

PODTILE: Facilitating Podcast Episode Browsing with Auto-generated Chapters

Azin Ghazimatin*[†]
Berlin, Germany, Spotify

Ekaterina Garmash*
London, UK, Spotify

Gustavo Penha
Amsterdam, Netherlands, Spotify

Kristen Sheets
San Francisco, US, Spotify

Martin Achenbach
Berlin, Germany, Spotify

Oguz Semerci
Boston, US, Spotify

Remi Galvez
New York, US, Spotify

Marcus Tannenberg
Gothenburg, Sweden, Spotify

Sahitya Mantravadi
New York, US, Spotify

Divya Narayanan
New York, US, Spotify

Ofeliya Kalaydzhyan
Boston, US, Spotify

Douglas Cole
Boston, US, Spotify

Ben Carterette
New York, US, Spotify

Ann Clifton
New York, US, Spotify

Paul N. Bennett
Boston, US, Spotify

Claudia Hauff
Delft, Netherlands, Spotify

Mounia Lalmas
London, UK, Spotify

Abstract

Listeners of long-form talk-audio content, such as podcast episodes, often find it challenging to understand the overall structure and locate relevant sections. A practical solution is to divide episodes into chapters—semantically coherent segments labeled with titles and timestamps. Since most episodes on our platform at Spotify currently lack creator-provided chapters, automating the creation of chapters is essential. Scaling the chapterization of podcast episodes presents unique challenges. First, episodes tend to be less structured than written texts, featuring spontaneous discussions with nuanced transitions. Second, the transcripts are usually lengthy, averaging about 16,000 tokens, which necessitates efficient processing that can preserve context. To address these challenges, we introduce PODTILE, a fine-tuned encoder-decoder transformer to segment conversational data. The model simultaneously generates chapter transitions and titles for the input transcript. To preserve context, each input text is augmented with global context, including the episode’s title, description, and previous chapter titles. In our intrinsic evaluation, PODTILE achieved a 11% improvement in ROUGE score over the strongest previous baseline. Additionally, we provide insights into the practical benefits of auto-generated chapters for listeners navigating episode content. Our findings indicate that auto-generated chapters serve as a useful tool for engaging with less popular podcasts. Finally, we present empirical evidence that

using chapter titles can enhance the effectiveness of sparse retrieval in search tasks.

CCS Concepts

• **Computing methodologies** → **Natural language generation.**

Keywords

Chapterization, Processing Long Documents, Generative Models

ACM Reference Format:

Azin Ghazimatin[1][2], Ekaterina Garmash[1], Gustavo Penha, Kristen Sheets, Martin Achenbach, Oguz Semerci, Remi Galvez, Marcus Tannenberg, Sahitya Mantravadi, Divya Narayanan, Ofeliya Kalaydzhyan, Douglas Cole, Ben Carterette, Ann Clifton, Paul N. Bennett, Claudia Hauff, and Mounia Lalmas. 2024. PODTILE: Facilitating Podcast Episode Browsing with Auto-generated Chapters. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24)*, October 21–25, 2024, Boise, ID, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/00.0000/0000000.0000000>

1 Introduction

We define *chapterization* as the task of dividing a document into semantically coherent, non-overlapping segments and assigning each segment an appropriate title that reflects its content. This process, also referred to as structured summarization [24] or smart chaptering [47], has been shown to provide users with a convenient and structured content overview and simplify navigation across a document [8, 20]. The value of chapterization has been acknowledged for its role in facilitating other tasks such as information retrieval [51] and the summarization of lengthy documents [8, 27]. With the increasing volume and availability of spoken user-generated content, like podcasts and videos, the need for chapterization has grown,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '24, October 21–25, 2024, Boise, ID, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0436-9/24/10
<https://doi.org/00.0000/0000000.0000000>

*Equal contribution.

[†]Corresponding author. Email: azing@spotify.com

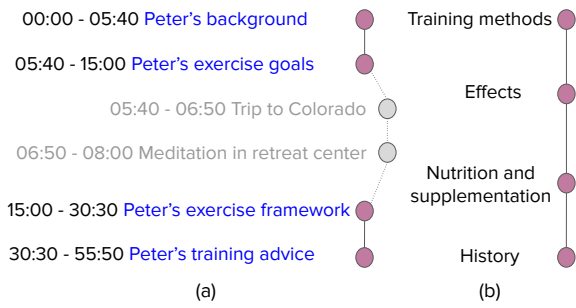


Figure 1: Chapters (purple circles) for (a) an episode about training tips vs. (b) a structured Wikipedia article about training. The episode chapters have short tangential discussions (gray circles), shared context (Peter’s experience), and a consistent title style. In contrast, Wikipedia chapters focus on the main topic with short titles that lack global context.

offering significant benefits in content compression and navigation [8, 27].

Podcast and video chapterization can ideally be provided by content creators themselves since there is no standardized format or protocol for chapter annotations. This, however, is frequently not the case; on our platform that hosts audio podcasts, the vast majority of episodes do not have creator-provided chapters. We bridge this gap by automating chapterization using a large language model-powered system trained on available creator chapters.

Most of previous research has concentrated on chapterizing structured written texts, such as Wikipedia articles, news, and journals [36, 39, 52, 63]. There are however a few studies that focus on spoken discourse [24, 35, 47, 61, 68]. Yet, chapterizing spoken language documents, particularly podcast episodes, presents unique challenges compared to segmenting short, structured texts. Spoken discourse is usually more fluid, topically diverse, and less structured, and often features frequent digressions due to its interactive, real-time, and informal nature [17, 28, 47].

