# Towards Abstractive Grounded Summarization of Podcast Transcripts

**Kaiqiang Song,**[†‡] **Chen Li,**[‡] **Xiaoyang Wang,**[‡] **Dong Yu,**[‡] **Fei Liu**[†]

[‡]Tencent AI Lab, Seattle, WA
[†]University of Central Florida, Orlando, FL

{riversong,ailabchenli,shawnxywang,dyu}@tencent.com
feiliu@cs.ucf.edu

## Abstract

Podcasts have shown a recent rise in popularity. Summarization of podcasts is of practical benefit to both content providers and consumers. It helps people quickly decide whether they will listen to a podcast and/or reduces the cognitive load of content providers to write summaries. Nevertheless, podcast summarization faces significant challenges including factual inconsistencies of summaries with respect to the inputs. The problem is exacerbated by speech disfluencies and recognition errors in transcripts of spoken language. In this paper, we explore a novel abstractive summarization method to alleviate these issues. Our approach learns to produce an abstractive summary while grounding summary segments in specific regions of the transcript to allow for full inspection of summary details. We conduct a series of analyses of the proposed approach on a large podcast dataset and show that the approach can achieve promising results. Grounded summaries bring clear benefits in locating the summary and transcript segments that contain inconsistent information, and hence improve summarization quality in terms of automatic and human evaluation.

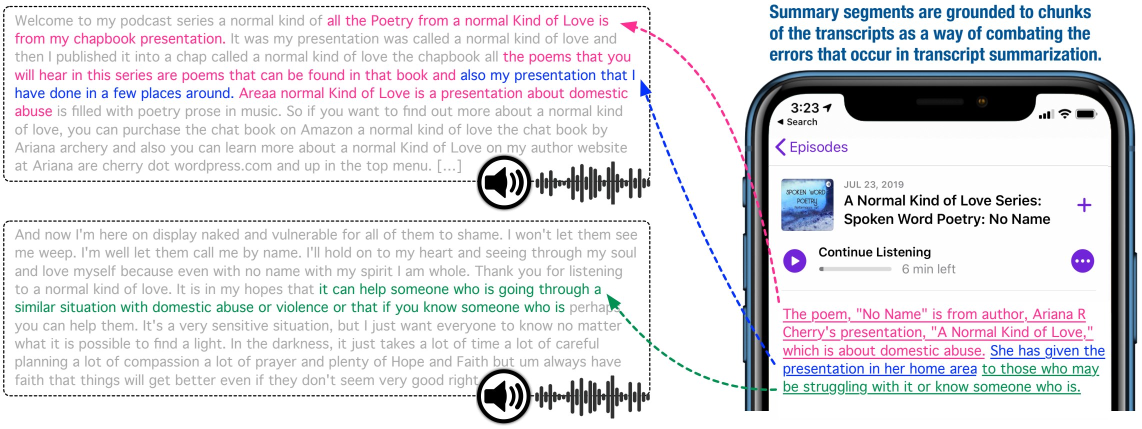
## 1 Introduction

Podcasts are one of the most popular forms of new media. As of today, over 155 million people listen to a podcast every week (Christian, 2021). With the growing interest, there is an increased demand for textual summaries that foretell the content of podcasts. Those summaries help people decide, in a few seconds, if they will listen to a podcast or subscribe to the channel. They are helpful for users who want to find podcasts previously listened to. Furthermore, they can be re-purposed for social media posts or email marketing campaigns, enabling content creators to make their podcasts accessible to a larger audience.

It is desirable to generate *grounded* summaries from podcast transcripts, where spans of summary text are closely tethered to the original audio. Figure 1 provides an example of a grounded abstractive summary. When a user clicks on a summary segment, she will be directed to an audio clip that gives further detail of the conversational context. Grounded summaries give us a preview of notable podcast clips (Shalom, 2019) and they may further release summarization service providers from potential legal claims by directing users to the original audio. This is because, speech recognizers induce transcription errors and abstractive summarization models may hallucinate facts that are not entailed by the original (Kryscinski et al., 2020), both can cause podcast summaries to contain misleading or inaccurate information. With grounded summaries, users are able to frame, interpret, and place into context any system-generated summaries, thus reducing the barriers to deploy podcast summarization technology.

One may attempt to align summary text and podcast transcripts in a post-processing step to generate grounded summaries. Unfortunately, hallucinations do not allow for proper alignments as they are not found in the transcripts (Maynez et al., 2020). Hierarchical attention models may seem promising for this task (Liu and Lapata, 2019). However, the excessive length of the transcripts makes it difficult to produce attention distributions over the entire transcripts. Recent evidence suggests that attention weights are not reliable indicators of the relative importance of inputs (Jain and Wallace, 2019), thus it remains an open question whether attention can be used to find alignments between transcripts and summary segments.

In this paper, we seek to generate grounded summaries from podcast transcripts by exploring an *on-demand* abstractive summarizer. It mimics how a human might approach a lengthy transcript – the expert would identify a portion of the transcript that is deemed most important and relevant to the existing summary, use it as a ground to produce a new



Welcome to my podcast series a normal kind of all the Poetry from a normal Kind of Love is from my chapbook presentation. It was my presentation was called a normal kind of love and then I published it into a chap called a normal kind of love the chapbook all the poems that you will hear in this series are poems that can be found in that book and also my presentation that I have done in a few places around. Areaa normal Kind of Love is a presentation about domestic abuse is filled with poetry prose in music. So if you want to find out more about a normal kind of love, you can purchase the chat book on Amazon a normal kind of love the chat book by Ariana archery and also you can learn more about a normal Kind of Love on my author website at Ariana are cherry dot wordpress.com and up in the top menu. [...]

And now I'm here on display naked and vulnerable for all of them to shame. I won't let them see me weep. I'm well let them call me by name. I'll hold on to my heart and seeing through my soul and love myself because even with no name with my spirit I am whole. Thank you for listening to a normal kind of love. It is in my hopes that it can help someone who is going through a similar situation with domestic abuse or violence or that if you know someone who is perhaps you can help them. It's a very sensitive situation, but I just want everyone to know no matter what it is possible to find a light. In the darkness, it just takes a lot of time a lot of careful planning a lot of compassion a lot of prayer and plenty of Hope and Faith but um always have faith that things will get better even if they don't seem very good right

Summary segments are grounded to chunks of the transcripts as a way of combating the errors that occur in transcript summarization.

The poem, "No Name" is from author, Ariana R Cherry's presentation, "A Normal Kind of Love," which is about domestic abuse. She has given the presentation in her home area to those who may be struggling with it or know someone who is.

