Toward Unifying Text Segmentation and Long Document Summarization

Sangwoo Cho,[1] **Kaiqiang Song,**[1] **Xiaoyang Wang,**[1] **Fei Liu,**[2] **Dong Yu**[1]

[1]Tencent AI Lab, Bellevue, WA
[2]Department of Computer Science, Emory University, Atlanta, GA

{swcho, riversong, shawnxywang, dyu}@global.tencent.com
fei.liu@emory.edu

Abstract

Text segmentation is important for signaling a document's structure. Without segmenting a long document into topically coherent sections, it is difficult for readers to comprehend the text, let alone find important information. The problem is only exacerbated by a lack of segmentation in transcripts of audio/video recordings. In this paper, we explore the role that *section segmentation* plays in extractive summarization of written and spoken documents. Our approach learns robust sentence representations by performing summarization and segmentation simultaneously, which is further enhanced by an optimization-based regularizer to promote selection of diverse summary sentences. We conduct experiments on multiple datasets ranging from scientific articles to spoken transcripts to evaluate the model's performance. Our findings suggest that the model can not only achieve state-of-the-art performance on publicly available benchmarks, but demonstrate better cross-genre transferability when equipped with text segmentation. We perform a series of analyses to quantify the impact of section segmentation on summarizing written and spoken documents of substantial length and complexity.

1 Introduction

One of the most effective ways to summarize a long document is to extract salient sentences (Goldstein et al., 2000). While abstractive strategies produce more condensed summaries, they suffer from hallucinations and factual errors, which pose a more difficult generation challenge (Lebanoff et al., 2020; Goyal and Durrett, 2021). In this study, we focus on extractive summarization of lengthy documents, including both written documents and transcripts of spoken language. Extractive summaries have the potential to be highlighted on their source materials to facilitate viewing, e.g., Google's browser allows text extracts to be highlighted on the webpage via a shareable link (Lyons, 2021).

As a document grows in length, it becomes crucial to bring structure to it. Examples include chapters, sections, paragraphs, headings and bulleted lists (Power et al., 2003). All of them allow readers to find salient content buried within the document. Particularly, having *sections* is a differentiating factor between a long and a mid-length document. A long document such as a research article contains over 5,000 words (Cohan et al., 2018). It is an order of magnitude longer than a mid-length document such as a news article (See et al., 2017). Writing a long document thus requires the author to meticulously organize the content into sections. In this paper, we equip our summarizer with the ability to predict section boundaries and leverage this ability to improve long document summarization.

Importantly, sections are essential to both written and spoken documents. A majority of summarization approaches concentrate on written documents, assuming the sections are given. They exploit document structure by hierarchically building representations from words to sentences, then to larger sections and documents (Xiao and Carenini, 2019; Liu and Lapata, 2019; Narayan et al., 2020). It remains an open question as to whether spoken transcripts can be handled in a similar manner. E.g., the transcript for a 1.5-hour video lecture contains >10,000 words. There are no section boundaries. Instead, the lecture content is loosely organized around a series of talking points. Discourse cues, e.g., "*so next we need to...*," have been shown to correlate with the underlying document structure (Hearst, 1997). We thus aim to leverage such cues to infer section boundaries, which help summarization of both spoken and written documents.

Our model learns robust sentence representations by performing the two tasks of extractive summarization and section segmentation simultaneously, enhanced by an optimization-based framework to select important and diverse sentences. It mimics what a human would do when identifying salient

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 106 - 118
December 7-11, 2022

content from a lengthy document. Text segmentation was previously studied as a standalone problem (Arnold et al., 2019; Xing et al., 2020; Lukasik et al., 2020). For example, Koshorek et al. (2018) break Wikipedia articles into sections according to tables of contents. In this work, we enhance extractive summarization with a new addition of section segmentation. We train our model on written documents with known section boundaries, then adapt it to transcripts where such information is unavailable to exploit its transferability. We observe that by predicting section boundaries, our model learns to not only encode salient content but also recognize document structure information.

Ensuring that a summary covers a broad range of important topics is pivotal. A long document may discuss multiple topics. It is inadequate for a summary to have a narrow information focus and miss the important points of the document. Crucially, we design a new regularizer drawing on learned sentence representations and determinantal point process (Kulesza and Taskar, 2012; Cho et al., 2019a) to ensure a set of representative and diverse sentences is selected for the summary. We evaluate our proposed approach against strong summarization baselines and on multiple datasets ranging from scientific articles to lecture transcripts, whose average document length is 3k–8k words. Our findings suggest that the approach can achieve state-of-the-art performance and demonstrate better transferability when equipped with a segmentation component. Our contributions are summarized as follows.

- We investigate a new architecture for extractive long document summarization that has demonstrated a reasonable degree of transferability from written documents to spoken transcripts.
- Our model learns effective sentence representations by performing section segmentation and summarization in one fell swoop, enhanced by an optimization-based framework that utilizes the determinantal point process to select salient and diverse summary sentences.
- The model achieves state-of-the-art performance on publicly available summarization benchmarks. Further, we conduct a series of analyses to examine why segmentation aids extractive summarization of long documents. Our code and models are available online: `https://github.com/tencent-ailab/Lodoss`

2 Related Work

There is growing interest in generating concise summaries from long documents. Most summarizers are enabled by Transformer-based models that can process *long sequences*. E.g., Longformer (Beltagy et al., 2020) replaces Transformer's self-attention mechanism with dilated sliding window attention to reduce computation and memory usage. Other methods include content-based and temporal sparse attention (Child et al., 2019; Zaheer et al., 2020; Roy et al., 2021; Huang et al., 2021) and hierarchical attention that builds representations from words to sentences and eventually to documents (Zhang et al., 2019; Rohde et al., 2021). Our work builds on Longformer to process input documents of substantial length while focusing on probing document structure for summarization.