Another challenge is the considerable length of podcast episodes, whether measured by time or word count when transcribed. This not only increases computational costs but also poses a modeling challenge; many podcasts contain long-range semantic dependencies that need to be captured by chapterization. For instance, Figure 1(a) shows a podcast episode where the discussion diverges into a tangent about traveling before returning to the main topic of exercising. Such tangents are typical of informal conversational podcasts. To predict a chapterization like the one in Figure 1(a), a model must track the overarching context and theme. “Knowing” that the main topic is physical exercise helps the model distinguish segments about different aspects of this topic. Additionally, tracking predicted chapters throughout the episode helps the model generate consistent titles (in this example, focused on the guest named “Peter”). Chapterizing a Wikipedia article as illustrated in Figure 1(b), however, does not face these challenges since it is shorter and more structured.

Lately, there has been a growing focus on chapterizing conversational datasets. In [24], segmentation and title assignment are modeled jointly, enhancing the predictive capabilities of both tasks.

This model leverages LongT5 [21] as the pre-trained sequence-to-sequence large language model (LLM). However, the context size of LongT5 is 16k which is not sufficient for processing podcast transcripts with 16k tokens on average. In Retkowski and Waibel [47], a two-stage chapterization model is used to first segment and then generate titles for the identified segments. This model uses longer context by incorporating previous chapter titles as left context summaries to generate chapter titles. The model’s two-stage design, however, inhibits information sharing between the two tasks.

We can address the challenge of long inputs and long-distance dependencies in podcasts in several ways. First, a sufficiently large and powerful backbone LLM *may* provide a large enough context window to process an entire episode’s transcript and produce accurate chapters. However, using a large LLM incurs significant computational and financial costs and may not fully capture all long-distance dependencies. To efficiently address these challenges, we propose PODTILE, a chapterization model that builds on the strengths of existing models, particularly [24], and extends them by dedicating a small portion of input text to explicit global context encoded as text: specifically, podcast episode metadata that reflects the overall content of the episode and previously generated chapter titles. This allows a reasonably-sized¹ LLM to handle long and unstructured content effectively, without solely relying on the LLM’s power. Following [24], we use LongT5 encoder-decoder model, which offers a compromise between efficiency and model power.

We validate our proposed approach using two public non-podcast datasets and one internal podcast dataset. Our findings indicate that using global context as part of the input text enhances the quality of chapter titles, particularly for longer documents in conversational datasets. We recently deployed PODTILE on our platform. Usage statistics indicate that podcast listeners find the auto-generated chapters helpful for browsing through episodes, particularly in lesser-known podcasts. Finally, we assess the utility of our generated chapter titles in a retrieval downstream task using the TREC Podcast Track dataset [26]. Adding these titles to episode descriptions significantly enhances sparse retrieval effectiveness compared to an extractive summarization baseline.

We summarize our contributions as follows:

- introduction of a new model, PODTILE, which effectively extends [24] to address the challenges of podcast chapterization;
- extensive intrinsic and extrinsic evaluations demonstrating the effectiveness and utility of the proposed approach;
- deployment of the model in a user-facing production system and preliminary analysis of usage patterns for podcast chapters.

2 Related Work

We review related work, which has guided us in the various decisions we made to develop and deploy PODTILE.

Text segmentation. Early approaches for text segmentation (aka boundary detection) were unsupervised due to lack of sufficient supervised data. These approaches involve computing a cohesion score or mutual information [55] between consecutive blocks of sentences. This can be achieved using TF-IDF (or its variations) [10, 23],

¹With less than a billion parameters.

LDA topics [48], probabilistic language models, word2vec embeddings [2], or transformer-based embeddings [13, 41]. The coherence scores are then plotted against the sentences, with the valleys considered as boundaries. When similarities are modeled as edge weights in the semantic relatedness graph of the document, maximal cliques are treated as semantically coherent segments [18].

The availability of large labeled data led to the increased use of supervised methods for addressing unique segmentation nuances across different domains. These approaches generally involve training a boundary classifier on a sequence of input sentences [9, 30, 52, 53, 58, 65] or on pairs of left and right context blocks [39, 54]. To represent the input, these methods utilize various techniques, including statistical features [14, 53], neural networks [3, 12, 30, 32, 50, 56, 58], or transformers [4, 33, 35, 38, 39, 52, 65].

Previous work on text segmentation primarily focuses on detecting segment boundaries without addressing title assignment which is necessary for podcast chapterization. Next, we review previous studies that address both segmentation and title assignment.

Joint segmentation and title assignment. Prior studies suggest that jointly modeling the segmentation task and title assignment/generation offers mutual benefits for both tasks [24]. In scenarios where the set of topics is considered closed, it is common practice to feed the learned representations into a multi-class classifier for title assignment [3, 19, 31, 37, 38, 53]. However, if the set of titles is open, a generative approach is employed [24, 35, 36, 59, 66]. Given the diversity of podcast episode titles, we also adopt a generative approach similar to [24].

There is also a substantial body of literature on multi-modal segmentation [16, 60, 61, 65]. However, since our model is uni-modal, we do not cover this topic in this paper.

Capturing long-range dependency. Chapters in a document can be seen as structured summaries of content [24], similar to a summarization task. Therefore, chapterization is expected to benefit from capturing long-range context. Most recent studies aimed at making transformers more efficient for processing long texts focus on sparsifying or approximating the attention mechanism [5, 6, 11, 22, 29, 49, 57]. Another method for capturing long context is to hierarchically or incrementally merge the output of input chunks to facilitate information flow between them [7]. However, this approach is slow and computationally expensive. Ge et al. [15] recently introduced in-context auto-encoders to compress long context into a few tokens, which can be passed as additional input to an LLM with a limited context window. Learning these tokens, however, requires extensive pre-training. In our work, we use LongT5, which employs attention sparsification using global transient attention to efficiently capture longer context. Additionally, we augment the input chunks with document metadata to preserve context beyond the typical context size limit of most transformers.