Figure 1: An example of a *grounded* summary where spans of summary text are tethered to the original audio. The user can tap to hear the audio clip, thus interpreting a system-generated summary in context.

piece of the summary, and that process is repeated until the summary is finished. Our summarizer employs a novel regularization technique that enables it to visit portions of the transcript in chronological order, while allowing zigzags in order to produce a coherent summary. This has another implication. It implies that we may estimate what percentage of a podcast transcript is covered by the summary and thus adjust that when necessary.

Distinguishing our work from earlier research on extract-then-abstract methods (Hsu et al., 2018; Chen and Bansal, 2018; Gehrmann et al., 2018; Lebanoff et al., 2019; Jin et al., 2020; Pilault et al., 2020), we require selected transcript chunks to have high salience, but also those salient content must appear at the beginning of the selected chunks, so that the corresponding audio clips can provide good *jump-in points* for users to start listening. Our experiments are performed on a large podcast summarization dataset containing over 100,000 English podcasts (Clifton et al., 2020). We show that our proposed grounded summarizer can perform competitively or better than the state-of-the-art methods, including the recent methods that leverage large, pretrained models (Lewis et al., 2020; Beltagy et al., 2020) as judged by automatic metrics and human evaluation. Our contributions in this paper are as follows.

- We address the problem of podcast summarization by investigating an on-demand summarizer that produces grounded abstracts. The abstracts help users quickly decide if they will listen to the podcasts and offer a sampler of salient podcast clips. The on-demand summarizer does not need to encode the entire transcript, hence substantially reduces the GPU memory footprint.
- We conduct a series of analyses to gain insights into the impact of specific design decisions. They include how a transcript chunk should be defined, whether those transcript chunks overlap, to what extent the summary content is taken verbatim from selected chunks, and how the summary may be extended to cover more information.
- Through extensive experiments on a benchmark podcast dataset, we demonstrate the effectiveness of our proposed approach and show results that are comparable to human writer performance. The approach opens an avenue towards generating a new kind of abstractive summaries that allow users to verify the information consistency of summary parts against the original audio clips.[1]

## 2 Related Work

With the rapid rise of podcasts comes the need for automatic summarization of podcast transcriptions. While comparatively understudied, recent work has shown great progress. Clifton et al. (2020) present the Spotify dataset that was adopted in TREC 2020 for the podcast summarization task.[2] Our participating system in TREC 2020 focuses on identifying salient segments from transcripts and using them as input to an abstractive summarizer (Song et al., 2020). Reddy et al. (2021) develop classifiers to detect and eliminate extraneous marketing materials in podcasts to aid summarization. In this paper, we explore techniques that generate *grounded podcast summaries* where pieces of summary text are tied to short podcast clips.

[1] Our model and code have been made publicly available: https://github.com/tencent-ailab/GrndPodcastSum

[2] https://trec.nist.gov/data/podcast2020.html

One of the most serious problems of neural abstractive summarization is that the summaries can contain factually incorrect information and hallucinations (Falke et al., 2019; Kryscinski et al., 2020; Maynez et al., 2020; Lebanoff et al., 2020). Without grounded summarization, users have to listen to the full episodes to find connections between details of the summaries and the original podcasts. If successful, grounded summaries will benefit a number of summarization tasks where the input involves lengthy transcripts, including meetings (Li et al., 2019; Koay et al., 2020, 2021; Zhong et al., 2021), medical conversations (Liu and Chen, 2019), interviews (Zhu et al., 2021), livestreams (Cho et al., 2021) and more.

An extract-then-abstract strategy could be used to produce grounded abstractive summaries (Chen and Bansal, 2018; Gehrmann et al., 2018; Hsu et al., 2018; Jin et al., 2020; Pilault et al., 2020). Most of these approaches are tailored to written documents, e.g., news, Wikipedia, and scholarly articles. They extract sentences from the documents and use them as input to an abstractive summarization model to produce a summary. Nevertheless, transcripts of spoken language lack essential document structure such as sentence, paragraph and section boundaries, making it unclear how these approaches will perform on podcasts.

Attention provides another mechanism for aligning the summary and transcript segments. The use of sparse attention allows a summarization model to potentially scale to longer documents (Beltagy et al., 2020; Kitaev et al., 2020; Huang et al., 2021). Hierarchical Transformer encodes multiple paragraphs in a hierarchical manner to allow them to exchange information (Liu and Lapata, 2019; Fabbri et al., 2019; Chen and Yang, 2020). However, it is shown that attention weights are not reliable indicators of the relative importance of inputs, as alternative attention distributions would have yielded similar results (Jain and Wallace, 2019).

Our approach in this paper is to better align summary segments with chunks of the transcripts to allow easy tracing of inconsistent information. It features a generator that writes a summary from beginning to end, and a savvy selector that knows when to switch to a new transcript chunk and where to switch to. Differing from PG networks (See et al., 2017) and retrieval-augmented generation (Guu et al., 2020; Lewis et al., 2021), our selector places heavy emphasis on modeling and selection of transcript chunks. A desirable chunk is expected to be about 2 minutes long and places important information at the beginning to enable easy user verification. In the following section, we present details of the model implementation.

## 3 Our Approach

A major challenge facing podcast summarization is the dramatic length difference between source and target sequences. At a speaking rate of 122 words per minute for spontaneous speech (Polifroni et al., 1991), the full transcript of a 1-hour long episode contains roughly 7,000 words and that of a 1.5-hour long episode could reach 10,000 words. In contrast, a podcast summary is short, containing on average 61 words according to Manakul and Gales (2020). The ratio of their lengths could reach as high as 100-to-1, and this motivates our study of abstractive *grounded* summarization where summary segments are grounded to selected chunks of transcripts as a way of combating the inevitable errors that occur in podcast summarization.

Let $\mathbf{x}$ be the sequence of tokens in the source transcript and $\mathbf{y}$ be the sequence of tokens in the summary. These tokens share the same vocabulary $\mathcal{V}$. We use $\mathbf{x}_{\mathcal{C}}$ to denote a chunk of the transcript, and $\mathcal{C}$ gives the indices of tokens that belong to the chunk. The full transcript can be decomposed into a sequence of chunks, denoted by $\{\mathcal{C}_1, \cdots, \mathcal{C}_M\}$. The chunks may have varying sizes and overlap with each other; they are the grounds for generating a podcast summary. Our assumption is twofold. Firstly, we assume a summary segment is produced by conditioning on the previously generated tokens ($\mathbf{y}_{<j}$) and a specific chunk of the transcript. Secondly, there exists a function $\mathcal{G}(\mathbf{x}, \mathbf{y}_{<j})$ (Eq. (1)) that determines the most appropriate grounding chunk for generating all tokens of the segment. Particularly, when the entire transcript is treated as a single chunk, it reduces to the standard conditional generation model $p_\theta(y_j|\mathbf{y}_{<j}, \mathbf{x})$.

