From Text Segmentation to Smart Chaptering: A Novel Benchmark for Structuring Video Transcriptions

Fabian Retkowski[1], **Alexander Waibel**[1,2]
[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh PA, USA
{retkowski,waibel}@kit.edu

Abstract

Text segmentation is a fundamental task in natural language processing, where documents are split into contiguous sections. However, prior research in this area has been constrained by limited datasets, which are either small in scale, synthesized, or only contain well-structured documents. In this paper, we address these limitations by introducing a novel benchmark YTSEG focusing on spoken content that is inherently more unstructured and both topically and structurally diverse. As part of this work, we introduce an efficient hierarchical segmentation model MiniSeg, that outperforms state-of-the-art baselines. Lastly, we expand the notion of text segmentation to a more practical "smart chaptering" task that involves the segmentation of unstructured content, the generation of meaningful segment titles, and a potential real-time application of the models.

1 Introduction

Text segmentation, also occasionally referred to as document segmentation or topic segmentation, is the task of delimiting the boundaries of topically (or functionally) coherent segments of text, placing them in a hierarchical structure, typically a linear one. Text segmentation has been shown to support a number of applications and downstream tasks where long documents are involved, such as information retrieval (Prince and Labadié, 2007; Shtekh et al., 2018; Chivers et al., 2022) or text summarization (Zechner and Waibel, 2000; Cho et al., 2022; Liu et al., 2022b).

Despite its significance, the field currently lacks robust benchmarks, as it is attested and evident in recent works (Lukasik et al., 2020; Glavaš et al., 2021). Most datasets like Choi (Choi, 2000) either suffer from their small scale or are purely synthetic. In practice, WIKI-727K (Koshorek et al., 2018) is the only larger-scale available benchmark, consisting of more than 727,000 Wikipedia documents. However, Wikipedia documents may fall short of fully representing the diversity and complexities of real-world text segmentation challenges. These documents are well-structured, informative, and have a fixed style (in accordance with Wikipedia's Manual of Style). This uniformity may not adequately reflect the unstructured and varied nature of text found in other sources.

Structuring a document proves to be particularly valuable in two cases, both of which frequently occur in spoken and conversational content: first, when the content is inherently unstructured; and second, when the content, being structured or semi-structured, lacks explicit or formal organization. Consequently, text segmentation plays a crucial role in many recently rolled-out AI-powered features in applications such as Discord, YouTube, Microsoft Teams, and Zoom.

Based on this observation and the lack of available, large-scale benchmarks, we have developed a novel benchmark centered around diverse spoken content. We adopt a more holistic and practical approach than prior works, viewing text segmentation as a valuable application that involves both the prediction of segment boundaries as well as the generation of segment titles. Previous works have not considered and evaluated the generation of section headings, which is crucial for practical applications. We term this challenge *smart chaptering* to describe the transformation of unstructured content into a high-level semantic structure with meaningful headings, a critical process for improving document comprehension and organization. This term reflects the additional capabilities required for a system to offer practical utility. Correspondingly, we introduce MiniSeg, a small-scale and state-of-the-art hierarchical segmentation model focused on efficiency, thus viable for use in practical settings. Finally, in addition to traditional offline settings, we also evaluate our approach in online scenarios where real-time processing is crucial, further expanding its practical applicability.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 406–419

In summary, the contributions of our paper are:

- The introduction of a novel larger-scale text segmentation benchmark YTSEG which addresses an important limitation of this research area, the lack of robust benchmarks, and gives researchers a chance to evaluate their models on a benchmark other than WIKI-727K. In addition, it is the first available benchmark around speech text segmentation.
- We introduce MiniSeg, an efficient, hierarchical, state-of-the-art text segmentation model that demonstrates the effectiveness of a number of incremental methodological improvements compared to previous models.
- We extend the (offline) text segmentation task by online segmentation and title generation. For these tasks, we provide a strong set of baselines on our benchmark.

2 YTSEG Dataset

As part of this work, we introduce a new benchmark, YTSEG, to evaluate text segmentation systems on less structured and more diverse content than previous benchmarks aimed for. The dataset consists of 19,299 English YouTube videos with their transcripts and chapters. An example of a YouTube video organized into chapters can be found in the screenshot provided in Figure A1. We processed the data to adapt it for the text segmentation task, aligning the sentences in the transcription with video chapters. In addition, we release YTSEG[TITLES] for the training and evaluation of generative models predicting the chapter titles. By including video and audio data, the benchmark also paves the way for multi-modal approaches. The dataset, along with the instructions and scripts, is available online and released under a CC BY-NC-SA 4.0 license[1].

2.1 Collection

For the dataset, we utilize `yt-dlp` to collect videos, their transcripts, and their chapters. The dataset collection is limited to videos with closed captions and chapters provided by the content creator. In both cases, YouTube exposes whether the information is automatically generated.[2] We note that only a small subset of YouTube videos fall under this category, which motivates the following procedure.

In the first step, we define a wide variety of seed keywords as listed in Section E. Their purpose is to surface a higher-quality and diverse set of videos that are more likely to have manually provided closed captions and chapters. Then, we utilize the YouTube search and various search filters (e.g., to filter videos without closed captions or to surface more recent content or long-form videos). Based on the video search results, we select corresponding channels to be crawled after reviewing a sample of the channel's videos for the audio language and the quality of its closed captions and chapters.

2.2 Preprocessing

Following previous benchmarks for the text segmentation task, our dataset aims to provide segment boundaries on a sentence level. For this, we sentence tokenize the closed captions using the pre-trained PUNKT tokenizer available in the NLTK library (Bird et al., 2009). We annotate the sentences with respective timestamps based on the closed captions. It is important to note that the closed captions do not always respect sentence boundaries, which may necessitate potential splitting and joining of sentences while linearly interpolating timestamps based on their character length. Unlike purely textual datasets, the segment boundaries might not necessarily agree with sentence boundaries (i.e., YouTube chapters can start or end in the middle of a sentence). Thus, sentences spanning two chapters are assigned to the chapter with the greater time-based overlap. We also observed instances where sentences remain unassigned when the first chapter starts later or the last chapter ends earlier than the first, respectively, the last caption. To address this issue, we add an additional "Intro" or "Outro" chapter in these cases.