While abstractive strategies could produce succinct summaries, they are prone to hallucinations and factual errors that can mislead the reader (Falke et al., 2019; Kryscinski et al., 2020; Maynez et al., 2020; Pagnoni et al., 2021). The problem is exacerbated when the inputs are spoken transcripts, where false starts, repetitions, interjections, ungrammatical sentences are abundant (Shriberg, 1994). They may cause errors to propagate through abstractive systems (Shang et al., 2018; Li et al., 2019; Zhu et al., 2020; Koay et al., 2020, 2021; Zhong et al., 2021; Chen et al., 2022). Instead, we pursue a more flexible strategy to produce extractive summaries, allowing the reader to grasp the essentials without having to read all materials.

Our work differs from previous extractive methods in its focus on document segmentation, which holds promise for summarizing lengthy documents. Important sentences are often located at the beginning or end of documents (Baxendale, 1958; Marcu, 1998). This simple heuristic gives strong results on news summarization (Kedzie et al., 2018; Chen and Bansal, 2018; Narayan et al., 2018; Mao et al., 2021; Liu et al., 2022). We take one step further, jointly partitioning a document into multiple sections and estimating sentence salience given their proximity to section boundaries. We then explore segmentation of written and spoken documents to understand the model's transferability.

Previous studies rely heavily on lexical cohesion to perform text segmentation (Hearst, 1997; Passonneau and Litman, 1997; Malioutov and Barzilay, 2006). Despite their success, establishing coherence in a text goes beyond repeating keywords. A

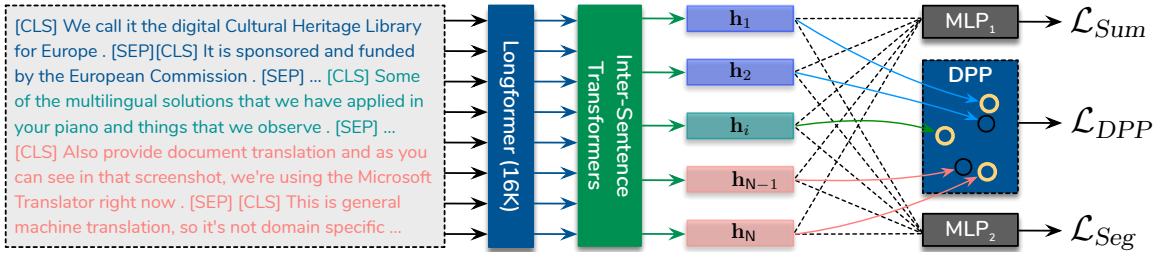

Figure 1: An overview of our system named "***Lodoss***." It builds effective sentence representations by combining two essential tasks of section segmentation and sentence extraction. We introduce a new regularizer $\mathcal{L}_{\text{DPP}}$ drawing on determinantal point processes to collectively measure the quality of a set of extracted sentences, ensuring they are informative and diverse.

good writer often use *discourse cues* to create parallel structures, give examples, compare and contrast, or show addition. Our method draws inspiration from neural text segmentation models to predict section boundaries (Koshorek et al., 2018; Arnold et al., 2019; Xing et al., 2020; Lukasik et al., 2020). In particular, Koshorek et al. (2018) learn sentence representations and label each sentence as ending a segment or not. Lukasik et al. (2020) compare three model architectures based on Transformers and report results on a Wikipedia dataset. In this paper, we focus on unifying section segmentation and summarization into a single optimization framework, exploiting document structure to accurately locate salient content. In what follows, we describe our approach in greater detail.

3 Our Approach

Let $\mathcal{D} = \{s_1, s_2, ..., s_N\}$ be a document containing N sentences.[1] Our goal is to create an extractive summary of the document by selecting a subset of K sentences that retains the most important information. The task of long document summarization is significantly more challenging than other summarization tasks (Daume III and Marcu, 2002). It has a high compression rate, e.g., >85%, excluding most sentences from the summary and suggesting an extractive summarizer must be able to accurately identify summary-worthy sentences.

Fig. 1 shows a schematic overview of our system named "**Lodoss**" (***Lo**ng **do**cument **s**ummarization with **s**egmentation*). It learns robust sentence representations by performing both tasks simultaneously. Further, it introduces a new regularizer drawing on determinantal point processes (Cho et al., 2019b; Perez-Beltrachini and Lapata, 2021) to measure the quality of all summary sentences collectively, ensuring they are informative and have minimum redundancy.

We employ the Longformer (Beltagy et al., 2020) equipped with dilated window attention to produce contextualized token embeddings for an input document. Windowed attention allows each token to attend only to its local window to reduce computation and memory usage. It has the added benefit of easing section segmentation. The left and right context of a section break can be captured by the local window, which reveals any words they have in common and new words seen in this context. Our Longformer model utilizes a large position embeddings matrix, allowing it to process long documents up to 16K tokens. We use dilation, changing window size across layers from 32 (bottom) to 512 (top) to increase its receptive field.