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{N} p_\theta(y_j|\mathbf{y}_{<j}, \mathcal{G}(\mathbf{x}, \mathbf{y}_{<j})) \qquad (1)$$

Thus, the crucial point is a coarse segmentation of the source transcript and an alignment between the transcript chunks and summary segments. In this work we use a sliding window to produce transcript chunks, with window size $\mathcal{W}$ and stride size

$\mathcal{S}$.[3] The sizes can be measured in terms of tokens. E.g., $\mathcal{W}$=256 and $\mathcal{S}$=128 tokens will produce a series of fixed-length chunks that overlap with each other. The rationale for using overlapping chunks is to find those that serve both as grounds for summary generation and good jump-in points for user verification. The sizes can also be measured by the number of sentences. E.g., $\mathcal{W}$=20 and $\mathcal{S}$=20 sentences produce a set of varying-length, non-overlapping chunks. In spoken language, a series of consecutive short sentences often indicates the content is relatively unimportant (Marge et al., 2010).

Given a summary segment $\widetilde{\mathbf{y}}$, we designate $\mathbf{x}_{\mathcal{C}}$ as a **grounding chunk** if it attains the highest score $\mathcal{S}(\mathbf{x}_{\mathcal{C}}, \widetilde{\mathbf{y}})$ (Eq. (2)). This position-biased coverage score favors the transcript chunk that covers summary bigrams and puts summary content at the beginning to aid humans in performing content verification. It measures the percentage of unique summary bigrams $\mathcal{B}(\widetilde{\mathbf{y}})$ covered by a chunk $\mathbf{x}_{\mathcal{C}}$. Particularly, $\mathbb{I}\left[b_k \in \mathbf{x}_{\mathcal{C}}\right]$ is an indicator that returns 1 if the bigram $b_k$ appears in $\mathbf{x}_{\mathcal{C}}$ and 0 otherwise. Each bigram $b_k$ has an associated weight $w_k$ (Eq. (3)). If it appears in the first position of $\mathbf{x}_{\mathcal{C}}$ ($\texttt{pos}_k = 0$), it receives a weight of one. Otherwise, the weight is decayed according to the relative position of the bigram's first occurrence in the chunk ($\texttt{pos}_k$) and $\gamma$ is a coefficient for the decay.[4]

$$\mathcal{S}(\mathbf{x}_{\mathcal{C}}, \widetilde{\mathbf{y}}) = \frac{1}{|\mathcal{B}(\widetilde{\mathbf{y}})|} \sum_{b_k \in \mathcal{B}(\widetilde{\mathbf{y}})} w_k \, \mathbb{I}\left[b_k \in \mathbf{x}_{\mathcal{C}}\right] \quad (2)$$

$$w_k = 1 - \gamma \frac{\texttt{pos}_k}{|\mathcal{C}|}; \quad \gamma \in [0, 1] \quad (3)$$

We proceed by training a neural encoder-decoder model to generate an abstractive summary from the grounding transcript chunks. Each segment of the summary (= sentence)[5] is generated conditioned on its grounding chunk $\mathbf{x}_{\mathcal{C}}$ and all the previously generated tokens $\mathbf{y}_{<j}$. The process starts from the first chunk of the transcript $\mathbf{x}_{\mathcal{C}_1}$. The encoder converts this grounding chunk into a sequence of hidden vectors $[\mathbf{h}_1^{\mathcal{C}}, \ldots, \mathbf{h}_m^{\mathcal{C}}]$ (Eq. (4)). The decoder predicts the next summary token $y_j$ (Eq. (5)) and continues to do so until a "*switch point*" is detected. At this point the current summary segment is finished and the decoder is poised to select the next transcript chunk $\mathbf{x}_{\mathcal{C}_{\text{new}}}$ and generate a new summary segment from it. The decoding process finishes when a special symbol ([sep]) is predicted that indicates the end of the summary.

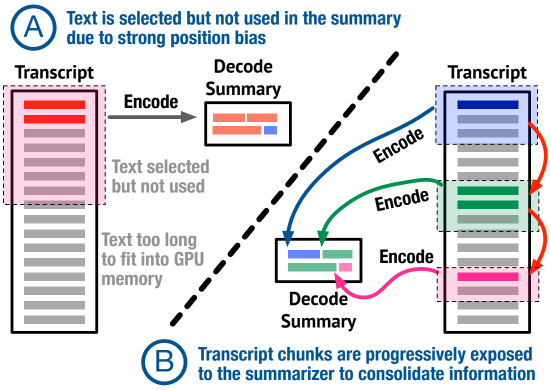

A
Text is selected but not used in the summary due to strong position bias
Transcript
Encode
Decode Summary
Text selected but not used
Text too long to fit into GPU memory
Transcript
Encode
Switch
Encode
Switch
Encode
Decode Summary
B
Transcript chunks are progressively exposed to the summarizer to consolidate information

Figure 2: Strong position bias can cause the abstractor to use only content at the beginning of the input to generate a summary. By exposing the chunks progressively, our approach makes use of this characteristic to consolidate information from multiple transcript chunks.

$$[\mathbf{h}_1^{\mathcal{C}}, \ldots, \mathbf{h}_m^{\mathcal{C}}] = \texttt{Encode}(\mathbf{x}_{\mathcal{C}}) \quad (4)$$

$$y_j = \texttt{Decode}(\mathbf{y}_{<j}, [\mathbf{h}_1^{\mathcal{C}}, \ldots, \mathbf{h}_m^{\mathcal{C}}]) \quad (5)$$

$$\mathcal{G}(\mathbf{x}, \mathbf{y}_{<j}) = \begin{cases} \mathbf{x}_{\mathcal{C}_1}, & j = 1 \\ \mathbf{x}_{\mathcal{C}_{\text{new}}}, & j > 1 \texttt{ \& switch} \\ \mathcal{G}(\mathbf{y}_{<j-1}), & j > 1 \texttt{ \& no-switch} \end{cases}$$

There is a notable difference between our approach and most extract-then-abstract approaches that select important sentences from the document and provide them to the abstractor all-at-once. As illustrated in Figure 2, strong position bias causes the abstractor to use only content at the beginning of the input to generate a summary. By exposing the chunks progressively, our approach naturally makes use of this characteristic to consolidate information from multiple source chunks. It reduces the amount of computation necessary to train the encoder-decoder model, as only selected transcript chunks are encoded which is equal to the number of summary segments. Moreover, it is possible to encourage the summary to have a good coverage

[3] Discourse segmentation is beyond the scope of this work. There is little to no data available to build a discourse segmentation tool and little existing work on discourse analysis of podcasts. We refer the reader to Joty et al. (2019) for recent advances in discourse processing research.