3 Method

Our work builds on the method by Inan et al. [24], modeling chapterization as simultaneous segmentation and title generation in a sequence-to-sequence fashion. The input is the text to be chapterized, and the output is a textual specification of chapter boundaries and titles. This approach uses a pre-trained LLM fine-tuned on

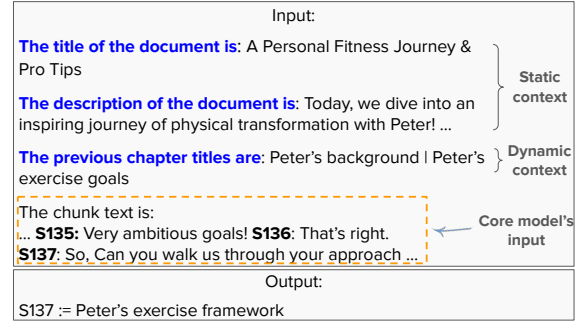


Figure 2: The input and output formatting of the chapterization model. The dotted box is the input to the core model.

supervised data, leveraging its vast linguistic and real-world knowledge. As a text-based model, it effectively integrates segmentation and title prediction, while also incorporating diverse contextual information to enhance the accuracy of the prediction, which is the main contribution of this paper.

We refer to the model by Inan et al. [24] as our core model, which we detail and explain its application to the podcast domain in Section 3.1. Our contribution involves incorporating additional contextual cues into this core model to improve generalization on long-input data and mitigate the limitations of the local nature of chapterization inference. Specifically, we explore:

- (1) **Static context** (Section 3.2): Metadata outlining the *overall* content of the document. This is useful when the model cannot access the entire document at once. Specific implementations depend on the domain and dataset, detailed in Section 4.
- (2) **Dynamic context** (Section 3.3): The intermediate state of the left-to-right chapterization process. This information provides access to earlier chapterization decisions, guiding the selection of subsequent chapters.

3.1 Core model

Our core model is based on the segmentation and labeling framework **Gen (seg+label)** from Inan et al. [24], which has been demonstrated experimentally to be the best-performing variant of their model. This model employs an encoder-decoder architecture with an underlying Transformer LLM. While any existing LLM adhering to the seq2seq API can be used, our experiments specifically use the LongT5 pre-trained LLM [21], in line with Inan et al. [24].

The input-output formatting for our core model is illustrated in Figure 2. We augment the raw input text by adding index numbers before each sentence. This allows the decoder to predict the start of a chapter by referencing one of these indices. The output sequence is a chronologically ordered concatenation of strings formatted as:

$\{\text{first_sentence_index}\} := \{\text{title}\}$

with character “|” as the separator. Given that input texts can be arbitrarily long—particularly in media such as podcasts (see Table 1)—and open-source LLMs have limited input capacities,² the initial step in our approach involves chunking the input text into segments that can be processed by an LLM. Each training datapoint consists of a chunk of input text and the corresponding output string, which

²LongT5 context size is 16,384, see https://huggingface.co/docs/transformers/en/model_doc/longt5.

includes chapter boundaries and titles relevant to that chunk. If a chunk does not contain any chapter boundaries, the output string is "No chapter boundaries were found." The chunking process uses a sliding non-overlapping window with a size smaller than the LLM's input capacity.

This necessity for chunking the input can result in predictions that are locally informed and are not based on the broader context about the entire input text or about the chapterization predictions made in the preceding chunks. Given the considerable average length of podcasts and the frequent presence of long-distance dependencies, such locality may result in suboptimal chapter quality. To address this limitation, we propose incorporating global context using methods described below.

3.2 Adding static context

When processing chunked input (explained above), the model lacks access to content before or after a given chunk. This can result in predicted chapters that are either not specific enough to distinguish content outside the chunk or too focused on details specific to the chunk but irrelevant to the overall discussion.

To address this, we propose including metadata that outlines the document's overall content, providing a general context. We call this *static context*, as it is provided prior to chapterization and remains unchanged. The specific content and structure of this metadata varies by dataset, detailed in Section 4. Figure 2 shows an example with the title and description of an episode as static context.

3.3 Adding dynamic context

Another disadvantage of local chunked processing is the model's lack of awareness of prior chapterization decisions for a given input document. As a result, each local prediction step may produce boundaries and titles that are inconsistent with previous decisions. This can lead to issues such as repetitive titles, different levels of chapter granularity, and varying linguistic styles in titles.

To provide dynamic information about the state of the chapterization process, we add the sequence of titles already predicted for the earlier portions of the document to the input text. Figure 2 shows how previously predicted titles are added to the input text.

4 Experimental setup

4.1 Datasets

We downsampled our **podcast dataset** from a proprietary internal catalog, using only English episodes that were chapterized by their creators. Our final dataset contains 10.8k episodes, uniformly sampled with several filters. Chapters in these episodes range from 30 seconds to 30 minutes, and titles are shorter than 15 words. We randomly split the resulting dataset into train/validation/test partitions of 8k/1k/1k episodes. For each episode, we use both title and description as the static context since 96% of episodes in our catalog have descriptions, with 57% of those longer than 20 words. The majority (91%) of episodes in our dataset are conversational, featuring multi-speaker discussions.

To gauge PODTILE's effectiveness across different domains, we use two other publicly available English datasets. **WikiSection** [3] is a Wikipedia-based dataset limited to two categories, *en_disease* and *en_city*, with normalized section titles for discriminative title

prediction. Examples of segment titles are in Table 1. We use only the English documents and use the title and abstract of each document as the static context. The second dataset, **QMSum** [68], is a collection of meeting transcripts annotated with topic segments and labels. While it is closer to podcasts as it involves conversational data, it is a low-resource dataset with only 232 data points. We use the user-generated meeting summaries as the static context.

Table 1 presents descriptive statistics for the three datasets. Compared to Wikisection, the podcast and QMSum datasets feature longer documents and chapters, with more descriptive chapter titles. The podcast dataset shows significant variability in the number of chapters per episode and title length, indicating greater diversity.