Our summarizer is built on top of Longformer by stacking two layers of inter-sentence Transformers to it. We append [CLS] to the beginning of each sentence and [SEP] to the end, following the convention of (Liu and Lapata, 2019). This modified sequence of tokens is sent to Longformer for token encoding. We obtain the vector of the i-th [CLS] token as the representation for sentence s_i with rich contextual information. These vectors are added to sinusoidal position embeddings, then passed to two layers of inter-sentence Transformers to capture document-level context. Such global context is especially important for identifying salient sentences, whereas sinusoidal position embeddings indicate the relative position of sentences. The output vectors are denoted by $\{\mathbf{h}_i\}_{i=1}^{N}$.

Summarization and Section Segmentation. We address both problems simultaneously in a single framework (Eq. (1-2)). Particularly, $y_{sum,i} = 1$ indicates the i-th sentence is to be included in the summary; $y_{seg,i} = 1$ suggests the sentence starts (or ends) a section. Both tasks are related at their core.

[1]Modern speech-to-text services provide automatic punctuation. Transcripts are punctuated using commas, periods, question marks and semicolons. It allows us to break down a transcript into a sequence of utterances akin to sentences of a written document.

A section usually starts or concludes with summary-worthy sentences; predicting section boundaries helps us effectively locate those sentences. Moreover, the discourse cues for identifying major section boundaries, e.g., "*so next we need to...*," are portable across domains and genres. It allows us to perform a series of ablations to adapt our summarizer from written to spoken documents.

$$\hat{y}_{sum,i} = \sigma(\mathbf{w}_{sum}^{\top}\mathbf{h}_i + b_{sum}) \quad (1)$$

$$\hat{y}_{seg,i} = \sigma(\mathbf{w}_{seg}^{\top}\mathbf{h}_i + b_{seg}) \quad (2)$$

Our base model, "**Lodoss**-*base*," minimizes the per-sentence empirical cross-entropy of the model w.r.t. gold-standard summary labels (Eq. (3)). It learns to identify salient sentences despite that content salience may vary across datasets. Further, our joint model, "**Lodoss**-*joint*," optimizes both tasks through multi-task learning: $\mathcal{L}(\Theta) = \mathcal{L}_{sum} + \mathcal{L}_{seg}$. It adds to the robustness of derived sentence representations, because the acquired knowledge for section segmentation is more transferable across domains. Here, $\hat{y}_{sum,i}$ and $\hat{y}_{seg,i}$ and are predicted scores for summarization and segmentation; $y_{sum,i}$ and $y_{seg,i}$ are ground-truth sentence labels.

$$\mathcal{L}_{sum} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_{sum,i}\log\hat{y}_{sum,i} + (1-y_{sum,i})\log(1-\hat{y}_{sum,i})\Big) \quad (3)$$

$$\mathcal{L}_{seg} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_{seg,i}\log\hat{y}_{seg,i} + (1-y_{seg,i})\log(1-\hat{y}_{seg,i})\Big) \quad (4)$$

A DPP Regularizer. It is especially important to *collectively* measure the quality of a set of extracted sentences, instead of handling sentences individually. We introduce a new regularizer leveraging the determinantal point processes (Kulesza and Taskar, 2012; Zhang et al., 2016; Perez-Beltrachini and Lapata, 2021) to encourage a set of salient and diverse sentences to be selected for the summary. With the DPP regularizer, a ground-truth summary $Y^{'}$ is expected to achieve the highest probability score compared to alternatives. It provides a summary-level training objective that complements the learning signals of our **Lodoss**-*joint* summarizer.

DPP defines a probabilistic measure for scoring a subset of sentences. Let $\mathcal{Y} = \{1, 2, ..., \mathsf{N}\}$ be the ground set containing N sentences. The probability of a subset $Y \subseteq \mathcal{Y}$, corresponding to an extractive summary, is given by Eq. (5), where $\det(\cdot)$ is the determinant of a matrix; $L \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ is a positive semi-definite matrix; L_{ij} indicates the similarity between sentences i and j; L_Y is a principal minor of L indexed by elements in Y; I is an identity matrix of the same dimension as L.

$$\mathcal{P}(Y) = \frac{\det(L_Y)}{\det(L+I)} \quad (5)$$

We make use of the quality-diversity decomposition for constructing L: $L = \mathrm{diag}(\mathbf{q}) \cdot \mathbf{S} \cdot \mathrm{diag}(\mathbf{q})$, where $\mathbf{q} \in \mathbb{R}^{\mathsf{N}}$ represents the quality of sentences; $\mathbf{S} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ captures the similarity of sentence pairs. In our model, the sentence quality score q_i is given by $\hat{y}_{sum,i}$ (Eq. (1)), indicating its importance to the summary. The sentence similarity score is defined by: $S_{i,j} = \cos(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i\mathbf{h}_j}{\|\mathbf{h}_i\|\times\|\mathbf{h}_j\|}$. We employ batch matrix multiplication (BMM) to efficiently perform batch matrix-matrix products.