[4] If a summary segment cannot be mapped to a chunk using Eqs. (2-3), we perform the following: $\widetilde{\mathbf{y}}$ is assigned to the first chunk $\mathcal{C}_1$ if it is the first segment of the summary. Otherwise, $\widetilde{\mathbf{y}}$ is assigned to the same chunk as the previous summary segment to improve coherence. We require $\mathbf{x}_{\mathcal{C}}$ and $\widetilde{\mathbf{y}}$ to have a minimum of four shared bigrams (stopwords-only bigrams are excluded). Future work may consider aligning transcripts and summaries based on propositions (Ernst et al., 2020).

[5] We use sentences as summary segments; other sentence-like segments are possible in future work.

of the source content by specifying a minimal set of grounding chunks to be used for generation.

**Regularizing Chunk Selection.** Learning function $\mathcal{G}(\mathbf{x}, \mathbf{y}_{<j})$ that predicts a transcript chunk $\mathbf{x}_\mathcal{C}$ to switch to is crucial for success at inference time. Let there be $M$ transcript chunks and $N$ summary segments in a training instance. We define $p_j^c$ to be the model probability that the $c$-th chunk is predicted as the ground for generating the $j$-th summary segment; $c^*$ is the gold chunk obtained using Eq. (2-3). Our learning objective is a cross-entropy loss against the gold labels with a novel regularizing term $\mathcal{R}$ to enable chunks to be selected as per their original order in the transcript, while allowing zigzags to produce a coherent summary (Eq. (6-7)).

$$\mathcal{L}(\phi) = -\textstyle\sum_{j=1}^{N} \log p_j^{c^*} + \alpha\mathcal{R} \tag{6}$$

$$\mathcal{R} = \tfrac{1}{N} \textstyle\sum_{j=1}^{N} \sum_{c=1}^{M} \max(0, s_{j+1}^c - s_j^c) \tag{7}$$

Particularly, $s_j^c = \sum_{c'=1}^{c} p_j^{c'}$ denotes the sum of the probability assigned to all chunks up to the $c$-th position, in order to generate the $j$-th summary segment. We encourage $\sum_{c=1}^{M} \max(0, s_{j+1}^c - s_j^c)$ to be a small value so that if a chunk (up to the $c$-th position) is assigned to the $j$-th summary segment, it is unlikely to be assigned to the $(j+1)$-th segment. $\mathcal{R}$ is designed to regularize the loss and penalize violations; $\alpha$ is its coefficient which will be tuned on the validation set.

Given a partial summary $\mathbf{y}_{<j}$, selecting the next transcript chunk depends on two factors. Firstly, it should be a chunk that contains salient content at its beginning. We use $\mathcal{I}(\mathbf{x}_\mathcal{C})$ to denote the importance of the chunk. It is obtained by encoding the chunk into a vector $\mathbf{h}_{\mathbf{x}_\mathcal{C}}$ using RoBERTa (Liu et al., 2019), then apply a feedforward network to it to estimate the importance (Eq. (9)).[6]

$$p_j^c \propto \exp(\mathcal{I}(\mathbf{x}_\mathcal{C}) + \mathcal{R}(\mathbf{x}_\mathcal{C}, \mathbf{y}_{<j})) \tag{8}$$

$$\mathcal{I}(\mathbf{x}_\mathcal{C}) = \texttt{FFN}_1(\mathbf{h}_{\mathbf{x}_\mathcal{C}}) \tag{9}$$

$$\begin{aligned} \mathcal{R}(\mathbf{x}_\mathcal{C}, \mathbf{y}_{<j}) = {} & \texttt{FFN}_2([\mathbf{h}_{\mathbf{x}_\mathcal{C}} || \mathbf{h}_{\mathbf{y}_{<j}}]) \\ & + \texttt{LowRank}(\mathbf{h}_{\mathbf{x}_\mathcal{C}}^\top \mathbf{W}\, \mathbf{h}_{\mathbf{y}_{<j}}) \end{aligned} \tag{10}$$

Secondly, the chunk may be relevant to the partial summary $\mathbf{y}_{<j}$. We define the relevance score $\mathcal{R}(\mathbf{x}_\mathcal{C}, \mathbf{y}_{<j})$ to capture two levels of interaction between the candidate chunk, represented by $\mathbf{h}_{\mathbf{x}_\mathcal{C}}$ and the last hidden state of the partial summary, represented by $\mathbf{h}_{\mathbf{y}_{<j}}$. Their linear interaction is captured by a feedforward network ($\texttt{FFN}_2$) and bilinear interaction is modelled by $\mathbf{h}_{\mathbf{x}_\mathcal{C}}^\top \mathbf{W}\, \mathbf{h}_{\mathbf{y}_{<j}}$ where a low-rank approximation is used: $\texttt{LowRank}(p^\top \mathbf{W} q) = (p^\top \mathbf{U})(\mathbf{V}^\top q)$. The score $p_j^c$ is the likelihood that the $c$-th chunk is assigned to the $j$-th summary segment considering saliency and content relevancy.

**Switch Point.** A skilled writer pauses after writing down a sentence. We borrow that intuition to inform the construction of a switch-point predictor. The model combines the last hidden state of the summary sequence $\mathbf{h}_{\mathbf{y}_{<j}}$ and the embedding of the anticipated token $\mathbf{E}(y_j)$, and use a feedforward network $\texttt{FFN}_3$ to predict if the $j$-th decoding step corresponds to a "*switch point*" (Eq. (11)). During training, the last token of each summary sentence is a ground-truth switch point. At inference time, the model predicts a switch point if $p(\texttt{switch})$ exceeds a threshold, at which point we compute $p_j^c$ to decide the next chunk. Note that the model may choose use the same transcript chunk after switching.

$$p(\texttt{switch}) = \sigma(\texttt{FFN}_3([\mathbf{h}_{\mathbf{y}_{<j}} || \mathbf{E}(y_j)])) \tag{11}$$

## 4 Podcast Data

With over 100,000 podcast episodes, the Spotify dataset (Clifton et al., 2020) is one of the largest corpora available for podcast search and summarization. It encompasses a wide range of topics: travel, business, sports, book reviews, mysteries, guided meditations, nutrition and weight loss, among others. Each episode is accompanied by an audio file, an automatic transcript generated by Google's Speech-to-Text API,[7] and metadata provided by the podcast creator. We do not use the audio data in this paper. Our summarizer takes as input a transcript and uses the creator-provided episode description as the reference summary.