4.2 Baselines

We compare PODTILE against the following baselines:

CATS [52]: A multi-task learning model that combines boundary classification with coherent sequence detection that differentiate correct sequences of sentences from the corrupt ones. This model is chosen due to the recent state-of-the-art performance of hierarchical encoders for segmenting video transcripts [47].

Gen (seg + label) [24]: A single-stage seq2seq model that uses LongT5 to jointly generate chapter titles and boundaries (structured summarization), similar to PODTILE's core model.

GPT-4 [1]: Zero-shot learning with GPT-4, using an extended context of 128k tokens (gpt-4-0125-preview). We instruct the model to chapterize the entire transcript and return the chapter titles and starting sentence IDs in JSON format for easy parsing. The experiment was conducted in the second week of May 2024.

4.3 Implementation details

We use LongT5 [21] (base size, ~220M parameters) with transient global attention, as our backbone model. The training setup includes a batch size of 1, a learning rate of $5.0e-5$ with scheduler type of linear, and a maximum of 4 epochs. The same setting was used for other datasets with the exception of learning rate $1.0e-4$ for Wikisection. We use input chunks of up to 8000 words,³ with 7000 words dedicated to the document text and up to 1000 words to the metadata. In Gen(seg+label), all the 8000 words are used for document text. On average, each transcript in the podcast dataset is broken into 1.75 chunks. Training and inference for offline evaluations were conducted on a Ray [40] cluster with a single node and a single GPU. Training on the podcast dataset took approximately 3 days. Inference of 1.1k episodes lasts an average of 1 hour.

4.4 Evaluation metrics

It is common to evaluate chapter boundaries and generated titles separately using their respective metrics [24]. For segmentation evaluation, we use **WindowDiff** [45], which measures the average difference between the number of boundaries in predicted and reference values over spans of k sentences. This metric is parametrized by k , the sliding window size, usually set to half the average segment length (in sentences). We estimate k for each dataset using the training partition and report it in Table 2. Lower metric values indicate more accurate segmentation.

³A conservative choice to ensure there are no more tokens than 16k.

Table 1: Statistics for datasets used in the experiments. The terms “doc” and “desc” denote document and description, respectively.

Dataset (size)	Avg. no. chapters	Avg. segment length (sentences)	Avg. title length (words)	Avg. no. tokens (doc title desc)	Avg. no. words (doc title desc)	Example title
Podcast (10,804)	11.3 ± 7.5	80.9 ± 83.7	6.2 ± 5.7	16,098 20 130	11,845 11 102	How your smile affects others
Wikisection (23,129)	5.2 ± 3.8	11.0 ± 13.7	1.0 ± 0.0	1,603 6 134	1,068 2 85	city.geography, disease.genetics
QMSum (232)	4.7 ± 1.9	163.3 ± 156.9	6.4 ± 3.6	13,391 0 130	8,467 0 104	Design and availability of actual components

For titles, reference-based summarization metrics like ROUGE [34] and BERTScore [67] are commonly used. Previous work often computes these metrics on summaries created by concatenating chapter titles sequentially, which hinders individual title assessment. To evaluate titles individually, we employ a heuristic alignment method between reference and predicted chapters. For each chapter c_i in one set (reference or prediction), we find the chapter c_j in the other set with the highest overlap at sentence level, then match their titles. Note that this matching process is asymmetric, meaning a title matched from the reference to the prediction set does not guarantee a reverse match. We use SBERT title representations [46]⁴ to apply soft-matching distance and define the metrics as:

$$\text{ROUGEL}_{F1,aligned} = \frac{\sum_{(t,t') \in \text{Matches}_{all}} \text{ROUGEL}_{F1}(t,t')}{|\text{Matches}_{all}|} \quad (1)$$

$$\text{SBERT}_{prec} = \frac{\sum_{(t,t') \in \text{Matches}_{pred}} \text{SBERT}(t,t')}{|\text{Matches}_{pred}|} \quad (2)$$

$$\text{SBERT}_{recall} = \frac{\sum_{(t,t') \in \text{Matches}_{ref}} \text{SBERT}(t,t')}{|\text{Matches}_{ref}|} \quad (3)$$

where Matches_{pred} is the set of title pairs (t, t') where a predicted title t' is matched with reference title t with highest overlap. Similarly, Matches_{ref} is a set of title pairs (t, t') where reference title t is matched with predicted title t' with highest overlap. Matches_{all} denotes the union of Matches_{pred} and Matches_{ref} . For simplicity, we refer to $\text{ROUGEL}_{F1,aligned}$ as ROUGEL_{F1} hereinafter. SBERT_{F1} is computed as the geometric mean of (2) and (3).

4.5 Ethics Statement

We display auto-generated chapters for episodes that do not have creator-provided chapters. Users are informed that these chapters are generated by AI with the following disclaimer: *The chapters are auto-generated.* Additionally, we ensure compliance with the terms and conditions of Spotify for Podcasters and allow creators to overwrite AI-generated chapters or opt-out of this feature at their discretion. To protect users from potentially harmful AI-generated content, we employ a safety mechanism to remove sensitive or inappropriate titles before they are displayed. We also allow for the immediate manual removal of any reported harmful content.

5 Offline Results

We present the findings from our experiments, addressing four research questions. The results are detailed in Tables 2, 3 and 4.

⁴We use SBERT instead of BERT for title representation because we measure distances between entire chapter lists, where the atomic elements are titles. Unlike BERTScore, which measures similarity between sentences at the word level for tasks like machine translation and image captioning, SBERT is better suited for our purpose.