DPP rewards a summary if it contains a subset of important and diverse sentences. A summary containing two sentences i and j has a high probability score $\mathcal{P}(Y = \{i, j\})$ if the sentences are of high quality and dissimilar from each other. Conversely, if two identical sentences are included in the summary, the determinant $\det(L_Y)$ is zero. Modeling pairwise repulsiveness helps increase the diversity of the selected sentences and eliminate redundancy. As illustrated in Eq. (6), our DPP regularizer is defined as the negative log-probability of the ground-truth extractive summary $Y^{'}$. It has the practical effect of promoting selection of the ground-truth summary while down-weighting alternatives.

$$\mathcal{L}_{DPP} = -\log\frac{\det(L_{Y'})}{\det(L+I)} \quad (6)$$

Our final model, "**Lodoss**-*full*," is shown in Figure 1. It adds the DPP regularizer to the joint model (Eq. (7)); β is a coefficient that balances sentence-level cross-entropy losses and summary-level DPP regularization. Θ are all of our model parameters.

$$\mathcal{L}(\Theta) = (\mathcal{L}_{sum} + \mathcal{L}_{seg}) + \beta\mathcal{L}_{DPP} \quad (7)$$

4 Experiments

In this section, we detail our experimental settings for long document extractive summarization. Our datasets include collections of scientific articles and lecture transcripts, their associated summaries and section boundaries. We contrast our approach with strong summarization baseline systems and report

results on three standard benchmarks. Model ablations and human assessment of summary quality are presented in §5.

4.1 Datasets

For written documents, we choose to experiment with scientific articles (Cohan et al., 2018) as they follow a logical document structure. They come with human summaries and sections, delimited by top-level headings. The scientific articles are gathered from two open-access repositories: arXiv.org and PubMed.com. Particularly, arXiv consists of papers in the fields of mathematics, physics, astronomy, electrical engineering, computer science, and more. PubMed contains research articles and their abstracts on life sciences and biomedical topics. These datasets contain 148K and 216K instances, respectively (Table 1). Their source documents are an order of magnitude longer than standard news articles (Grusky et al., 2018).

For transcript summarization, we utilize lectures gathered from VideoLectures.NET (Lv et al., 2021). These lectures have been automatically transcribed using Microsoft's Speech-to-Text API. Further, the transcripts are time aligned with lecture slides. All utterances aligned to a single slide are grouped into a cluster, they form a transcript *section*. Text extracted from slides are used as ground-truth summaries. The dataset contains a total of 9,616 videos. Each video contains about 33 slides. It helps us lay the groundwork for unifying summarization and segmentation on spoken documents, using lecture slides to provide weak supervision for both tasks.[2] We show data statistics in Table 1, including number of sentences/words per document and summary, reported for train, validation and test sets.

Ground-Truth Labels. A label $y_{sum,i}$ is assigned to each sentence of the document: 1 indicates the sentence belongs to the ORACLE summary, 0 otherwise. An ORACLE is created by adding one sentence at a time incrementally to the summary, so that it improves the average of ROUGE-1 and -2 F-scores (Kedzie et al., 2018). ORACLE summaries give the ceiling performance of an extractive summarizer. ORACLE summaries for scientific papers are created by us; those for lecture transcripts are provided by Lv et al. (2021) generated by aligning transcript utterances with lecture slides.

Scientific papers come with sections: we specify

	#Insts	*Document* #Wds	#Tkns	#Snts	*Summary* #Wds	#Snts
		PubMed				
Train	134,915	3,044	3,865	86.3	202	6.8
Val	6,633	3,112	3,982	87.9	203	6.8
Test	6,658	3,093	3,914	87.5	205	6.9
		arXiv				
Train	203,037	6,038	8,583	206.4	280	9.9
Val	6,436	5,894	8,152	204.2	162	5.6
Test	6,440	5,906	8,132	205.7	163	5.7
		VideoLec				
Train	7,692	4,192	4,901	291.9	456	24.5
Val	962	4,222	4,931	294.1	466	24.7
Test	962	4,387	5,131	306.2	479	25.4

Table 1: Statistics of our datasets. #Wds, #Tkns and #Snts are average number of words, tokens and sentences, respectively. We report these for training, validation and test sets. Tokenization was performed using BPE (Sennrich et al., 2016).

$y_{seg,i} = 1$ if it is the first (or last) sentence of a section, 0 otherwise. Both the first and last sentences of a section could contain discourse connectives indicating a topic shift. Clear document structure depends on establishing where a section ends and the next one begins. For lectures, all transcript utterances are time aligned with lecture slides, creating mini-sections. We explore alternative definitions of transcript sections in §4.3.

System Predictions. At inference time, our system extracts a fixed number of sentences (K) from an input document. These sentences have the highest probability scores according to Eq. (1).[3] K is chosen to be close to the average number of sentences per reference summary. We set K=7 and 5 for the PubMed and arXiv datasets, respectively, following the convention of Xiao and Carenini (2019). We use K=3 for lectures (Lv et al., 2021). Section predictions are given by Eq. (2).

4.2 Experimental Settings

Our implementation uses HuggingFace (Wolf et al., 2020), PyTorch (Paszke et al., 2019) and PyTorch Lightning (Falcon, 2019). We use the Adam optimizer. Its initial learning rate is set to be $3e^{-5}$. The learning rate is linearly warmed up for 10% of the total training steps. The training was performed on 8 NVIDIA Tesla P40 GPUs. The models were trained on each dataset for 20 epochs, using a batch size of 8 with gradient accumulation every 4 steps.

[2]We investigated other transcript datasets (Carletta et al., 2006) and found they do not contain enough training instances.

[3]Our training objective focuses on learning robust sentence representations (Eq. (7)). We choose to extract sentences based on such representations over a DPP inference algorithm. The latter is reported to give fewer summary sentences, yielding high precision but low recall scores (Zhang et al., 2016).