**Data Filtering.** Episode descriptions provided by podcast creators show wide variations in quality. When noisy descriptions are used as reference summaries, they can cause a summarizer to hallucinate content. We conduct aggressive filtering of the training data to remove low-quality creator descriptions so as to maintain a balance between the amount of training examples available and quality

[6] The parameters of $\texttt{FFN}_1$ are pretrained on an extraction task that favors chunks that contain summary content at the beginning. For each chunk, we compute its position-biased coverage score (Eq. (2)) against the entire summary. 1/4 of the chunks that yield the highest coverage scores are designated as positive instances, the remaining are negative instances. $\texttt{FFN}_1$ is thus pretrained as a binary classifier.

[7] https://cloud.google.com/speech-to-text

| | |
|---|---|
| **Creator Description** | Tune in as Natalie and Jessica debate physical vs. chemical exfoliation options, and see what our ultimate verdict is on the best type and specific products we love! |
| **hk_uu_podcast1** | In this episode, Jessica and Natalie go head-to-head in the Great Exfoliation Debate! They each advocate their own type of exfoliator and try each other's products to see if they're worth the price difference. They also do a wine pairing and talk about the pros and cons of each of the products they tried. |
| **UCF_NLP2** | In this weeks episode, Jessica and Natalie go head-to-head in the great exfoliation debate. They each advocate for their own type of exfoliator, and then try each other's products for 10 minutes to see what they think. We also talk about the pros and cons of each type of product and recommend a wine to pair with this episode. Santa Julia Winemakers Reserve Mountain Blend . |
| **cued_speechUniv2** | In this episode of the Great Exfoliation Debate, Jessica and Natalie talk about their favorite types of exfoliators and the pros and cons of each of their favorite products. We also do a wine pairing and talk about the benefits and drawbacks of different types of chemical and physical exfoliation products. |
| **GrndAbs-tn** | Natalie and Jessica are back with another episode of Skincare Somali A's. This week we're talking about what we like to call the "Great Exfoliation Debate." We'll also be doing our wine pairing this week. Santa Julia Winemakers' Reserve Mountain Blend (2016) |
| **GrndAbs-to** | This week we are talking about what we like to call the "Great Exfoliation Debate." Because we've got two different points of view and we are going to Duke it out mano a mano this week. We will also of course do our wine pairing because we are your Somali A's and this week we're going with something a little bit more aggressive...a little bit bold. |
| **GrndAbs-sn** | In this episode, Natalie and Jessica debate the pros and cons of exfoliation. Exfoliation is this step in your skincare routine that is taking off all the dead skin cells on your face. And the point of Exfoliating is to reveal brighter, healthier skin while reducing the size of your pores. In this week's episode, we'll be discussing the pros, cons, and what we think is the best way to exfoliate your skin. |
| **GrndAbs-so** | Natalie and Jessica are back to debate the merits of exfoliation. This week, they are going mano a mano and will be debating the pros and cons of using exfoliating on your face. We will also do our wine pairing because we are your Somali A's and this week we're going with something a little bit more aggressive. We would like to recommend Santa Julia Winemakers' Reserve Mountain Blend. That is a Malbec and Cab Franc blend from 2016. It's just a bit of a middle of the road wine but super super tasty |

Table 1: Grounded abstractive summaries (GrndAbs-*) demonstrate a high level of specificity compared to summaries without grounding. The latter contains more generic content. The segments of grounded summaries are tethered to specific transcripts chunks. If a listener finds the summary segment interesting, they can tap to hear the selected segment in context.

of those examples. We clean up reference summaries on the token-, sentence- and summary-level. Tokens that correspond to URLs, email addresses, @mentions, #hashtags, and those excessively long tokens (>25 characters) are directly removed from the summaries. Each sentence in the summary is given a salience score that is the sum of IDF scores of its words. A low score (<10) indicates the sentence contains few informative words and it is thus removed from the summary. Finally, if, after sentence removal, the reference summary is too short or cannot be properly aligned to transcript chunks (§3), the instance is removed from the dataset.[8] This process filters out a substantial amount of low-quality reference summaries, yielding 40,302 episodes in the training set. The Spotify dataset has a standard test set of 1,027 episodes and 179 of them are set for human evaluation.

**Baselines.** Our baselines consist of three of the best performing systems in the TREC 2020 competition on podcast summarization. These systems were judged the best performing by both automatic metrics and human evaluation performed by NIST assessors. All systems make use of the BART-large model (Lewis et al., 2020). The model is tuned first on a news summarization dataset, i.e., CNN/DM or XSum, then fine-tuned on the podcast dataset. Due to the long length of the transcripts, Karlbom and Clifton (2020) describe a combined Longformer-BART model that replaces the BART attention layers with attentions of Longformer (Beltagy et al., 2020); their system is named hk_uu_podcast1. Song et al. (2020) develop an extractive module to select segments from transcripts, then integrate the extractor with BART abstractor to generate summaries (UCF_NLP2). Their baseline (UCF_NLP1) directly truncates the transcript to the first 1,024 tokens. Manakul and Gales (2020) develop a similar baseline (cued_speechUniv3) using the first 1,024 tokens. Further, they perform sentence filtering using a hierarchical attention model (cued_speechUniv1/2/4) and ensembles of models from different data shuffles and checkpoints (cued_speechUniv1/2). In this paper, our system is called GrndAbs for generating grounded abstracts. It has 4 options: -to, -tn, -so, -sn, indicating the sliding window is defined in terms of tokens (-t) or sentences (-s), overlapping (-o) or non-overlapping (-n). We obtain outputs from these competitive baselines and our system to examine both the successes and failures of these attempts.

## 5 Results and Analysis

**Experimental Settings.** Our encoder-decoder model uses BART-large as the base model before fine-tuning it on the podcast dataset. We use the AdamW (Loshchilov and Hutter, 2017) optimizer, where the momentum parameters are set to 0.9 and 0.999. The regularizing coefficient $\alpha$ is tuned on the validation set in the range of $\{0, 0.01, \underline{0.1}, 1\}$.