Q1: How does PODTILE perform on conversational datasets? Table 2 shows that PODTILE (row 4), with both static and dynamic context enabled, significantly outperforms the strongest baseline, Gen (seg+label) (row 3), on the podcast dataset according to title metrics (paired t-test, p-value < 0.05). A similar trend is observed in the QMSum dataset (rows 11-13), though not statistically significant which might be due to its small test set (35 documents). This highlights the importance of capturing global context for high-quality title generation. Segmentation accuracy, measured by WinDiff, remains close to the baseline, indicating that segmentation relies less on global context. Comparison with CATS (row 1) suggests that coherence modeling in this method is less effective on conversational datasets compared to structured texts. The lower performance of GPT-4 zero-shot inference (row 2) highlights the challenge of chapterizing long conversational documents without fine-tuning, even for powerful models like GPT-4. On Wikisection (rows 8-10), where documents are short and well-structured, our model performs comparably to Gen(seg+label), as expected.

Q2: Do static and dynamic context contribute equally to improving title metrics? The results in Q1 suggest that our new contextual features improve the title quality of conversational data more than boundary accuracy. To examine the individual effects of static and dynamic context on titles, we conduct an ablation study (rows 5-7). Disabling static context (row 6) causes a more significant decrease in title metrics than disabling dynamic context (row 5).⁵ After examining a few examples, we speculate that lower performance in the dynamic context-only model may be due to a chapterization style different from the ground truth,⁶ hinting at the insufficiency of the state-of-the-art reference-based metrics and a single ground truth for chapterization.

For a deeper understanding of the context’s effect on titles, particularly *title consistency* across chapters within an episode, we examine title length variation. We compute the coefficient of variation⁷ for each episode and average it across the test set. Higher average coefficients indicate lower consistency. The baseline (row 3) shows the highest variation (0.6), while PODTILE and the dynamic context-only model score the lowest (0.55). The static context-only model’s score (0.58) is close to the baseline. These results highlight the limitations of reference-based metrics used in Table 2 and show that dynamic context positively contributes to title quality, aligning with the original motivation for this feature (conditioning the next title on the already predicted ones).

⁵Although enabling only the dynamic context (row 6) improves boundary metrics compared to when both features are disabled (row 7), but slightly reduces title quality.

⁶For example, in an episode about scary stories, the model using dynamic context predicted generic titles like “Story 1”, “Story 2”, and so on, whereas the ground truth had specific titles like (“Invisible Humanoid”, “Family of Sasquatch”, ...).

⁷Coefficient of variation = (title length std) / (mean title length).

Table 2: Comparison of PODTILE against the baselines according to boundary and title metrics across three datasets: Podcast, Wikisection, and QMSum. The best metric values for each dataset are marked in bold. \ddagger denotes statistical significance of PODTILE over the strongest baseline. \downarrow indicates that lower values are better. k denotes the parameter value of *WinDiff*. Notation 7000+1000 (chunk size) means that 7000 words are used for input document text and 1000 for static and dynamic context.

	Model	Chunk size (words)	Static/Dynamic context	WinDiff ↓	ROUGEL _{F1} ↑	SBERT _{F1} ↑
Podcast dataset (<i>k</i> = 45)						
(1)	CATS [52]	-	-	0.505	-	-
(2)	GPT-4 [1]	-	✓/✗	0.448	0.134	0.315
(3)	Gen (seg+label) [24]	8000	✗/✗	0.364	0.208	0.394
(4)	PODTILE	7000+1000	✓/✓	0.365	0.231 [‡]	0.414 [‡]
(5)	PODTILE (Ablation)	7000+1000	✓/✗	0.368	0.235[‡]	0.418[‡]
(6)		7000+1000	✗/✓	0.368	0.209	0.392
(7)		7000	✗/✗	0.371	0.215 [‡]	0.400 [‡]
Wikisection dataset (<i>k</i> = 6)						
(8)	CATS [52]	-	-	0.113	-	-
(9)	Gen (seg+label) [24]	8000	✗/✗	0.188	0.873	0.925
(10)	PODTILE	7000+1000	✓/✓	0.134	0.866	0.924
QMSum dataset (<i>k</i> = 85)						
(11)	CATS [52]	-	-	0.469	-	-
(12)	Gen (seg+label) [24]	8000	✗/✗	0.443	0.196	0.326
(13)	PODTILE	7000+1000	✓/✓	0.443	0.234	0.365

Q3: Do longer documents benefit more from global context? The primary rationale behind integrating global (static and dynamic) context in PODTILE’s input was to improve the chapterization of long documents that exceed the model’s context size. Thus, we hypothesized that longer documents would benefit more from PODTILE compared to the baselines. This hypothesis is validated by the findings in Table 3. The first row shows the percentage improvement in title quality over the baseline, Gen(seg+label), for documents fully processed by PODTILE. The second row shows improvements for longer documents requiring chunking, which make up 80% of the test data. It is evident that longer documents see more substantial improvements. Table 4 demonstrates how using metadata for chapterizing long documents enhances chapter titles’ informativeness. PODTILE adds words like “Planet” and “Sandeep” from the metadata, which are missing in the input chunk with chapter boundaries due to an already established context.

Q4: How does the length and source of the static context impact chapterization? To test if longer static context enhances auto-generated chapter quality, we computed the Spearman rank correlation between static context length and the Δ ROUGEL_{F1} of PODTILE with and without static context. We found a negligible negative correlation, suggesting that longer static context does not necessarily improve metric scores.

Given the increasing use of LLMs for content generation, we further explored the robustness of PODTILE to AI-generated static context. For this, we instructed GPT-4 to generate episode descriptions based on the episode transcripts and used them in place of creator-provided descriptions. As a result, we observed a 7% drop in ROUGEL_{F1} compared to PODTILE that uses creator descriptions (row 4 in Table 2). We conclude that creator-provided static context is more effective for chapterization.

6 Deployment

Podcast chapters with creator-provided timestamps have been available on our platform. There overall coverage, however, is low. In April 2024, we started a limited roll-out of our chapterization model.