We exclude all videos for which inconsistent timestamps cannot be fixed[3], final sanity checks are not passed, or our procedure returns an error. These errors can stem from various reasons, such as empty captions, transcripts without punctuation, or a malformed VTT format. This affects 4.70% of the collected videos.

Finally, the data is split into stratified partitions for training, validation, and testing based on the

[1] https://huggingface.co/datasets/retkowski/ytseg

[2] While for closed captions, this information can be accessed directly via `yt-dlp`, for chapters, it is derived from raw data returned by `yt-dlp`'s low-level APIs.

[3] In a number of instances, we observed certain inconsistencies in the provided timestamps fixable by simple rules.

Channel Name	# Videos
YaleCourses	1015
The List	984
Mashed	904
Bestie	577
Google Cloud Tech	551
Linus Tech Tips	461
Unveiled	425
Flipping Physics	390
Looper	385
GeologyHub	364

(a) YouTube Channels by Number of Videos

Channel Name	Length[h]
YaleCourses	907.05
CS50	276.21
Rich Roll	179.16
Andrew Huberman	178.92
The List	168.27
Mashed	159.24
Tech With Tim	131.37
Linus Tech Tips	120.14
SAS Users	106.12
David Bombal	93.09

(b) YouTube Channels by Content Length

Table 1: YouTube Channels by Number of Videos and Content Length

channel identifier (see Table A1a). As part of this process, channels with only a single video form a separate group.

2.3 Data Statistics

The dataset comprises 19,299 videos from 393 channels, amounting to 6,533 content hours. The topics are wide-ranging, covering domains such as science, lifestyle, politics, health, economy, and technology. The videos are from various types of content formats, such as podcasts, lectures, news, corporate events & promotional content, and, more broadly, videos from individual content creators. Table 1a and the analysis depicted in Figure 1 offer insights into the dataset's diversity, while Table 1b shows that the content hours are dominated by the long-form formats such as podcasts and lectures. The dataset's structural diversity is also evident in its data statistics, as depicted in Table 2. In contrast to WIKI-727K, our benchmark exhibits a greater number of segments per document and a higher number of sentences per segment while simultaneously showing a wider variation.

	YTSEG	WIKI-727K
Document Length [Sent.]	196.2 ± 267.2	57.6 ± 46.9
Video Length [min.]	20.3 ± 25.3	–
Segment Length [Sent.]	21.5 ± 34.2	13.6 ± 20.3
Segment Duration [min.]	2.49 ± 2.98	–
Segments per Document	9.12 ± 5.42	3.48 ± 2.23
Title Length [Words]	4.03 ± 2.75	2.01 ± 1.49
Concentration Index[1]	9.50%	24.96%

[1] with $n = 20$

Table 2: Data Statistics for YTSEG and WIKI-727K

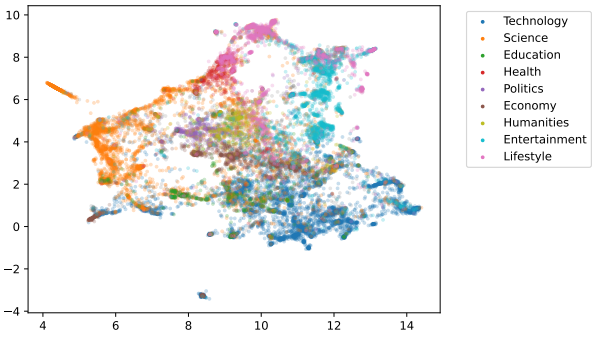

Figure 1: UMAP (McInnes et al., 2018) plot of YTSEG video titles, embedded using Instructor (Su et al., 2023). Category labels are assigned through zero-shot classification with LLaMA 2 (Touvron et al., 2023).

2.4 Chapter Titles

Based on the same data partitioning outlined in Section 2.2, we prepared another view on the same dataset, providing 173,195 pairs of sections and chapter titles. We refer to this dataset view as YTSEG[TITLES]. As a result of the same data partitioning, all section title pairs of a particular video will be assigned to the same data partition. The resulting data split can be found in Table A1b. We removed every pair for which the section title exceeds 75 characters to exclude atypically and excessively lengthy titles[4] accounting for 1.5% of the total number of titles. Despite this, the title length is meaningfully longer and more diverse than in titles in the WIKI-727K dataset (see Table 2). We also point out that the titles in the WIKI-727K dataset are highly concentrated. The 20 most fre-

[4] Some of these lengthy titles tend to be complete sentences or summaries, deviating from our understanding of a title. We also note the computational advantages of excluding them.

quent titles in the dataset account for 24.96% of the overall dataset, with the title "History" alone constituting 7.53% of it. On the contrary, for the YTSEG dataset, this concentration is notably lower, as the top 20 titles collectively represent just 9.50% of the dataset. These titles predominantly consist of functional segments like "Introduction" or "Conclusion".

3 Methodology

In the following, we present the models designed for the text segmentation and title generation task, which are used in our experiments and applied to the newly introduced benchmark. We also elaborate on the modifications we have made to adapt them for online implementation.

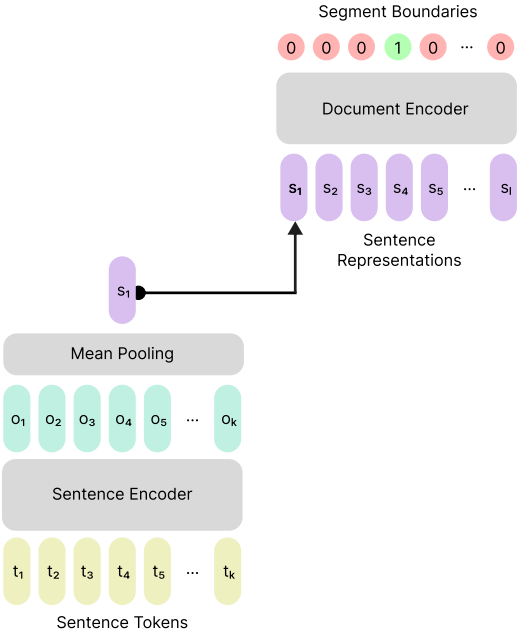

Figure 2: The hierarchical architecture of the segmentation model consists of a sentence encoder and a document encoder returning the binary segment boundaries.