[8] A summary is required to contain a minimum of 10 BPE tokens and have >2 shared bigrams with all of its grounding chunks. Only words whose IDF scores are greater than 1.2 are considered when computing sentence salience scores.

| Run ID | R-1(%) | R-2(%) | R-L(%) | BertS(%) | BLEURT | SummL |
|---|---|---|---|---|---|---|
| cued_speechUniv1 | 30.54 | 11.25 | 21.05 | 84.17 | -0.7434 | 58.16 |
| cued_speechUniv2 | 30.52 | 11.36 | 21.16 | 84.20 | -0.7491 | 56.93 |
| cued_speechUniv3 | 28.44 | 9.55 | 19.52 | 83.77 | -0.7897 | 55.58 |
| cued_speechUniv4 | 29.00 | 10.42 | 19.95 | 83.99 | -0.7781 | 51.75 |
| UCF_NLP1 | 30.09 | 12.07 | 21.75 | 84.16 | -0.7508 | 57.35 |
| UCF_NLP2 | 30.44 | 11.99 | 21.67 | 84.14 | -0.7382 | 57.85 |
| hk_uu_podcast1 | 29.02 | 10.70 | 20.66 | 84.21 | -0.7992 | 44.63 |
| **GrndAbs-so** | 25.42 | 7.95 | 16.93 | 82.62 | -0.8164 | 80.44 |
| **GrndAbs-sn** | 25.58 | 8.27 | 16.99 | 82.64 | -0.8220 | 78.80 |
| **GrndAbs-to** | 25.79 | 8.38 | 17.15 | 82.67 | -0.8028 | 82.98 |
| **GrndAbs-tn** | 25.79 | 8.25 | 17.20 | 82.71 | -0.8130 | 79.90 |

Table 2: Results on the standard test set containing 1,027 episodes. Our evaluation metrics include ROUGE variants (R-1, R-2 and R-L), BERTScore and BLEURT. We report the length of the summary (SummL) measured in words.

|  | E↑ | G↑ | E+G↑ | Fair↓ | Bad↓ |
|---|---|---|---|---|---|
| cued_speechUniv2 | 22.09 | **51.36** | 73.45 | 22.67 | **3.88** |
| UCF_NLP2 | 22.29 | 46.71 | 69.00 | 20.93 | 10.08 |
| hk_uu_podcast1 | 18.60 | 45.93 | 64.53 | 25.78 | 9.69 |
| creator_description | 13.95 | 42.05 | 46.00 | 30.43 | 13.57 |
| **GrndAbs-tn** | **25.19** | 50.58 | **75.77** | **20.16** | 4.07 |

Table 3: Human evaluation results. 25% of grounded abstractive summaries are rated as *Excellent* and 76% receive a rating of either *Excellent* (E) or *Good* (G).

For summary decoding, we use beam search with a beam size $K$=4 and a length penalty $p$=2. Our sliding window, measured in terms of tokens or sentences, only contain whole sentences. We use the Byte-Pair Encoding (BPE) tokenizer with a vocabulary size $\mathcal{V}$=50,265. For transcripts and reference summaries, we use the SpaCy tool to segment them into sentences (model en_core_web_lg 2.2.5).

**Example Summaries.** In Table 1, we provide a direct comparison of system summaries. This podcast is hosted by Natalie and Jessica who call themselves "*Skincare Sommeliers.*" The episode is named "*The Great Exfoliation Debate*." We find that grounded abstractive summaries (GrndAbs-*) have a higher level of specificity compared to summaries without grounding. Segments of grounded summaries are tied to specific transcripts chunks. If a listener finds a summary segment interesting, they can tap to hear the selected summary segment in context. Our baselines are highly competitive. Their summaries tend to contain more generic content. The description provided by podcast creators is relatively short and at times it does not directly summarize the episode. There are clear benefits in automatic summarization of podcasts, which can reduce the cognitive load and the time it takes for podcast creators to write the summary.

**Automatic Metrics.** In Table 2, we report results on the standard test set containing 1,027 podcast episodes. The metrics include ROUGE (Lin, 2004) variants that compare system summaries with creator descriptions based on n-gram overlap. Further, we experiment with recently developed metrics: BertScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020) that draw on deep neural representations to evaluate generated text. Our approach does not outperform the baselines in ROUGE evaluation against creator descriptions. However, the gap has been substantially reduced when more advanced metrics (BertScore and BLEURT) are considered. There are two possible explanations. First, grounded summaries are about 50% longer than plain abstractive summaries. Their average length is about 80 words per summary, yielding low precision scores. Second, the quality of creator descriptions can be poor. Jones et al. (2020) report only 40% of such descriptions are of Good or Excellent quality, indicating future work may consider creating high-quality ground-truth summaries. Among the four variants of our approach, we observe that their difference is not prominent. The token-based, non-overlapping windows (-tn) variant outperforms others in terms of R-1 and R-L. This system is used in subsequent experiments and analyses.

**Human Evaluation.** It is imperative to perform human evaluation given that creator-provided descriptions are of poor quality and ground-truth summaries are nonexistent. We follow the TREC guidelines to ask human evaluators to assign each summary to one of the four grades: *Excellent*, *Good*, *Fair* and *Poor*. The excellent summary will accurately conveys the most important content of the episode (topical content, genre, and participants).

| System | Q1: People Names | Q2: People Add Info | Q3: Main Topics | Q4: Podcast Format | Q5: Title Context | Q6: Summ Redund | Q7: Good English | Q8: Start/End Points |
|---|---|---|---|---|---|---|---|---|
| creator_description | 60.08 | 50.19 | 80.81 | 59.61 | 57.00 | 16.28 | 88.76 | 60.16 |
| hk_uu_podcast1 | 64.15 | 47.29 | 85.63 | 57.62 | 58.95 | **10.85** | 94.76 | 70.35 |
| UCF_NLP2 | 67.38 | 51.55 | 87.02 | 63.57 | 62.52 | 14.40 | **95.15** | 71.71 |
| cued_speechUniv2 | 69.12 | 50.67 | 87.98 | 64.73 | 63.62 | 12.87 | 94.93 | **77.00** |
| GrndAbs-tn | **75.15** | **64.47** | **89.73** | **69.51** | **66.15** | 17.09 | 94.55 | 73.35 |

Table 4: Average scores per human judgment of 179 testing summaries on 8 Yes/No questions. An assessor quickly skimmed the episode, and made judgments for each summary of the episode. "creator_description" represents the episode description. "cued_speechUniv2," "UCF_NLP2" and "hk_uu_podcast" are the top-3 teams in the Podcast Challenge. Our system "GrndAbs-tn" learns to produce abstractive summary while grounding summary segments in specific portions of the transcript to allow for full inspection of summary details.

| | |
|---|---|
| **Q1** | Does the summary include **names of the main people** (hosts, guests, characters) involved or mentioned in the podcast? |
| **Q2** | Does the summary give any **additional information** about the people mentioned (such as their job titles, biographies, personal background, etc)? |
| **Q3** | Does the summary include the **main topic(s)** of the podcast? |
| **Q4** | Does the summary tell you anything about **the format of the podcast**; e.g. whether it's an interview, whether it's a chat between friends, a monologue, etc? |
| **Q5** | Does the summary give you **more context on the title** of the podcast? |
| **Q6** | Does the summary contain **redundant information**? |
| **Q7** | Is the summary written in **good English**? |
| **Q8** | Are the **start and end of the summary** good sentence and paragraph start and end points? |

Table 5: There are 8 yes-or-no questions asked about the summary quality. An ideal summary should receive a "yes" (1) for all questions but Q6.