System	PubMed R-1	PubMed R-2	PubMed R-L	arXiv R-1	arXiv R-2	arXiv R-L
Abstractive Systems						
Discourse	38.93	15.37	35.21	35.80	11.05	31.80
TLM-I+E	42.13	16.27	39.21	41.62	14.69	38.03
BigBird-*base*	43.70	19.32	39.99	41.22	16.43	36.96
BigBird-*large*	46.32	20.65	42.33	46.63	19.02	41.77
LED-4K	–	–	–	44.40	17.94	39.76
LED-16K	–	–	–	46.63	19.62	41.83
HAT	48.25	21.35	36.69	46.74	19.19	42.20
Extractive Systems						
ORACLE	61.49	34.70	55.92	59.41	30.05	52.34
LEAD-10	37.45	14.19	34.07	35.52	10.33	31.44
SumBasic	37.15	11.36	33.43	29.47	6.95	26.30
LexRank	39.19	13.89	34.59	33.85	10.73	28.99
ExtSum-LG	44.85	19.70	31.43	43.62	17.36	29.14
... + RdLoss	45.39	20.37	40.99	44.01	17.79	39.09
Sent-PTR	45.01	19.91	41.16	42.32	15.63	38.06
Our System (Extractive)						
Lodoss-*base*	48.10	22.53	43.51	47.64	19.73	41.71
Lodoss-*joint*	48.83	23.13	44.23	47.97	20.13	42.03
Lodoss-*full*	**48.93**	**23.51**	**44.40**	**48.20**	**20.50**	**42.28**
Lodoss-*full*-LG	**49.38**	**23.89**	**44.84**	**48.45**	**20.72**	**42.55**

Table 2: ROUGE results on the PubMed and arXiv datasets.

We run hyperparameter search trials on the validation set, with $\beta \in \{1, \underline{0.1}, 0.01, 0.001\}$. We adopt half-precision (FP16) to speed up training for all models, with the exception of the full model, where full-precision (FP32) is used to ensure a stable performance of eigenvalue decomposition required by the DPP regularizer. The best results are with 16K tokens. We use 4K input for all ablations to save computation unless otherwise noted.

4.3 Summarization Results

Baseline Systems. We compare our system with strong summarization baselines. *SumBasic* (Vanderwende et al., 2007) is an extractive approach that adds sentences to the summary if they contain frequently occurring words. *LexRank* (Erkan and Radev, 2004) measures sentence salience based on eigenvector centrality. *ExtSum-LG* (Xiao and Carenini, 2019, 2020) leverages local and global context to extract salient sentences. *+RdLoss* further adds a redundancy loss term to the learning objective to help the model eliminate redundancy in long document summarization. *Sent-PTR* (Pilault et al., 2020) uses a hierarchical seq2seq sentence pointer model for sentence extraction.

Our abstractive baselines include the following: *Discourse* (Cohan et al., 2018) utilizes a hierarchical encoder to model the document structure and an attentive decoder to generate the summary. *TLM-I+E* (Pilault et al., 2020) generates a paper abstract using the Transformer language model, where the introduction section and extracted sentences are provided as context. *BigBird* (Zaheer et al., 2020) and *LED* (Beltagy et al., 2020) use sparse attention and windowed attention to process long input sequences. *HAT* (Rohde et al., 2021) adds hierarchical attention layers to an encoder-decoder model to summarize long documents.

	System	P	R	F	R-1	R-2	Avg(R)
None	LexRank	17.38	3.66	5.07	17.07	5.78	10.02
	TextRank	21.26	4.38	6.10	20.50	6.68	11.85
	Lo-jnt-sgl	43.94	7.83	12.09	23.86	15.11	20.60
	Lo-fll-sgl	47.87	8.48	13.11	24.12	15.85	21.04
arXiv	Lo-jnt-sgl	46.44	8.12	12.63	24.46	15.93	21.29
	Lo-fll-sgl	47.18	8.39	12.97	24.38	16.04	21.28
	Lo-jnt-grp	49.31	8.80	13.59	25.01	16.93	22.02
	Lo-fll-grp	48.00	8.30	12.95	24.34	16.18	21.31
PubMed	Lo-jnt-sgl	48.11	8.44	13.02	24.76	16.27	21.57
	Lo-fll-sgl	47.69	8.52	13.08	24.61	16.24	21.49
	Lo-jnt-grp	51.00	9.17	14.10	24.89	16.88	21.90
	Lo-fll-grp	49.29	8.97	13.69	24.72	16.63	21.72

Table 3: Results on lecture transcripts. The metrics reported are Precision, recall, F-scores, and Rouge scores. Our model can be trained from scratch, or pretrained on either arXiv or PubMed. We explore alternative definitions of *sections*: all utterances aligned to a single slide is a section ('sgl') vs. using six major sections per transcript ('grp').

Results on Scientific Papers. We compare three of our model variants, listed below. Standard evaluation metrics (ROUGE; Lin, 2004), including R-1, R-2 and R-L, are used to measure the quality of system summaries. It allows our model to be directly compared to previous approaches. More ablations and human assessment are provided in §5.