3.1 Offline Segmentation

The model used in our experiments (see the architecture outlined in Figure 2) closely resembles the previous work of Koshorek et al. (2018) and Lukasik et al. (2020). Both proposed a hierarchically structured network consisting of a sentence encoder and a document encoder. The sentence encoder processes each token within a sentence to generate a corresponding sentence representation. Following this, the document encoder performs a sequence labeling task, where its aim is to predict whether each sentence serves as a segment boundary. The network is trained in a supervised fashion using a binary classification objective.

In the following, we highlight the methodological differences between our MiniSeg model and the hierarchical BERT architecture outlined in Lukasik et al. (2020).

- We utilize a pre-trained sentence transformer based on MiniLM (Reimers and Gurevych, 2019; Wang et al., 2020) for the sentence encoder (33M parameters). This network has specifically been trained on paraphrase data in a siamese network structure to produce meaningful sentence representations.

- For the document encoder, we use a randomly initialized RoFormer encoder with 12 layers, 8 attention heads, and 384-dimensional embeddings (26M parameters). This Transformer variant uses rotary positional embeddings (RoPE) introduced by Su et al. (2021).

- Motivated by the class imbalance, we opted to use a weighted binary cross-entropy term. In a thoughtful adjustment, we assigned double the weight to segment boundaries with $w = [1, 2]$. A similar reweighting has been performed by Ghosh et al. (2022).

- Instead of using the `[CLS]` token, we apply mean pooling on the output embeddings to create fixed-sized sentence representations, as it has been shown to outperform other pooling strategies (Reimers and Gurevych, 2019).

- While training, we randomly sample a subset of gradients for the sentence-encoding subnetwork in each backward pass. We set this gradient sampling rate to 0.5, meaning half of the documents are backpropagated through the sentence encoder. In our experiments, this has been shown to reduce the memory requirement while having a regularizing effect and improving the final performance.

3.2 Online Segmentation

Our proposed online segmentation model mirrors the offline segmentation model in its architecture (see Section 3.1). The major difference is that we limit the future context that the model can process, for which we use a different masking strategy that we refer to as *progressive context accumulation*.

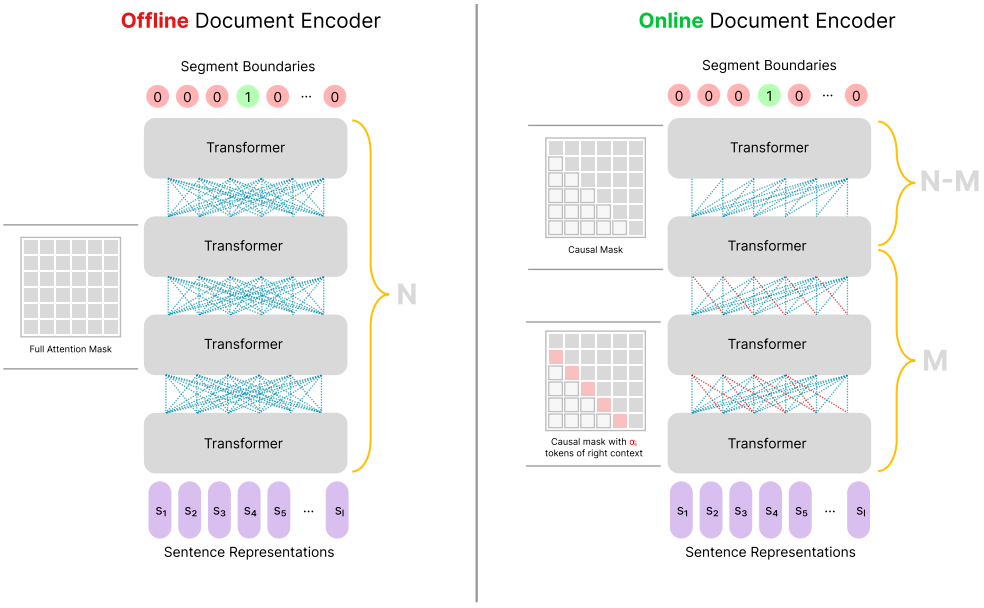

Figure 3: Our offline document encoder is a typical transformer encoder with N transformer layers, each of which applies a full attention mask. Consequently, the encoder can attend to the whole document. In contrast, our online document encoder has $N - M$ layers with causal attention masks that only allow attention to past context, while the initial M layers have attention masks with limited right-side context, that, over these M layers, accumulate to a defined future context size c.

Starting from a purely causal model, we replace a subset of causal attention masks with masks that allow attention to a controlled amount of future context. The corresponding architecture and idea are illustrated in Figure 3.

Specifically, in the early M layers of the document encoder, we employ causal attention masks with an offset that is the limited right-side context α_i where $i \in [1, M]$ is the index of the layer. These masks provide each layer with selective access to a portion of additional future context, and their sizes sum up to our predefined target future context size, denoted as c. In the later $N - M$ layers, we transition back to causal attention masks to prevent any additional future context from leaking into the predictions.

This approach introduces a structural hyperparameter α defining the partitioning distribution of the total future context size c to be allocated to each of the first M layers. We note the relation between the introduced hyperparameters: $M = |\alpha|$ and $c = \sum \alpha_i$.

3.3 Title Generation

We fine-tune a BART-large (Lewis et al., 2020) model on our YTSEG[TITLES] dataset for section title generation. BART is a transformer encoder-decoder model pre-trained on a denoising task.