Table 3: PODTILE’s title metrics improvement (%) over the baseline Gen (seg+label) for short (no chunking needed) and long (chunking needed) transcripts.

Needs chunking?	Δ ROUGEL _{F1} %	Δ SBERT _{F1} %	% in Test data
NO	6.61	2.92	20
YES	12.08	5.51	80

Table 4: Anecdotal examples indicating how metadata can improve informativeness of chapter titles in PODTILE in comparison with the baseline. Gen (seg+label) is unable to infer the bold words from its input text.

PODTILE	Ground-truth	Gen (seg+label)
Sigma MC-11 Adapter	Gear of the Week: Sigma’s MC-11	Sigma Mount Converter
Planet of Lana Reviews	Planet of Lana Reviews	Lana Review
Sandeep ’s Family Background	About Sandeep ’s Background	Birth

Since auto-generated chapters broadens availability of chapters, we expect that if they have high quality, we would see higher engagement with chapters. Overall we saw an 88.12% increase in chapter-initiated plays after the roll-out.

To understand user engagement with podcast chapters, we measured the percentage of listeners who interacted with chapters by playing or scrolling them. We compared engagement ratios between episodes with auto-generated chapters and those with creator-provided chapters. A lower ratio would indicate that auto-generated chapters are less attractive or useful. Our model, designed to mimic creator chapters, assumes this ratio should not exceed 1. We collected data over the last 10 days and plotted the 7-day moving average in Figure 3. The overall trend shows stable, positive growth. Notably, less popular shows had a higher engagement ratio (almost 0.75) compared to more popular shows (about 0.53). This

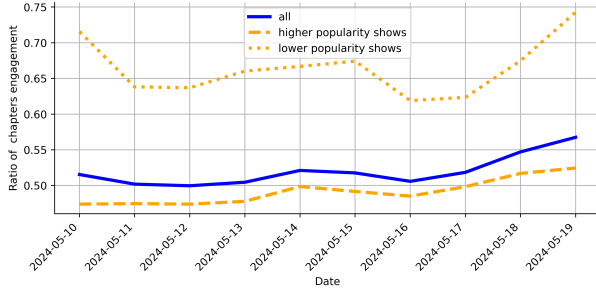


Figure 3: Ratio of relative chapters engagement between episodes with auto-generated titles and creator-provided titles, plotted as the moving average over previous 7 days.

suggests that auto-generated chapters are particularly beneficial for less popular shows, enhancing user engagement effectively.

We divided users into five groups based on their total consumption since PODTILE’s deployment and calculated the percentage of chapter plays for each group. Figure 4 shows that creator-provided chapters are predominantly used by heavy listeners, likely due to their lower coverage. In contrast, auto-generated chapters have a more balanced distribution, with 50.84% of play counts from super light to upper medium users. This shows auto-generated chapters help users with limited time navigate episodes efficiently.

To examine the impact of episode duration on chapter usage, we computed the Spearman rank correlation between duration of episodes with auto-generated chapters and the percentage of their chapter users. The weak correlation (0.17) indicates that episode duration alone does not determine chapter usage. However, in entertainment categories like “TV and Shows”, “Leisure”, and “Arts”, longer episodes receive more chapter plays, suggesting that both duration and content influence chapter usage.

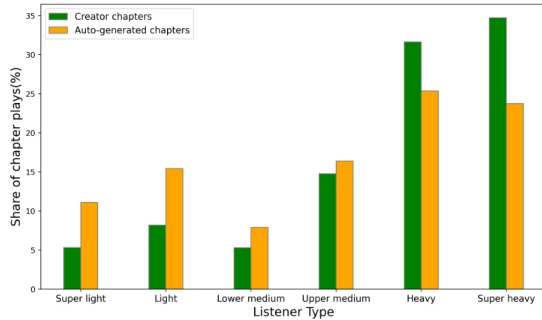


Figure 4: Percentage of creator-provided and auto-generated chapter plays across five user groups based on consumption.

7 Extrinsic Evaluation

Podcast chapterization primarily aims at facilitating navigation through episode content. This section shows how podcast chapters can also enhance episode search retrieval as a downstream task.

Textual descriptions of podcast episodes often miss key details that listeners seek. These details are usually in the transcripts, which are lengthy and costly to index. We propose using chapter titles as summaries instead of full transcripts to enhance episode

Table 5: Extrinsic results for the TREC podcast dataset. The \ddagger denotes statistical significance when compared to the Desc+princ using Students’ t-tests at 0.95 confidence interval. While Desc+trans is more effective its index size is more than 10 times bigger than Desc+chap.

Setting	nDCG	R@30	R@50	R@100	RR
Desc	0.239	0.241	0.264	0.324	0.441
Desc+princ	0.243	0.242	0.254	0.322	0.440
Desc+chap	0.276 \ddagger	0.265 \ddagger	0.315 \ddagger	0.362	0.528 \ddagger
Desc+trans	0.336	0.374	0.422	0.516	0.446

descriptions. This approach could reduce costs by at least tenfold⁸ compared to indexing entire transcripts. We believe that adding chapter titles to descriptions will significantly improve sparse retrieval in search by including important terms users search for.

To test this hypothesis, we design an experiment to explore the impact of indexing chapter titles on search effectiveness. For this, we use the TREC podcast dataset [26] collected for short segment retrieval and summarization task. This dataset contains human relevance judgments for 54 search queries, of 3 types: topical, re-finding, and known items. a pool of 100k episodes, and 900 labeled query-episode pairs. Note that in this experiment, we perform retrieval and report metrics on episode-level and not segment-level. We use BM25, implemented by Anseri [62], as the retrieval method, measure search success by nDCG, recall, and Reciprocal Rank (RR), and consider 4 methods for indexing episodes:⁹

- **Desc:** Only episode descriptions are indexed.
- **Desc+princ:** Descriptions are expanded with key sentences of the transcripts, extracted using the Principal Uniq-Ind [64].¹⁰
- **Desc+chap:** Descriptions are expanded with chapter titles and then indexed.
- **Desc+trans:** Both descriptions and full transcripts are indexed. This is expected to perform the best despite the high cost.