- **Lodoss**-*base*, using $\mathcal{L}_{sum}$
- **Lodoss**-*joint*, using $\mathcal{L}_{sum} + \mathcal{L}_{seg}$
- **Lodoss**-*full*, using $(\mathcal{L}_{sum} + \mathcal{L}_{seg}) + \beta\mathcal{L}_{DPP}$

Results on PubMed and arXiv datasets are shown in Table 2. Our models strongly outperform both extractive and abstractive baselines, suggesting the effectiveness of unifying section segmentation with summarization. The LEAD baseline, however, does not perform as well on long documents as it does on news articles. It is interesting to note that our models are trained with indirect signals, i.e., binary sentence labels derived from reference summaries, and they remain quite effective at capturing salient content on long documents.

We conduct significance tests using the approximate randomization method (Riezler and Maxwell, 2005; Dror et al., 2018). With a confidence level

	Lodoss	*PubMed* ROUGE-1 P(%)	R(%)	F(%)	ROUGE-2 P(%)	R(%)	F(%)	#Wds	*arXiv* ROUGE-1 P(%)	R(%)	F(%)	ROUGE-2 P(%)	R(%)	F(%)	#Wds
5-Sent	*base*	50.25	49.56	47.75	23.28	22.19	21.73	204.3	42.10	55.13	46.04	16.71	21.84	18.24	216.5
	joint	50.75	49.31	47.85↑	23.72	22.24	21.96↑	202.3↓	43.56	54.21	46.50↑	17.33	21.60	18.49↑	204.9↓
	full	51.27	49.16	48.04↑	24.01	22.26	22.11↑	198.6↓	43.37	54.61	46.59↑	17.27	21.75	18.53↑	207.2↓
6-Sent	*base*	47.58	53.57	48.32	22.18	24.21	22.18	235.9	38.93	58.85	45.27	15.67	23.76	18.23	250.6
	joint	48.06	53.26	48.43↑	22.57	24.21	22.39↑	233.6↓	40.28	58.09	45.87↑	16.25	23.57	18.54↑	238.5↓
	full	48.57	53.17	48.65↑	22.88	24.27	22.57↑	229.8↓	40.18	58.36	45.93↑	16.22	23.69	18.56↑	240.1↓
7-Sent	*base*	45.19	56.64	48.21	21.22	25.82	22.33	265.8	36.23	61.76	44.17	14.80	25.41	18.08	283.4
	joint	45.68	56.49	48.47↑	21.57	25.90	22.58↑	263.1↓	37.45	61.09	44.81↑	15.33	25.22	18.39↑	270.9↓
	full	46.06	56.40	48.60↑	21.84	25.95	22.73↑	260.4↓	37.32	61.39	44.83↑	15.28	25.37	18.40↑	273.3↓

Table 4: We vary the length of output summaries to contain 5-7 sentences and report summarization results on PubMed and arXiv. Our model **Lodoss**-*full* consistently outperforms other variants across all lengths and evaluation metrics.

of 99%, all of our Lodoss models are significantly better than BigBird-base and LED-4K. The differences between our model variants are also significant: between **Lodoss**-*base* and **Lodoss**-*joint*, between **Lodoss**-*joint* and **Lodoss**-*full*. Our results indicate that incorporating section segmentation and a summary-level DPP regularizer can help the model better locate salient sentences. Moreover, the large encoder ('-LG') results in improvements on both datasets.

Results on Lecture Transcripts. We could train our model from scratch using lecture transcripts, or pretrain the model on either arXiv or PubMed, then fine-tune it on transcripts. Results are shown in Table 3. The metrics reported are precision, recall, F-scores and Rouge scores. [4] We observe that models pretrained on written documents perform substantially better compared to training a model from scratch, and PubMed outperforms arXiv consistently except Lo-full-grp. It suggests that knowledge gained from summarizing written documents could be transferred to summarization of spoken transcripts. This is especially the case for our joint model (Lo-joint-⋆), where the models is equipped with the ability to recognize section boundaries. The Lo-joint-⋆ model consistently outperforms the model trained from scratch regardless of different segmentation labels. Note that F-scores are not necessarily aligned with the Rouge scores as the system can predict sentences with similar context that are not labeled as summaries.

We explore alternative definitions of a *transcript section*: all utterances aligned to a single slide is considered a section vs. using six major sections per transcript. The former leads to about 33 sections per transcript. The latter is achieved by finding 6 slides that are discussed the most, and using the first utterance of each slide as the start of a new section. Because scientific papers on PubMed and arXiv contain 6.06 and 5.68 sections averagely, this definition allows our model to be pretrained and fine-tuned under similar conditions. We find that using six major sections per transcript improves summarization performance.

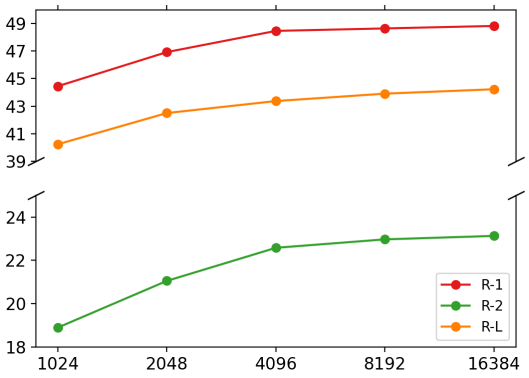

Figure 2: Effect of varying source sequence lengths (PubMed); x-axis shows the source sequence length measured by number of tokens; y-axis shows the ROUGE scores.

