoyanov and Claire Cardie. 2006. Partially Supervised Coreference Resolution for Opinion Summarization through Structured Rule Learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sydney, Australia, 336–344. https://aclanthology.org/W06-1640

Ryuichi Takanobu, Minlie Huang, Zhongzhou Zhao, Feng-Lin Li, Haiqing Chen, Xiaoyan Zhu, and Liqiang Nie. 2018. A Weakly Supervised Method for Topic Segmentation and Labeling in Goal-oriented Dialogues via Reinforcement Learning.. In *IJCAI*. 4403–4410.

Anh Truong, Peggy Chi, David Salesin, Irfan Essa, and Maneesh Agrawala. 2021. Automatic Generation of Two-Level Hierarchical Tutorials from Instructional Makeup Videos. (2021).

Aneesh Vartakavi and Amanmeet Garg. 2020. PodSumm–Podcast Audio Summarization. *arXiv preprint arXiv:2009.10315* (2020).

René Witte and Sabine Bergler. 2003. Fuzzy coreference resolution for summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*. 43–50.

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

Haijun Xia. 2020. Crosspower: Bridging Graphics and Linguistics. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 722–734.

Haijun Xia, Jennifer Jacobs, and Maneesh Agrawala. 2020. Crosscast: Adding Visuals to Audio Travel Podcasts. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 735–746.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive text summarization. *arXiv preprint arXiv:1910.14142* (2019).

Amy X Zhang, Lea Verou, and David Karger. 2017. Wikum: Bridging discussion forums and wikis using recursive summarization. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 2082–2096.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs.CL]

Zheng Zhao, Shay B Cohen, and Bonnie Webber. 2020. Reducing Quantity Hallucinations in Abstractive Summarization. *arXiv preprint arXiv:2009.13312* (2020).

Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, and Ling Fan. 2020. A Two-Phase Approach for Abstractive Podcast Summarization. arXiv:2011.08291 [cs.CL]

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. *arXiv preprint arXiv:2004.02016* (2020).

A ALGORITHM PSEUDOCODE

A.1 Coreferenced Semantic Segmentation Pseudocode

Algorithm 1: Coreferenced Semantic Segmentation

Input: List $\mathbf{T}_{in}$ of n speaker turn segments $\{S\}_{i=0}^{n}$ where word $w \in$ sentence $s \in S$, hyperparameters $m = 100$ for maximum coreference word span and $p = 3$ for minimum number of coreference mentions.

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

A.2 Hierarchical Concept Clustering and Merging Pseudocode

Algorithm 2: Hierarchical Concept Clustering and Merging

Input: List of summaries (sentences) $s \in S_i \in \mathbf{S}$ in speaker turn S_i for all speaker turns $\mathbf{S}$, Embedding `Sentence Transformer` Model M, $cut\_off = 5$ determining the smallest concatenated summary allowable

$L_{out} \leftarrow$ list();
for $S_i \in \mathbf{S}$ **do**
  $\mathbf{E} \leftarrow M.embed(s \in S_i)$ *# embed all summaries within current speaker turn*;
  $\mathbf{D} \leftarrow$ create_pairwise_cosine_distance($\mathbf{E}, \mathbf{E}$);
  $labels \leftarrow$ agglomerative_clustering($\mathbf{D}$);
  $cluster \leftarrow [s \in S_i].group\_concatenate(labels)$;
  $cluster_{filtered} \leftarrow [s >$ cut_off $\in S_i]$;
  $L_{out}.append([cluster_{filtered}])$;
return L_{out}

The final level of *Medium* to *Short* summaries contain far fewer summaries then *ASR Transcript* to *Long Summary* and *Long Summary* to *Medium Summary* due to previous merges. We remove the proximity constraint from *Medium* to *Short* and allow agglomerative clustering is across all summaries instead of within speaker turns.

B HEURISTIC SCORE DISCUSSION

B.1 Heuristic Score Motivation

In evaluation, a longer reference text (an ASR segment) and a model's generated summary are passed in as inputs to an evaluation model that computes a numerical score describing the summary's accuracy. Usually a text generation's quality estimation focuses on adequacy (content faithfulness) and fluency (coherence). Unlike typical evaluation, our dialog's evaluation setting is unsupervised which poses an extra set of challenges. Our heuristic adopts two individually unsupervised components, `BERTScore` and `Sentence Transformer` to measure adequacy (information retention, *or semantic content*) and to a lesser degree fluency (coherence *or grammaticality*). Evaluating a generated summary's fluency without a reference is notably challenging (discussed below) but is, to some extent captured, within `BERTScore`'s token embedding matching.

B.2 Heuristic Score Trade-offs

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

Because of the lack of human authored summaries in both training and evaluation data, we are unable to use standard summarization evaluation standards such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin 2004]. Moreover, existing *n*-gram evaluation metrics cannot be effectively used on summaries as they tend to fail at robustly capturing performance; shortened paraphrases containing semantically critical changes in word positions may be unfairly penalized due to a mismatch with the reference text.

One subsequent trade-off is the possibility of achieving adequacy at the cost of low fluency. Recall that semantic content is compared to an entire (unsummarized) dialog segment as a reference; thus a model can theoretically fool the metric by outputting many *shorter* and *less relevant* segments in an incoherent manner instead of a single relevant segment to achieve a better score, as is the case with **`M2`**'s marginally better score over **`M3`**'s. To ensure that this is not the case, we also qualitatively inspect our formative study's findings in Section 3.4.