- For **online title generation**, we limit the amount of text we provide from a given section to the model for generating titles. The term *input span* further refers to the number of starting sentences from a section. This approach enables titles to be prematurely generated in an online setting while maintaining a defined latency (in terms of sentences).

- **Conditional title generation**: To incorporate the context of the document's structure, we prepend previous section titles. This way, the model is conditioned on both the content of the current section and the preceding context, allowing the generation of more relevant and coherent titles. We note that this approach requires generating titles sequentially, which affects offline title generation. In contrast, for online title generation, titles are always generated sequentially.

4 Experiments

4.1 Segmentation

We perform the following experiments to evaluate our benchmark and our segmentation model:

		P (↑)	R (↑)	F1 (↑)	P_k (↓)	B^1(↑)
WIKI-727K	Bi-LSTM[2,4]	69.3 ± 0.1	49.5 ± 0.2	57.7 ± 0.1	–	–
	CS BERT[3,4]	69.1 ± 0.1	63.2 ± 0.2	66.0 ± 0.1	–	–
	Hier. BERT[3,4]	**69.8 ± 0.1**	63.5 ± 0.1	66.5 ± 0.1	–	–
	MiniSeg (Ours)	68.57 ± 0.13	**70.76 ± 0.13**	**69.65 ± 0.09**	17.57 ± 0.06	59.81 ± 0.12
YTSEG	MiniSeg	45.44 ± 0.83	41.48 ± 0.85	43.37 ± 0.60	28.73 ± 0.39	35.74 ± 0.68
WIKI-727K → YTSEG	MiniSeg	48.30 ± 0.84	43.56 ± 0.84	45.81 ± 0.60	27.13 ± 0.43	37.89 ± 0.70
	MiniSeg ($c = 0$)	43.69 ± 0.79	37.49 ± 0.76	40.35 ± 0.55	29.81 ± 0.38	33.11 ± 0.72
	MiniSeg ($c = 1$)	45.05 ± 0.82	40.05 ± 0.80	42.41 ± 0.58	28.70 ± 0.40	34.72 ± 0.79
	MiniSeg ($c = 3$)	46.02 ± 0.88	41.45 ± 0.77	43.61 ± 0.58	28.08 ± 0.43	36.13 ± 0.76
	MiniSeg ($c = 5$)	46.24 ± 0.75	42.23 ± 0.91	44.15 ± 0.60	27.91 ± 0.80	36.62 ± 0.78
	MiniSeg ($c = 8$)	46.92 ± 0.80	41.89 ± 0.79	44.26 ± 0.57	27.68 ± 0.42	36.81 ± 0.77
	MiniSeg ($c = 10$)	45.99 ± 0.81	41.31 ± 0.89	43.52 ± 0.61	27.97 ± 0.39	36.35 ± 0.74
	MiniSeg ($c = 20$)	46.46 ± 0.89	42.34 ± 0.83	44.30 ± 0.61	27.95 ± 0.41	37.07 ± 0.76

[1] Boundary Similarity (Fournier, 2013)
[2] Koshorek et al. (2018)
[3] Lukasik et al. (2020)
[4] Results as reported by Lukasik et al. (2020).

Table 3: Results of our text segmentation models and baselines on WIKI-727K and YTSEG. Standard deviations are estimated by bootstrapping the test set 100 times, similar as in Lukasik et al. (2020).

	P (↑)	R (↑)	F1 (↑)
Zero-Shot			
WIKI-727K	1.58 ± 0.89	15.85 ± 3.61	2.87 ± 1.47
YTSEG	12.55 ± 4.21	8.30 ± 2.78	9.99 ± 2.41
WIKI-727K → YTSEG	6.09 ± 1.92	**18.07 ± 3.36**	9.11 ± 2.19
Fine-Tuned on QMSUM			
No Pre-Training	21.62 ± 4.54	15.61 ± 3.58	18.13 ± 2.90
WIKI-727K	23.45 ± 5.42	11.88 ± 2.75	15.77 ± 2.71
YTSEG	**31.09 ± 5.35**	16.92 ± 3.20	**21.92 ± 2.99**
WIKI-727K → YTSEG	25.21 ± 4.71	15.82 ± 3.15	19.44 ± 2.76

Table 4: Text segmentation results of MiniSeg on the QMSUM dataset, both zero-shot and fine-tuned.

- First, we train our introduced MiniSeg model on the established benchmark WIKI-727K and compare it against the baselines of Koshorek et al. (2018) and Lukasik et al. (2020). For a fair comparison, we use the same setup as in Koshorek et al. (2018) by predicting top-level sections of the document and using the original preprocessing scripts.

- We establish a benchmark for the YTSEG dataset employing our MiniSeg model. This involves training the model on the dataset to set the baseline performance.

- In addition, we experiment with a two-stage training process where we first do a task adaptational pre-training of the model on the WIKI-727K dataset and then fine-tune it on our YTSEG benchmark.

- We test and fine-tune our model on QMSUM (Zhong et al., 2021) to evaluate whether our dataset and model can improve the segmentation of even more unstructured content such as meetings. This dataset provides 232 segmented meetings.

- Finally, we train online segmentation models with different future context sizes c. The corresponding partitioning α for each setting can be found in Table A3.

We evaluate our segmentation models using a combination of standard binary classification metrics, such as precision, recall, and F1 score, as well as metrics specifically tailored for text segmentation tasks, including P_k as introduced in the work of Beeferman et al. (1999) and Boundary Similarity, as discussed in Fournier (2013). The results of the experiments are presented in Table 3 and 4.