5 Ablations and Analyses

Effect of Summary Length. We vary the length of output summaries to contain 5-7 sentences and report summarization results on PubMed and arXiv (Table 4). Our model **Lodoss**-*full* consistently outperforms other variants across all lengths and evaluation metrics. The highest scores are obtained for PubMed with 7 output sentences, whereas 5 sentences work best for arXiv, as it gives a good tradeoff between recall and precision.

[4] Ground-truth abstractive summaries are unavailable for this dataset. We use sentences labeled as summaries to compute Rouge scores. We could not directly compare our results to those of (Lv et al., 2021) due to different settings used.

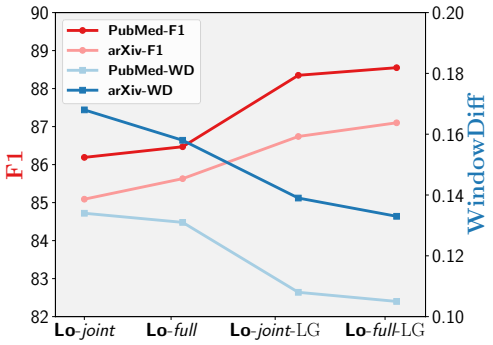

Figure 3: Section segmentation results evaluated by F1 (higher is better) and WinDiff (lower is better). Results are reported for PubMed and arXiv. Best performance is achieved with our **Lodoss**-*full* model. '-LG' means a Longformer-large model is used to encode the input document.

Model	Pos	PubMed		arXiv	
		F1	Avg-R	F1	Avg-R
Lodoss-*joint*	1st	86.19	38.73	85.09	36.71
Lodoss-*full*	1st	86.47	38.95	85.63	36.99
Lodoss-*full*-LG	1st	88.74	39.37	87.10	37.24
Lodoss-*joint*	Last	84.93	38.42	77.07	36.69
Lodoss-*full*	Last	85.26	38.92	78.41	36.95
Lodoss-*full*-LG	Last	87.19	39.36	81.71	37.25

Table 5: Segmentation (F1) and Summarization (ROUGE) results using different segmentation labels.

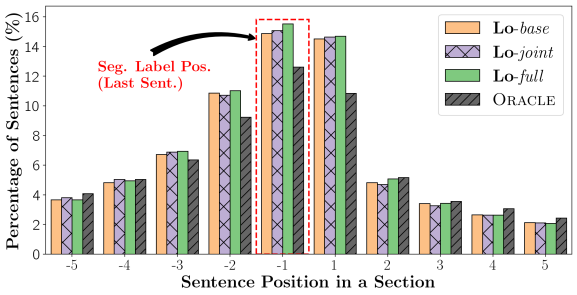

Figure 4: How often summary sentences are found near section boundaries (PubMed). "1" indicates a summary sentence is the first sentence of a section, whereas "-1" indicates it is the last sentence of a section. Both the first and last sentences of a section are likely to be selected for inclusion in the summary.

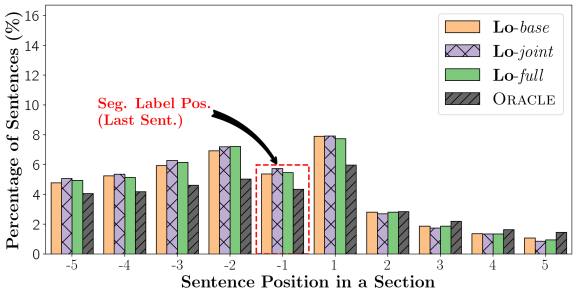

Figure 5: How often summary sentences are found near section boundaries (arXiv).

Effect of Source Sequence Length. We observe that our model performs increasingly better when longer source sequences are used (Figure 2). This is expected, as importance information will be left out if source sequences need to be truncated to a certain length. For example, using 4K tokens, we have to truncate about 50% of arXiv inputs.

Model's Performance on Section Segmentation. Figure 3 shows segmentation results for PubMed and arXiv. Our goal is to predict the first sentence of a section. F1 and WindowDiff scores (Pevzner and Hearst, 2002) are reported. Particularly, WindowDiff is a lenient measure for segmentation results. It uses a sliding window to scan through the input document. At each step, it examines whether the predicted boundaries are correct within the local window. We observe that both our full model and large pretrained models help the system to better predict section boundaries. Predicting the first sentence of a section is easier than predicting the last sentence (Table 5). This gives 4% and 6% gain, respectively, on PubMed and arXiv.

Effect of Our DPP Regularizer. Table 4 shows the average number of words per summary, where summaries are produced by different model variants. We find that summaries produced by **Lodoss**-*full* tend to be shorter compared to other summaries, and **Lodoss**-*full* remains the best performing model. It suggests that the DPP regularizer favors a diverse set of sentences to be included in the summary. The selected sentences are not necessarily long as they may contain redundant content.

Why Section Segmentation is Necessary. We investigate how often summary sentences are found near section boundaries. Results are shown in Figure 4 and 5, respectively for PubMed and arXiv. "1" indicates a summary sentence is the first sentence of a section, whereas "-1" indicates it is the last sentence of a section. Overall, both the first and last sentences of a section are likely to be selected for inclusion in the summary. The effect is stronger for PubMed and less so for arXiv. We conjecture that because arXiv papers are twice as long as PubMed papers, summary sentences may not always occur near section boundaries. In both cases, our models are able to leverage this characteristic to simultaneously identify summary sentences and perform section segmentation.