Table 5 summarizes the results. We observe that **Desc+chap** significantly outperforms the baselines (**Desc** and **Desc+princ**) according to nDCG, R@30, and R@50. This demonstrates that chapter titles effectively capture the essence of the transcript while significantly reducing the storage needed for indexing.

8 Conclusion

We developed a chapterization model that efficiently processes podcast episodes at scale using small LLMs. Our model captures long-range dependencies by incorporating short global context in each transcript chunk. We evaluated our model on internal and public datasets, demonstrating its competitive performance on both structured and conversational data. After deploying the model on our platform, we observed that users find auto-generated chapters helpful for browsing episode content. We also showed that chapter

⁸We compare the size of the inverted index created by indexing chapter titles with that generated from the whole transcripts.

⁹We leave the efficient application of the costly document expansion methods based on abstractive summarization [25, 44] or Doc2Query [42, 43] as future work.

¹⁰This method computes the ROUGE1_{F1} score between each sentence s_i and rest of the transcript, selecting the top sentences with the highest scores. “Uniq” means only unique n -grams are considered and “Ind” indicates independent scoring of sentences. Extractive summaries are limited to 24 words for fair comparison with chapter titles.

titles provide concise and informative summaries of transcripts, enhancing episode descriptions and improving search effectiveness.

We acknowledge that podcast chapterization is subjective, and a single ground-truth reference may not fully capture the model's capabilities. Therefore, we plan to extend our evaluation to include reference-free metrics. Additionally, we aim to leverage other modalities, such as audio and video, to further improve chapterization.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Alexander A Alemi and Paul Ginsparg. 2015. Text segmentation based on semantic word embeddings. *arXiv preprint arXiv:1503.05543* (2015).
- [3] Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A Gers, and Alexander Löser. 2019. SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. *Transactions of the Association for Computational Linguistics* 7 (2019), 169–184.
- [4] Haitao Bai, Pinghui Wang, Ruofei Zhang, and Zhou Su. 2023. SegFormer: a topic segmentation model with controllable range of attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 12545–12552.
- [5] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).
- [6] Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. 2024. Unlimiformer: Long-range transformers with unlimited length input. *Advances in Neural Information Processing Systems* 36 (2024).
- [7] Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. BoookScore: A systematic exploration of book-length summarization in the era of LLMs. *arXiv preprint arXiv:2310.00785* (2023).
- [8] Ciprian Chelba, Timothy J Hazen, and Murat Saraclar. 2008. Retrieval and browsing of spoken content. *IEEE Signal Processing Magazine* 25, 3 (2008), 39–49.
- [9] Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward Unifying Text Segmentation and Long Document Summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 106–118.
- [10] Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. 26–33.
- [11] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794* (2020).
- [12] Yifeng Ding, Yimeng Dai, Hai-Tao Zheng, and Rui Zhang. 2022. GiTS: Gist-driven Text Segmentation. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [13] Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. 334–343.
- [14] 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.
- [15] Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945* (2023).
- [16] Iacopo Ghinassi, Lin Wang, Chris Newell, and Matthew Purver. 2023. Multimodal Topic Segmentation of Podcast Shows with Pre-trained Neural Encoders. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. 602–606.
- [17] Reshmi Ghosh, Harjeet Singh Kajal, Sharanya Kamath, Dhuri Shrivastava, Samyadeep Basu, and Soundararajan Srinivasan. 2022. Topic segmentation in the wild: Towards segmentation of semi-structured & unstructured chats. *arXiv preprint arXiv:2211.14954* (2022).
- [18] Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised Text Segmentation Using Semantic Relatedness Graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. 125–130.
- [19] Zheng Gong, Shiwei Tong, Han Wu, Qi Liu, Hanqing Tao, Wei Huang, and Runlong Yu. 2022. Tipster: A Topic-Guided Language Model for Topic-Aware Text Segmentation. In *International Conference on Database Systems for Advanced Applications*. Springer, 213–221.
- [20] William M Gribbons. 1992. Organization by design: Some implications for structuring information. *Journal of technical writing and communication* 22, 1 (1992), 57–75.
- [21] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. LongT5: Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv:2112.07916* (2021).
- [22] Mandy Guo, Joshua Ainslie, David C Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient Text-To-Text Transformer for Long Sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 724–736.
- [23] Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics* 23, 1 (1997), 33–64.
- [24] Hakan Inan, Rashi Rungta, and Yashar Mehdad. 2022. Structured Summarization: Unified Text Segmentation and Segment Labeling as a Generation Task. *arXiv preprint arXiv:2209.13759* (2022).
- [25] Soyeong Jeong, Jinheon Baek, Chaehun Park, and Jong C Park. 2021. Unsupervised Document Expansion for Information Retrieval with Stochastic Text Generation. In *Proceedings of the Second Workshop on Scholarly Document Processing*. 7–17.
- [26] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth JF Jones, Jussi Karlgren, Aashish Pappu, Sravana Reddy, and Yongze Yu. 2021. TREC 2020 podcasts track overview. *arXiv preprint arXiv:2103.15953* (2021).
- [27] Rosie Jones, Hamed Zamani, Markus Schedl, Ching-Wei Chen, Sravana Reddy, Ann Clifton, Jussi Karlgren, Helia Hashemi, Aashish Pappu, Zahra Nazari, et al. 2021. Current challenges and future directions in podcast information access. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1554–1565.
- [28] Andreas H Jucker. 1992. Conversation: structure or process. *JR Searle et al., (On) Searle on conversation* (1992), 77–90.
- [29] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).
- [30] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text Segmentation as a Supervised Learning Task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 469–473.
- [31] Jeonghwan Lee, Jiyeon Han, Sunghoon Baek, and Min Song. 2023. Topic Segmentation Model Focusing on Local Context. *arXiv preprint arXiv:2301.01935* (2023).
- [32] Jing Li, Aixin Sun, and Shafiq Joty. 2018. SEGBOT: a generic neural text segmentation model with pointer network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4166–4172.
- [33] Raymond Li, Wen Xiao, Linzi Xing, Lanjun Wang, Gabriel Murray, and Giuseppe Carenini. 2022. Human Guided Exploitation of Interpretable Attention Patterns in Summarization and Topic Segmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 10189–10204.
- [34] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
- [35] Jiangyi Lin, Yaxin Fan, Xiaomin Chu, Peifeng Li, and Qiaoming Zhu. 2023. Multi-Granularity Prompts for Topic Shift Detection in Dialogue. *arXiv preprint arXiv:2305.14006* (2023).
- [36] Yang Liu, Chenguang Zhu, and Michael Zeng. 2022. End-to-End Segmentation-based News Summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*. 544–554.
- [37] Zhengyuan Liu, Siti Umairah Md Salleh, Hong Choon Oh, Pavitra Krishnaswamy, and Nancy Chen. 2023. Joint Dialogue Topic Segmentation and Categorization: A Case Study on Clinical Spoken Conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 185–193.
- [38] Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray Buntine. 2021. Transformer over Pre-trained Transformer for Neural Text Segmentation with Enhanced Topic Coherence. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3334–3340.
- [39] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. 2020. Text Segmentation by Cross Segment Attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4707–4716.
- [40] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. 2018. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*. 561–577.
- [41] Pedro Mota, Maxine Eskenazi, and Luísa Coheur. 2019. BeamSeg: A joint model for multi-document segmentation and topic identification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 582–592.
- [42] Rodrigo Nogueira, Jimmy Lin, and Al Epistemic. [n. d.]. From doc2query to docTTTTTquery. ([n. d.]).
- [43] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).
- [44] Min Pan, Teng Li, Yu Liu, Quanli Pei, Ellen Anne Huang, and Jimmy X Huang. 2024. A semantically enhanced text retrieval framework with abstractive summarization. *Computational Intelligence* 40, 1 (2024), e12603.