MiniSeg. The experiments presented in Table 3 demonstrate that MiniSeg outperforms the baselines, namely Bi-LSTM, cross-segmenter BERT, and hierarchical BERT, on the established WIKI-727K benchmark, even though it is equipped with only 59 million parameters. Given this parameter count, it is meaningfully more efficient compared to state-of-the-art baselines such as hierarchical BERT (220 million parameters) and cross-segmenter BERT (336 million parameters). Several factors contribute to the observed performance as demonstrated in an ablation study shown in Table 5, with the strength of the pre-trained sentence encoder emerging as one of the most crucial contributors. It is worth noting that BERT (Devlin et al., 2019), as used by Lukasik et al. (2020), was not

	P (↑)	R (↑)	F1 (↑)	P_k (↓)	B (↑)
MiniSeg	45.44 ± 0.83	41.48 ± 0.85	43.37 ± 0.60	**28.73 ± 0.39**	**35.74 ± 0.68**
w/o WBCE	**48.76 ± 0.95**	31.66 ± 0.77	38.39 ± 0.64	30.53 ± 0.36	30.42 ± 0.71
w/o RoPE[1]	42.13 ± 0.70	**42.05 ± 0.81**	42.09 ± 0.54	30.59 ± 0.47	33.75 ± 0.69
w/o Pre-Training[2]	38.73 ± 0.81	28.62 ± 0.61	32.92 ± 0.50	33.52 ± 0.36	25.45 ± 0.59
with `[CLS]` pooling	45.76 ± 0.80	41.39 ± 0.85	**43.47 ± 0.59**	28.87 ± 0.39	35.54 ± 0.68
w/o gradient sampling	43.41 ± 0.83	40.38 ± 0.79	41.84 ± 0.57	29.94 ± 0.41	34.10 ± 0.72

[1] A standard transformer with sinusoidal positional encodings is used as the document encoder.
[2] The weights of MiniLM, the sentence transformer, are initialized randomly.

Table 5: Results of ablated versions of MiniSeg on the YTSEG dataset.

trained to represent sentences in particular and that Liu et al. (2019) have described BERT as "significantly undertrained". Additionally, our approach relies on a weighted cross-entropy loss function, allowing us to balance precision and recall. Importantly, while our model exhibits lower precision compared to the baselines and ablated version, it excels in terms of recall. Smaller incremental gains can be attributed to RoPE and the sampling of gradients. Lastly, no noticeable effect is observed when replacing `[CLS]` pooling with mean pooling.

Task Adaptation. We find that the task-adaptational pre-training with WIKI-727K improves the result on the YTSEG benchmark (see Table 3). This outcome contrasts with the findings of Ghosh et al. (2022), who reported that such a pre-training step has a negative or negligible effect when applied to semi-structured content. While the domain is different (YouTube videos versus chat conversations), we emphasize that the dataset in Ghosh et al. (2022) is synthetically constructed and, as such, is qualitatively different from WIKI-727K and YTSEG presumably contributing to the varying effectiveness of the pre-training.

Meeting Segmentation. Our experiments on QMSUM displayed in Table 4 reveal that the pre-training with YTSEG improves the final performance of models on QMSUM. Even in zero-shot conditions, the beneficial effect of YTSEG becomes apparent. This underscores the domain proximity of video content and meetings, both of which are less structured and are spoken in nature. However, we note that due to its small size, QMSUM is not a robust benchmark, and the effect of WIKI-727K remains inconclusive.

Online Segmentation. In the results (see Table 3), we observe noticeable jumps in performance when increasing the future context size. However, diminishing returns set in after about three to five sentences of future context, especially when considering the latency trade-off in online segmentation. This strongly suggests that local context is more important. In fact, even a model without future context at all scores solidly (with the performance only increasing from 40.35 to 45.81 for a model without any future context and one with global context).

4.2 Title Generation

We conducted a series of experiments to evaluate the performance of our title generation model, summarized as follows:

- In our initial experiment, we conducted a comparative analysis of fine-tuning the BART model under two distinct conditions. The first involved training the model to generate titles solely based on the current section text, devoid of any contextual information. In the second setting, the model was trained with the added context of previous titles.

- For the online setting, we assessed different scenarios where the model receives a limited number of starting sentences $s \in [1, 3, 5, 10]$. We conducted this evaluation for both the context-less scenario and the scenario where previous titles were incorporated as context.

The generation of section titles can be considered an extreme form of summarization. As such, we evaluate our models using established metrics in summarization: ROUGE (Lin, 2004), which measures the lexical overlap, and BARTScore (Yuan et al., 2021), an increasingly used metric for semantic equivalence. The results of our experiments are shown in Table 6.

Importance of Context. The results strongly underscore the difficulty in generating chapter titles solely based on the content of the current section

		R1 (↑)	R2 (↑)	RL (↑)	BS[1](↑)
No Context	BART	36.42 ± 0.36	17.03 ± 0.28	36.19 ± 0.35	-4.21 ± 0.02
	BART ($s = 1$)	25.02 ± 0.31	10.98 ± 0.23	24.87 ± 0.30	-4.97 ± 0.02
	BART ($s = 3$)	31.40 ± 0.31	14.32 ± 0.26	31.21 ± 0.32	-4.51 ± 0.02
	BART ($s = 5$)	33.64 ± 0.31	15.61 ± 0.29	33.42 ± 0.35	-4.37 ± 0.02
	BART ($s = 10$)	34.83 ± 0.33	16.18 ± 0.27	34.60 ± 0.37	-4.30 ± 0.02
Previous Titles	BART	42.79 ± 0.34	22.07 ± 0.30	42.45 ± 0.31	-3.87 ± 0.02
	BART ($s = 1$)	27.66 ± 0.35	11.87 ± 0.26	27.47 ± 0.29	-4.83 ± 0.02
	BART ($s = 3$)	36.33 ± 0.37	17.74 ± 0.31	36.08 ± 0.30	-4.25 ± 0.02
	BART ($s = 5$)	36.02 ± 0.32	19.68 ± 0.32	35.74 ± 0.29	-4.09 ± 0.02
	BART ($s = 10$)	41.52 ± 0.33	21.24 ± 0.30	41.21 ± 0.30	-3.94 ± 0.03

[1] BARTScore (Yuan et al., 2021)

Table 6: Results of the title generation models on the YTSEG[TITLES] dataset.

without additional contextual information. An examination of the model's outputs (see Figure 4) reveals how the lack of context leads to degraded coherence between titles of a document. Notably, the model has no knowledge about the placement of the current section in the document, leading to the repetition of functional titles such as "Intro", which occur frequently in the dataset. Similarly, it also cannot generate sequential numbering or uphold uniform stylistic elements across the document's titles. In contrast, supplying the model with previously generated titles results in a meaningful increase in performance. This approach provides the model with the past context of the document structure, enabling stylistic continuity and a smoother flow between the titles.