Human Assessment of System Summaries. We focus on evaluating informativeness and diversity of summary sentences. Other criteria are not considered because extractive summaries can be high-

	Model	Avg.	4/5	3	1/2
Info↑	**Lodoss**-*joint*	2.89	10.05%	21.84%	68.11%
	Lodoss-*full*	**3.12**	**14.70**%	21.49%	**63.81**%
Div↓	**Lodoss**-*joint*	2.14	5.50%	13.16%	81.34%
	Lodoss-*full*	**2.03**	**3.40**%	14.06%	**82.54**%

Table 6: Human evaluation results for informativeness (higher is better) and diversity (lower is better).

lighted on their source materials, allowing them to be understood in context. Our evaluation focuses on the **Lodoss**-*joint* and **Lodoss**-*full* models. The informativeness and diversity metrics are defined as follows. As a toy example, let $\mathcal{S}_1$={1, 3, 7, 12} and $\mathcal{S}_2$={2, 3, 7, 9} be summaries produced by two models, respectively. The numbers are sentence indices. For informativeness, we take the union of summary sentences {1, 2, 3, 7, 9, 12} and ask human evaluators to judge the relevance of each sentence against the ground-truth summary on a scale of 1 (worst) to 5 (best). The informativeness score of a summary is the average of its sentence scores. For diversity, we obtain the symmetric difference of two summaries {1, 2, 9, 12} and ask humans to judge if each sentence has offered new content different from those of the common sentences {3, 7}. A good summary should contain diverse sentences that are dissimilar from each other.

Results are reported in Table 6. We performed evaluation using Amazon Mechanical Turk on 100 randomly selected summarization instances from arXiv. arXiv is chosen over other datasets because its content are more accessible to crowd workers. We recruited Masters turkers to work on our task. They must have completed at least 100 HITs and with $\geq$90% HIT approval rate. Each summary sentence was judged by 3 turkers. Overall, we find that **Lodoss**-*full* receives better relevancy and diversity ratings than **Lodoss**-*joint*. A substantial portion of the sentences receive a score of 1 or 2. It suggests that the extracted sentences lack informativeness despite that the DPP regularizer is effective at increasing the diversity of selected sentences and eliminating redundancy.

6 Conclusion

We tackle the problem of long document extractive summarization by combining two essential tasks of section segmentation and sentence extraction. We further design a regularizer drawing on determinantal point processes to ensure a set of representative and diverse sentences are selected for the summary. Extensive experiments and ablations demonstrate the effectiveness of our proposed approach. Our future work includes exploration of various text segmentation techniques to improve our understanding of the latent document structure. Another direction would be to extend our study to the realm of neural abstractive summarization with the help of learned document structure.

7 Limitations

The proposed summarization models are trained on scientific articles that are segmented into multiple sections by authors. Those section boundaries are utilized by the model to learn robust sentence representations and estimate sentence salience given their proximity to section boundaries. When section boundaries are unavailable, the model may not work as intended. Moreover, trained models may carry inductive biases rooted in the data they are pretrained on. Finetuning on target datasets helps mitigate the issue as the model has been shown to demonstrate a reasonable degree of transferability from written documents to other genres.

Acknowledgements

We are grateful to the reviewers for their insightful comments that have enriched our paper. Fei Liu is supported in part by National Science Foundation grant IIS-2303655.

References

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*, 7:169–184.

P. B. Baxendale. 1958. Machine-made index for technical literature: An experiment. *IBM J. Res. Dev.*, 2(4):354–361.

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

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2006. The AMI meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction*, pages 28–39. Springer Berlin Heidelberg.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

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

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019a. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1027–1038, Florence, Italy. Association for Computational Linguistics.

Sangwoo Cho, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. 2019b. Multi-document summarization with determinantal point processes and contextualized representations. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 98–103, Hong Kong, China. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. 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 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Hal Daume III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 449–456, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by*, 3.

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

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

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

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, and Fei Liu. 2021. A sliding-window approach to automatic creation of meeting minutes. In *Proceedings of the*

2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 68–75, Online. Association for Computational Linguistics.

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.

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

Alex Kulesza and Ben Taskar. 2012. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc.

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

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

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

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 preprint arXiv:2106.05606*.

Kim Lyons. 2021. Google introducing a feature in chrome 90 to create links to highlighted text on a webpage. *The Verge*.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Sydney, Australia. Association for Computational Linguistics.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed H. Awadallah, and Dragomir Radev. 2021. Dyle: Dynamic latent extraction for abstractive long-input summarization.

Daniel Marcu. 1998. Improving summarization through rhetorical parsing tuning. In *Sixth Workshop on Very Large Corpora*.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Multi-document summarization with determinantal point process attention. *J. Artif. Int. Res.*, 71:371–399.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

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

Richard Power, Donia Scott, and Nadjet Bouayad-Agha. 2003. Document structure. *Computational Linguistics*, 29(2):211–260.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *CoRR*, abs/2104.07545.

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*, 9:53–68.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.

Elizabeth Shriberg. 1994. Preliminaries to a theory of speech disfluencies. *Ph.D. thesis, Department of Psychology, University of California, Berkeley*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*, 43(6):1606–1618.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, Suzhou, China. Association for Computational Linguistics.

Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. 2020. Improving context modeling in neural topic segmentation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the*

10th International Joint Conference on Natural Language Processing, pages 626–636, Suzhou, China. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

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.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.