- [45] Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28, 1 (2002), 19–36.
- [46] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.
- [47] Fabian Retkowski and Alexander Waibel. 2024. From Text Segmentation to Smart Chaptering: A Novel Benchmark for Structuring Video Transcriptions. *arXiv preprint arXiv:2402.17633* (2024).
- [48] Martin Riedl and Chris Biemann. 2012. TopicTiling: a text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 student research workshop*. 37–42.
- [49] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. *Transactions of the Association for Computational Linguistics* 9 (2021), 53–68.
- [50] Imran Sehih, Dominique Fohr, and Irina Illina. 2017. Topic segmentation in ASR transcripts using bidirectional RNNs for change detection. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 512–518.
- [51] Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. 2018. Exploring influence of topic segmentation on information retrieval quality. In *Internet Science: 5th International Conference, INSCI 2018, St. Petersburg, Russia, October 24–26, 2018, Proceedings 5*. Springer, 131–140.
- [52] Swapna Somasundaran et al. 2020. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7797–7804.
- [53] Michael Pepper, Daniel Capurro, Fei Xia, Lucy Vanderwende, and Meliha Yetisgen-Yildiz. 2012. Statistical Section Segmentation in Free-Text Clinical Records. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. 2001–2008.
- [54] Anvesh Rao Vijjini, Hanieh Deilamsalehy, Franck Dernoncourt, and Snigdha Chaturvedi. 2023. Curricular Next Conversation Prediction Pretraining for Transcript Segmentation. In *Findings of the Association for Computational Linguistics: EACL 2023*. 2552–2562.
- [55] Eitan Wagner, Renana Keydar, Amit Pinchevski, and Omri Abend. 2022. Topical Segmentation of Spoken Narratives: A Test Case on Holocaust Survivor Testimonies. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 6809–6821.
- [56] Liang Wang, Sujian Li, Yajuan Lü, and Houfeng Wang. 2017. Learning to rank semantic coherence for topic segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1340–1344.
- [57] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768* (2020).
- [58] Jinxiong Xia, Cao Liu, Jiansong Chen, Yuchen Li, Fan Yang, Xunliang Cai, Guanglu Wan, and Houfeng Wang. 2022. Dialogue Topic Segmentation via Parallel Extraction Network with Neighbor Smoothing. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2126–2131.
- [59] Jinxiong Xia and Houfeng Wang. 2023. A Sequence-to-Sequence Approach with Mixed Pointers to Topic Segmentation and Segment Labeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2683–2693.
- [60] Linzi Xing, Quan Tran, Fabian Caba, Franck Dernoncourt, Seunghyun Yoon, Zhaowen Wang, Trung Bui, and Giuseppe Carenini. 2024. Multi-modal Video Topic Segmentation with Dual-Contrastive Domain Adaptation. In *International Conference on Multimedia Modeling*. Springer, 410–424.
- [61] Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2024. Vidchapters-7m: Video chapters at scale. *Advances in Neural Information Processing Systems* 36 (2024).
- [62] Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)* 10, 4 (2018), 1–20.
- [63] Hai Yu, Chong Deng, Qinglin Zhang, Jiaqing Liu, Qian Chen, and Wen Wang. 2023. Improving Long Document Topic Segmentation Models With Enhanced Coherence Modeling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 5592–5605.
- [64] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*. PMLR, 11328–11339.
- [65] Qinglin Zhang, Qian Chen, Yali Li, Jiaqing Liu, and Wen Wang. 2021. Sequence model with self-adaptive sliding window for efficient spoken document segmentation. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 411–418.
- [66] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 745–754.
- [67] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).
- [68] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938* (2021).