Online Generation. Expectedly, the model's performance improves as the input span s increases. While diminishing returns are observable, they are less pronounced compared to the segmentation models. It is worth pointing out that the BART model with $s = 3$ and which has access to the previous titles matches the performance of the BART model that has no context at all, once again underscoring the importance of context. Overall, considering both the segmentation models and the title generation models, we see 3 to 5 sentences as a reasonable trade-off between latency and performance for the future context size c and the input span s. We note that c can, in principle, be independently chosen from s for practical smart chaptering systems. The overall latency for the title generation is dependent on the segmentation model, though, as segment boundaries for the span s need to be determined before the generation of the title, as each sentence may belong to the next section.

5 Related Work

A number of benchmarks have been proposed for the text segmentation task. However, the majority are either small in size or synthetically constructed, often by concatenating documents or sections of documents (Choi, 2000; Chen et al., 2009; Glavaš et al., 2016, 2021). Larger-scale benchmarks are scarce and limited to a narrow type of documents such as Wikipedia articles (Koshorek et al., 2018) or specific domains like news (Liu et al., 2022b)[5]. Equally, these datasets have in common that they only encompass documents that are structured in nature. In contrast, research on spoken, conversational, or generally unstructured or semi-structured content is still in its infancy. Lv et al. (2021) and Cho et al. (2022) separately proposed segmentation for lecture video transcripts. Although these works are related to our task, they are only confined to a single domain (lecture videos), while segments are artificially constructed based on the presentation slides. In the context of conversational content, Ghosh et al. (2022) constructed a dataset by joining excerpts of different chat conversations, while Zhong et al. (2021) provided a small set of meeting transcriptions segmented by topic shifts.

Regarding title generation, this study focuses on generating section headings and chapter titles for video transcripts. While research exists in title generation across various domains, including news headlines (Gu et al., 2020; Liu et al., 2020; Cai et al., 2023), product titles (Yang et al., 2023; Zhu et al., 2022), video titles (Zeng et al., 2016; Yu et al., 2023), StackOverflow posts (Liu et al.,

[5] The news text segmentation dataset claimed to be made available by Liu et al. (2022b) is presently inaccessible through the provided GitHub link, effectively yielding WIKI-727K to be the only large-scale benchmark.

Configuring Cloud Operations on Google Cloud – Google Cloud Tech (`xIaaGef1QvI`)

Generated (No Context)	Generated (Previous Titles)	Reference
Intro	Intro	Introduction & agenda
Intro	How to know what's going on in the cloud	How the operations components play together
Operational use cases	Exploring the products of interest for the operations team	Google Cloud Operations Suite
Site Reliability Engineering	Google SRE	SRE Practices
Customer success story: Krikey	Customer success story: Krikey	Customer Story - Krikey
Wrap-up	Wrap up	Wrap up & additional resources

(a) An exemplary output showing duplicate section titles.

8 Email Etiquette Tips - How to Write Better Emails at Work – Harvard Business Review (`1XctnF7C74s`)

Generated (No Context)	Generated (Previous Titles)	Reference
Intro	Intro	Why bother with email etiquette?
Step 1: Have a call to action if appropriate	1. Have a call to action when appropriate	Include CTA in subject line
Stick with one email thread for the same topic	2. Stick with one email thread for the same topic	One email thread per topic
Tip #3	3. Explain why you added in or took out recipients	Manage recipients
Tip #5: summarize the sender's main points	4. Include your main point first followed by context	Summarize in your reply
Tip #2: Include the context	5. summarize the sender's main points in your reply	Start with the main point
6. Hyperlink Whatever Possible	6. Hyperlink whatever possible	Hyperlink whenever possible
Change Default Setting to reply instead of reply all	7. Change default setting to reply instead of reply all	Change default setting to "Reply" (not "Reply all")
Change Undo Send Option to 30 Seconds	8. Change undo send option to 30 seconds	Change undo send options
Outro	Outro	Outro

(b) An exemplary output showing inconsistent numbering and formatting.

Figure 4: A comparison of chapter titles generated by the fine-tuned BART models, both without any context and with previously generated titles, and the reference titles on select examples from YTSEG's validation dataset. A lack of context, as highlighted, leads to repetitive titles, coherence gaps, and variations in writing styles.

2022a; Zhang et al., 2022a, 2023), and pull requests (Irsan et al., 2022; Zhang et al., 2022b), our work addresses the unique challenges of video content structuring. One distinctive aspect lies in ensuring that all section headings of one document not only serve as informative signposts but also maintain a coherent and seamless flow between them. The closest work to ours has been conducted by Zhang et al. (2019), whose proposed model generates hierarchical outlines for Wikipedia documents.

6 Conclusion

In this work, we present a novel benchmark for smart chaptering. The task aims to segment unstructured content, in particular speech, conversations, and transcriptions, in a linear sequence of chapters and provides each chapter with a title. We think this benchmark is a valuable addition to the text segmentation landscape as larger-scale, non-synthetic benchmarks are scarce, and previous research focused primarily on well-structured, homogeneous documents. As part of this, we propose an efficient and state-of-the-art hierarchical segmentation model and a corresponding title-generating model, both of which have also been architected to work online. By combining our proposed segmentation and title generation models, various practical applications are conceivable. For example, content creators, podcasters, and educators could use it to structure their content for their audience. We see our work also as a stepping stone to support even more unstructured content and speech in a broader scope, such as meetings.