It should contain almost no redundant material, be coherent, comprehensible, and has no grammatical errors (Jones et al., 2020). We also asked the human evaluators to answer 8 yes/no questions regarding the quality of the summary as (Jones et al., 2020) suggested, those questions are shown in Table 5. We conduct these experiments on the test set containing 179 podcast episodes as (Jones et al., 2020) did, where each summary is evaluated by five Master workers recruited on the mechanical turk. As shown in Table 3, we find that humans prefer the lengthier grounded abstractive summaries, which substantially outperform all baselines. 25% of grounded abstractive summaries are rated as *Excellent* and 76% of them receive a rating of either *Excellent* or *Good*. Table 4 shows the results of the 8 questions. Comparing to previous best systems, our grounded abstractive summaries have a significant performance gain on retrieving important information including People Names(+6.03%), People Additional Information(+12.92%), Main Topics(+1.75%), Podcast Format(+4.78%) and Title related context(+2.47%) with slight redundancy.

### Chunk Selection and Switch Point Prediction.

We are curious to know how well our system performs on predicting grounding chunks: $\mathcal{G}(\mathbf{x}, \mathbf{y}_{<j})$. In this study, we assume switch points are known and report results on the validation set. Our decoder starts from the first transcript chunk and predicts the next chunk at each switch point. We find that it achieves an accuracy of 86.02% on identifying ground-truth chunks. Next, we examine the performance of switch point prediction. On the validation set, we observe that the predictor achieves 98.75%, 84.95% and 91.33%, respectively, for precision, recall and F-score. Moreover, each summary has an average of 3.67 switch points. A majority of the time (92.42%) the model decides to use the current chunk to continue to decode the next summary segment. At a small percentage (7.58%) the model decides to find to a new grounding chunk. We find 1.24 unique grounding chunks per summary. The statistics suggest that identifying grounding chunks is crucial for summary generation.

**Grounded Summaries.** In Table 7, we measure the percentage of summary n-grams that appear in the transcripts (for all baselines) or grounding chunks (for our approach). While the distributions of unigrams are largely similar, we observe that grounded abstractive summaries tend to reuse more bigrams and trigrams of their grounding chunks. Moreover, for trigrams that are found in the grounding chunks, we find 70% of them tend to appear at the beginning – the front half of the chunks. These results suggest that the grounding chunks identified by our approach can provide effective support for summary generation.

### What Made the Task Challenging?

We manually analyze a large amount of transcripts and their creator descriptions to identify the challenging points of podcast summarization in Table 8:

| | |
|---|---|
| **Excellent** | The summary accurately conveys all the most important attributes of the episode, which could include topical content, genre, and participants. In addition to giving an accurate representation of the content, it contains almost no redundant material which is not needed when deciding whether to listen. It is also coherent, comprehensible, and has no grammatical errors. |
| **Good** | The summary conveys most of the most important attributes and gives the reader a reasonable sense of what the episode contains with little redundant material which is not needed when deciding whether to listen. Occasional grammatical or coherence errors are acceptable. |
| **Fair** | The summary conveys some attributes of the content but gives the reader an imperfect or incomplete sense of what the episode contains. It may contain redundant material which is not needed when deciding whether to listen and may contain repetitions or broken sentences. |
| **Bad** | The summary does not convey any of the most important content items of the episode or gives the reader an incorrect or incomprehensible sense of what the episode contains. It may contain a large amount of redundant information that is not needed when deciding whether to listen to the episode. |

Table 6: Guidelines for human evaluation of podcast summaries provided by TREC.

| | 1-gram | 2-gram | 3-gram |
|---|---|---|---|
| cued_speechUniv1 | 87.86 | 56.33 | 33.37 |
| cued_speechUniv2 | 87.82 | 56.11 | 32.72 |
| cued_speechUniv3 | 84.96 | 52.05 | 31.24 |
| cued_speechUniv4 | 85.23 | 51.44 | 29.88 |
| UCF_NLP1 | 83.96 | 49.89 | 28.61 |
| UCF_NLP2 | 84.46 | 50.33 | 29.33 |
| hk_uu_podcast1 | 86.94 | 56.12 | 35.55 |
| **GrndAbs-to** | 85.83 | 57.31 | **39.03** |
| **GrndAbs-tn** | 86.38 | 58.44 | **40.38** |
| **GrndAbs-so** | 86.52 | 59.76 | **42.13** |
| **GrndAbs-sn** | 86.80 | 60.55 | **43.18** |

Table 7: Percentage of summary 1/2/3-grams appearing in the transcripts (for all baselines) or grounding chunks (for our approach). We observe that grounded abstractive summaries tend to reuse bigrams and trigrams of their grounding chunks.

- Substantial lexical mismatch exists between the spoken and written form of descriptions. Speech recognition errors are abundant. E.g., "*by Hans Christian Andersen*" has been misrecognized into "*buy homes Christian Andersen*."
- The creator descriptions are sometimes highly abstractive, do not always summarize the episode and contain teasers. E.g., "*A male perspective podcast to start a conversation...*" and "*Ever wondered how Ed Sheeran became famous*."
- The transcripts contain advertising inserts, e.g., "*I need to tell you about our sponsor...*" and the same description is used for different episodes that causes confusion to the model, e.g., "*The goal of Daily Fortnite is to build a community...*"

## 6 Conclusion

In this paper, we investigate podcast summarization to produce textual summaries for podcast episodes that help listeners to understand why they might want to play those podcasts. We present a new kind of podcast summary where spans of summary text are tethered to the original audio to allow users to interpret system-generated abstracts in context. Experiments on a benchmark dataset demonstrates the utility of our proposed approach.

## Acknowledgments

The authors would like to thank all anonymous reviewers for their insightful comments which helped improve this paper. This research was supported in part by the National Science Foundation (NSF) Grant #2143792.

## References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Sangwoo Cho, Franck Dernoncourt, Tim Ganter, Trung Bui, Nedim Lipka, Walter Chang, Hailin Jin, Jonathan Brandt, Hassan Foroosh, and Fei Liu. 2021. StreamHover: Livestream transcript summarization and annotation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6457–6474, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Christian. 2021. Why are podcasts so popular in 2021? *https://brandastic.com/blog/why-are-podcasts-so-popular/*.