Limitations

Our study is subject to several limitations. First, the benchmark only provides English transcriptions which means it cannot assess text segmentation algorithms in languages other than English or be utilized in multilingual or cross-lingual contexts, an important area of research. Second, while the benchmark is inherently multi-modal, our evalu-

ations were conducted solely on models trained on a single modality, which is the transcript, thus ignoring potentially valuable contextual information. Third, we want to note that the latency and real-timeliness of the online chaptering models depend on sentence lengths as the models operate on a sentence-level granularity. This dependence on sentence length restricts our ability to exert precise control over latency. Lastly, our title generation model suffers from exposure bias since it is trained using reference segmentations and prepending reference titles. In practical systems, we rely on both generated segment boundaries and titles, which can potentially lead to error propagation.

Acknowledgements

This research is supported by the project "How is AI Changing Science? Research in the Era of Learning Algorithms" (HiAICS), funded by the Volkswagen Foundation. We also thank Jan Niehues for insightful discussions.

References

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical Models for Text Segmentation. *Machine Learning*, 34(1):177–210.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*, 1st edition. O'Reilly, Beijing ; Cambridge [Mass.]. OCLC: ocn301885973.

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. 2023. Generating User-Engaging News Headlines. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3265–3280, Toronto, Canada. Association for Computational Linguistics.

Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. 2009. Global Models of Document Structure using Latent Permutations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 371–379, Boulder, Colorado. Association for Computational Linguistics.

Brian Chivers, Mason P. Jiang, Wonhee Lee, Amy Ng, Natalya I. Rapstine, and Alex Storer. 2022. ANTS: A Framework for Retrieval of Text Segments in Unstructured Documents. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 38–47, Hybrid. Association for Computational Linguistics.

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*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Fournier. 2013. Evaluating Text Segmentation using Boundary Edit Distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1702–1712, Sofia, Bulgaria. Association for Computational Linguistics.

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:2211.14954 [cs].

Goran Glavaš, Ananya Ganesh, and Swapna Somasundaran. 2021. Training and Domain Adaptation for Supervised Text Segmentation. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 110–116, Online. Association for Computational Linguistics.

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*, pages 125–130, Berlin, Germany. Association for Computational Linguistics.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating Representative Headlines for News Stories. In *Proceedings of The Web Conference 2020*, WWW '20, pages 1773–1784, New York, NY, USA. Association for Computing Machinery.

Ivana Clairine Irsan, Ting Zhang, Ferdian Thung, David Lo, and Lingxiao Jiang. 2022. AutoPRTitle: A Tool for Automatic Pull Request Title Generation. pages 454–458. IEEE Computer Society.

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)*, pages 469–473, New Orleans, Louisiana. 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.

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.

Dayiheng Liu, Yeyun Gong, Yu Yan, Jie Fu, Bo Shao, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. Diverse, Controllable, and Keyphrase-Aware: A Corpus and Method for News Multi-Headline Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6241–6250, Online. Association for Computational Linguistics.

Ke Liu, Guang Yang, Xiang Chen, and Chi Yu. 2022a. SOTitle: A Transformer-based Post Title Generation Approach for Stack Overflow. pages 577–588. IEEE Computer Society. ISSN: 1534-5351.

Yang Liu, Chenguang Zhu, and Michael Zeng. 2022b. End-to-End Segmentation-based News Summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 544–554, Dublin, Ireland. 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. ArXiv:1907.11692 [cs].

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)*, pages 4707–4716, Online. Association for Computational Linguistics.

Tengchao Lv, Lei Cui, Momcilo Vasilijevic, and Furu Wei. 2021. VT-SSum: A Benchmark Dataset for Video Transcript Segmentation and Summarization. ArXiv:2106.05606 [cs].

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861.

Violaine Prince and Alexandre Labadié. 2007. Text Segmentation Based on Document Understanding for Information Retrieval. In *Natural Language Processing and Information Systems*, Lecture Notes in Computer Science, pages 295–304, Berlin, Heidelberg. Springer.

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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. 2018. Applying Topic Segmentation to Document-Level Information Retrieval. In *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia*, CEE-SECR '18, pages 1–6, New York, NY, USA. Association for Computing Machinery.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv:2307.09288 [cs].

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in*

Neural Information Processing Systems, volume 33, pages 5776–5788. Curran Associates, Inc.

Bang Yang, Fenglin Liu, Zheng Li, Qingyu Yin, Chenyu You, Bing Yin, and Yuexian Zou. 2023. Multimodal Prompt Learning for Product Title Generation with Extremely Limited Labels. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2652–2665, Toronto, Canada. Association for Computational Linguistics.

Yakun Yu, Jiuding Yang, Weidong Guo, Hui Liu, Yu Xu, and Di Niu. 2023. TCR: Short Video Title Generation and Cover Selection with Attention Refinement. In *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 245–256, Cham. Springer Nature Switzerland.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating Generated Text as Text Generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Klaus Zechner and Alex Waibel. 2000. DIASUMM: Flexible Summarization of Spontaneous Dialogues in Unrestricted Domains. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. 2016. Title Generation for User Generated Videos. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 609–625, Cham. Springer International Publishing.

Fengji Zhang, Jin Liu, Yao Wan, Xiao Yu, Xiao Liu, and Jacky Keung. 2023. Diverse title generation for Stack Overflow posts with multiple-sampling-enhanced transformer. *Journal of Systems and Software*, 200:111672.

Fengji Zhang, Xiao Yu, Jacky Keung, Fuyang Li, Zhiwen Xie, Zhen Yang, Caoyuan Ma, and Zhimin Zhang. 2022a. Improving Stack Overflow question title generation with copying enhanced CodeBERT model and bi-modal information. *Information and Software Technology*, 148:106922.

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*, SIGIR'19, pages 745–754, New York, NY, USA. Association for Computing Machinery.

Ting Zhang, Ivana Clairine Irsan, Ferdian Thung, DongGyun Han, David Lo, and Lingxiao Jiang. 2022b. Automatic Pull Request Title Generation. In *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 71–81. ISSN: 2576-3148.