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*, pages 5903–5917, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ori Ernst, Ori Shapira, Ramakanth Pasunuru, Michael Lepioshkin, Jacob Goldberger, Mohit Bansal, and Ido Dagan. 2020. Superpal: Supervised proposition alignment for multi-document summarization and derivative sub-tasks.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254, Online. Association for Computational Linguistics.

Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J.F. Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. 2020. TREC 2020 podcasts track overview. In *Proceedings of the 29th Text REtrieval Conference (TREC)*.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Gabriel Murray. 2019. Discourse analysis and its applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–17, Florence, Italy. Association for Computational Linguistics.

Hannes Karlbom and Ann Clifton. 2020. Abstract podcast summarization using BART with longformer attention. In *Proceedings of the 29th Text REtrieval Conference (TREC)*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, Dillon Burns, Alec Kerrigan, and Fei Liu. 2020. How domain terminology affects meeting summarization performance. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5689–5695, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, and Fei Liu. 2021. A sliding-window approach to automatic creation of meeting minutes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 68–75, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Lidan Wang, Walter Chang, and Fei Liu. 2020. Understanding points of correspondence between sentences for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 191–198, Online. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

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*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5460–5466, Florence, Italy. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Potsawee Manakul and Mark Gales. 2020. CUED_SPEECH at TREC 2020 podcast summarisation track. In *Proceedings of the 29th Text REtrieval Conference (TREC)*.

Matthew Marge, Satanjeev Banerjee, and Alexander Rudnicky. 2010. Using the Amazon Mechanical Turk to transcribe and annotate meeting speech for extractive summarization. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 99–107, Los Angeles. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Joseph Polifroni, Stephanie Seneff, and Victor W. Zue. 1991. Collection of spontaneous speech for the ATIS domain and comparative analyses of data collected at MIT and TI. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Sravana Reddy, Yongze Yu, Aasish Pappu, Aswin Sivaraman, Rezvaneh Rezapour, and Rosie Jones. 2021. Detecting extraneous content in podcasts. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1166–1173, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Dalit Shalom. 2019. From audio waves to words: Episodes of "the daily" now come with transcripts. *https://open.nytimes.com/from-audio-waves-to-words-episodes-of-the-daily-now-come-with-transcripts-298ab8cb9481*.

Kaiqiang Song, Chen Li, Xiaoyang Wang, Dong Yu, and Fei Liu. 2020. Automatic summarization of open-domain podcast episodes. In *Proceedings of the 29th Text REtrieval Conference (TREC)*.

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

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization.

# A Appendix

## What made the task of podcast summarization challenging?

| | |
|---|---|
| | **Lexical Mismatch between Written and Spoken Text** |
| **[S]** | ASMR reading of The Snow Man by Hans **Christian Andersen**, 1861. |
| **[T]** | Hello, my darling. *I need to tell you about our sponsor anchor dot f m. Anchor is a podcast creation and distribution tool. And it gives you everything you need to record edit. Plus they'll distribute your podcast to all of the major channels including Spotify Apple podcasts and Google podcasts free of charge you can make money with no minimum listenership and it couldn't be easier. Download the anchor app or go to Anchor dot f m– to get started sweet dreams.* Hello, my darling and Welcome to our story time. For the 12 Days of Christmas. **Our next story is** the Snowman buy homes **Christian Andersen** and we have our warm and toasty fireplace to keep us cozy while I read to you if you like what you hear [...] |
| | **Highly Abstractive Reference Summary** |
| **[S]** | A male perspective podcast **to start a conversation for men out there to begin the healing process** of what they bottle up inside. |
| **[T]** | [...] And you know what? I'm tired and I've sat down with a lot of guys in the past year a lot of women in the past year. I've shared my ideas with them and I really just want to inspire people **to start a conversation to help them begin the healing process** of you know, what I don't want to hold up things on the inside anymore. So I've been thinking about this word feelings feelings feelings feelings [...] |
| | **Same Summary for Different Podcast Episodes** |
| **[S]** | The goal of Daily Fortnite is to build a positive community of Fortnite players so we can all enhance our enjoyment of Fortnite together. |
| **[T]** | Welcome back to another episode of daily Fortnight your daily podcast about Fortnight. I'm your host Mikey AKA Mike. Daddy AKA magnificant Mikey. So today we have the fishing frenzy results are in you can go check that out on the leaderboard [...] |
| | **Part of the Summary is Irrelevant to the Transcript** |
| **[S]** | If you're like me **you sometimes suffer from "imposter syndrome".** I hope these short positive messages will help my tile contractor friends to know their worth, overcome "imposter syndrome" and continue to grow their contracting businesses! |
| **[T]** | [...] I will be doing a brief, you know podcast episode every week on mindset and I'm thinking of calling it mindset Monday [...] *The other thing I want to talk to you today about is a new sponsor of tile money. So I want to thank that my new sponsor* [...] So this new mindset segment that I want to record for the for the podcast episodes. You know, it got me thinking recently Chris Ford posted up. A question to the group about this thing called **impostor syndrome and it's something so many of us I struggle with it myself personally.** |
| | **Potential Teaser Texts in the Summary** |
| **[S]** | Ever wondered how Ed Sheeran became famous or how Stormzy writes his songs? [...] **Straight Up**, a game-changing **new podcast pulling back the curtain on UK music at its most exciting moment yet**, lifts the lid on all this and more. |
| **[T]** | **This is straight-up the 490 UK music podcast** hosted by journalists me cackling Johnston. I met Eleanor Halls will be taking you through the biggest music headlines the hottest entry closet and spotlighting the artists that we're into right now [...] **our guests will pull back the curtain on the musicians that everyone's talking about to top it all off.** We chat all of our guests over their favorite drink. So why not grab a glass and join us for the stories? [...] |

Table 8: What made the task of podcast summarization challenging? a) Lexical mismatch between spoken and written forms and speech recognition errors ("*by Hans Christian Andersen*" was mistranscribed into "*buy homes Christian Andersen.*") b) Highly abstractive creator description, e.g., "*A male perspective podcast to start a conversation...*" c) The same summary is used for different podcast episodes, e.g., "*The goal of Daily Fortnite is to build a positive community...*" d) The creator description does not summarize or describe the episode, e.g., "*I hope these short positive messages will help my tile contractor friends...*" and "*Ever wondered how Ed Sheeran became famous*". e) The podcast is improvised, its content lacks discourse structure, the transcript contains frequently recognition errors and advertising inserts, e.g., "*I need to tell you about our sponsor...*"