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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Wenya Zhu, Yinghua Zhang, Yu Zhang, Yuhang Zhou, Yinfu Feng, Yuxiang Wu, Qing Da, and Anxiang Zeng. 2022. DHA: Product Title Generation with Discriminative Hierarchical Attention for E-commerce. In *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 275–287, Cham. Springer International Publishing.

A YouTube Chapters

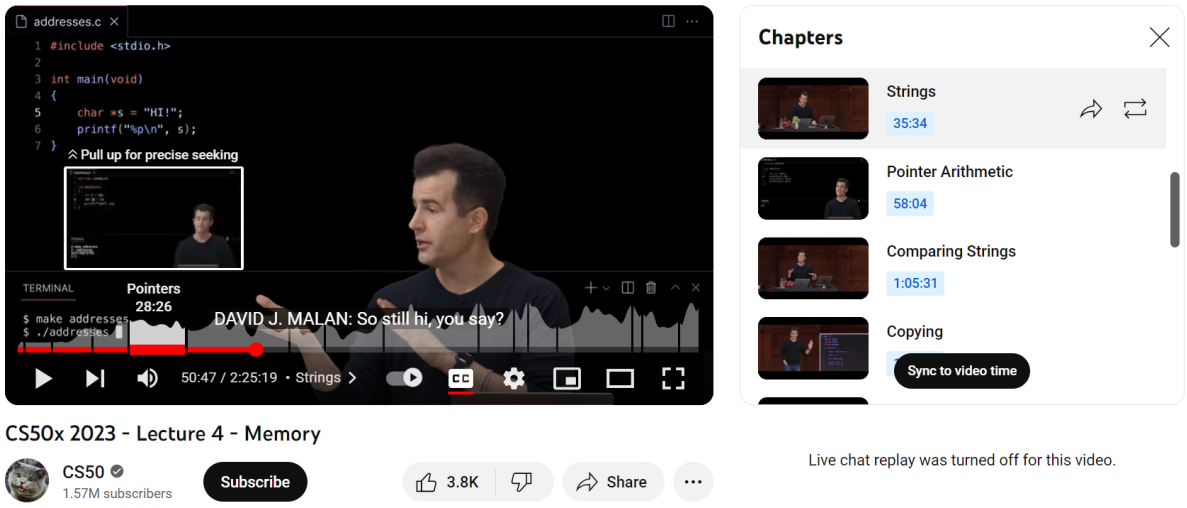

Figure A1: A screenshot of a YouTube video featuring segments as chapters, which form the basis of our new text segmentation benchmark YTSEG.

B Data Splits

Partition	# Examples	
Training	16,404	(85 %)
Validation	1,447	(7.5 %)
Testing	1,448	(7.5 %)
Total	19,229	

(a) YTSEG data split

Partition	# Examples	
Training	146,907	(84.8 %)
Validation	13,206	(7.6 %)
Testing	13,082	(7.6 %)
Total	173,195	

(b) YTSEG[TITLES] data split

Table A1: Data splits for YTSEG and YTSEG[TITLES]

C Hyperparameters

We manually tuned hyperparameters and provide the parameter sets responsible for the results disclosed in this research. While training, we continuously calculated the relevant test metrics (such as the F1 score for segmentation) on the validation data and performed model selection based on this information.

Hyperparameter	Value
Sentence Encoder	`sentence-transformers/all-MiniLM-L6-v2`
Loss Function	Weighted Binary Cross-Entropy
Cross-Entropy Weights	$[1, 2]$
Learning Rate	2.5×10^{-5}
Batch Size	115,000 Tokens
Epochs	15
Learning Rate Schedule	Cosine
Optimizer	AdamW
Dropout Rate	0.1
Gradient Sampling Rate	0.5

Table A2: Hyperparameters for MiniSeg training on the YTSEG dataset

c	M	α
1	1	$[1]$
3	2	$[2, 1]$
5	3	$[2, 2, 1]$
8	4	$[2, 2, 2, 2]$
10	5	$[2, 2, 2, 2, 2]$
20	7	$[4, 4, 4, 2, 2, 2, 2]$

Table A3: Overview of the partitioning α and the number of future-context-accumulating layers M used in the corresponding online segmentation models with future context size c.

Hyperparameter	**Value**
Base	`facebook/bart-large`
Learning Rate	5×10^{-5}
Batch Size	10,000 Tokens
Epochs	2
Learning Rate Schedule	Cosine
Optimizer	AdamW
Dropout Rate	0.1
Decoding Strategy	Beam Sampling
Beam Size	5
Top k	50
Top p	0.95

Table A4: Hyperparameters for training and evaluating the title generation model on YTSEG[TITLE]

D Evaluation

We use the `segeval`[6] package (Fournier, 2013) for the computation of segmentation performance metrics, including P_k and Boundary Similarity. In both cases, we adhere to the default parameter settings. For the evaluation of the title generation models, we rely on the `rouge-metric`[7] package that wraps and reimplements the official ROUGE-1.5.5 Perl script (Lin, 2004). Lastly, for the BARTScore, we utilize the official implementation and ParaBank2-trained BART model[8] provided by Yuan et al. (2021).

E Seed Keywords

- lecture
- podcast
- meetup
- theory
- math
- physics
- chemistry
- climate history
- geometry
- electrical engineering
- media theory
- fashion
- tech
- explained
- "analysis of"
- "introducing"
- "simplified"
- "explanation of"
- "the art of"
- "mechanics of"
- "recent advances in"
- "in a nutshell"
- "the theory of"
- "guide to"
- "why do *"
- "why does *"
- "exploring *"
- "* talk"
- robotics
- computer vision
- virtual reality
- insurance
- dietary
- azure
- brain
- linear algebra
- oled
- silicon
- linux
- deployment
- nature
- adobe
- ui design
- rna
- pytorch
- self driving cars
- machine learning
- data science

[6] https://segeval.readthedocs.io/
[7] https://github.com/li-plus/rouge-metric
[8] https://github.com/neulab/BARTScore