# TIB: A Dataset for Abstractive Summarization of Long Multimodal Videoconference Records

Théo Gigant, Frédéric Dufaux, Camille Guinaudeau, Marc Decombas

# TIB: A Dataset for Abstractive Summarization of Long Multimodal Videoconference Records

Théo Gigant
theo.gigant@l2s.centralesupelec.fr
Université Paris-Saclay, CNRS, CentraleSupélec,
Laboratoire des signaux et systèmes
Gif-sur-Yvette, France
JustAI
France

Frédéric Dufaux
frederic.dufaux@l2s.centralesupelec.fr
Université Paris-Saclay, CNRS, CentraleSupélec,
Laboratoire des signaux et systèmes
Gif-sur-Yvette, France

Camille Guinaudeau
guinaudeau@nii.ac.jp
Japanese French Laboratory for Informatics, CNRS
Japan
University Paris-Saclay
France

Marc Décombas
marc@justai.co
JustAI
France

## ABSTRACT

Large language models and multimodal language-vision models give impressive results on current available summarization benchmarks, but are not designed to handle long multimodal documents. Most summarization datasets are composed of either mono-modal documents or short multimodal documents. In order to develop models designed for understanding and summarizing real-world videoconference records that are typically around 1 hour long, we propose a dataset of 9,103 videoconference records extracted from the German National Library of Science and Technology (TIB) archive, along with their abstract. Additionally, we process the content using automatic tools in order to provide the transcripts and key frames. Finally, we present experiments for abstractive summarization, to serve as baseline for future research work in multimodal approaches.

## CCS CONCEPTS

• **Information systems** → **Summarization**; • **Computing methodologies** → **Natural language generation**; **Video summarization**; **Visual content-based indexing and retrieval**.

## KEYWORDS

multimedia dataset, multimodal documents, automatic summarization

## 1 INTRODUCTION

Oral presentations are the preferred form of conveying information in a lot of situations including conferences, meetings and lectures. During a presentation, a speaker talks to an audience and uses both speech and visual aids (often in the form of a slide show) in order to deliver a message. According to [9], the number of videoconference meetings skyrocketed during COVID-19 pandemic, with an average 12 meetings each month per person, and stabilized around 9 to 10 meetings in 2022. In education, lectures were also replaced by video lectures during the pandemic, with different layouts, including videoconference records [28]. Scientific communication was also disrupted by the COVID-19 with in-person conferences being replaced by virtual conferences [4]. Virtual or hybrid conferences will likely continue to be a popular format in the future as suggested in [16], and can easily be recorded.

Real world videoconference presentation records are composed of speech, camera and shared-screen video streams, that are audio and video modalities respectively. The records usually lasts from dozens of minutes to a few hours, and the shared-screen is often a slideshow that can be reduced to a few dozens key frames.

In these presentations, the slide show is often used to carry information that illustrates the point, helps understand the outline and is not always redundant with the speech.

Abstractive summarization aims at creating a shorter human-readable summary from a long document, while preserving most of the information. Summarization datasets that are publicly available focus either on multimodality or long-form documents, but never both. As a consequence, state-of-the-art summarization methods are not evaluated on their performance at summarizing long documents that have multiple modalities, such as videoconference records.

In this work, we create a large dataset to address this challenge and make long-range multimodal summarization of videoconference records training more accessible. TIB is a dataset of 9,103 records of presentations along their human-written abstract, collected from the archive of the German National Library of Science and Technology: *Technische Informationsbibliothek* (TIB). The data covers records of scientific presentations, such as lectures and conferences, along metadata and a target summary. The videos were processed in order to extract the transcript and the slideshow, using automatic tools. Moreover, we present a statistical analysis on the collected dataset. Finally we carry out preliminary experiments for abstractive summarization using only the text modality derived from the audio transcript. These results can be used as a baseline in future research work in multimodal approaches.

The dataset is shared on the HuggingFace dataset hub[1], with train, validation and test splits containing respectively 80% (7,282), 10% (910) and 10% (911) of the dataset records.

[1] https://huggingface.co/datasets/gigant/tib

## 2 RELATED WORK

Our work is related to multiple topics: text summarization, long range text summarization, multimodal video summarization and multimodal multi-document summarization.

To our knowledge, no existing dataset focus on summarization of long videos, with multiple modalities as input, and a textual summary as output. TIB is also different from most video summarization tasks as the visual modality is very redundant and often pictures documents with low granularity, such as slides.

### 2.1 Text Summarization

Automatic Text Summarization systems are often focused on relatively short text documents, and ignore inputs and outputs in a modality that is not text, and text that is too long [10]. Most standard datasets for benchmarking single document text summarization models focus on summarizing news articles, the most common being CNN/DailyMail, introduced in [23]. Existing methods, trained on such datasets, achieve low accuracy and efficiency for summarizing longer texts [18].

### 2.2 Long Range Text Summarization

Some datasets focus on longer documents, such as academic papers, reports, patents or state bills. ArXiv and PubMed [5] are summarization datasets comprised of scientific papers collected from online repositories. In both of these datasets, the input is the full text of the article, and the target summary is the abstract. BIGPATENT [29] is a dataset of 1.3 million records of U.S. patent documents with their human-written abstractive summary. GOVREPORT [14] is a collection of U.S. Government Accountability Office reports with human-written summaries. BillSum [17] is comprised of Congressional and California state bills along with human-written summaries. BookSum [18] is a dataset for long-form abstractive summarization of books.

### 2.3 Multimodal Video Summarization

As shown in [1], most video summarization benchmarks ignore the audio modality and focus on extracting from an input video a set of representative video frames or video fragments, called respectively video storyboards and video skims. The most common datasets for video summarization are SumMe [13] and TVSum [30]. SumMe is a dataset for supervised video summarization, video skims composed of human selected "*superframes*" are the target summaries. TVSum is an unsupervised video summarization dataset containing videos with shot-level importance score and titles, video skim summaries are created using shots that are representative and relevant to both the title and the video.

How2 videos [27] is a multimodal abstractive summarization dataset. In the 300h version, both audio and visual modalities of the input videos are provided along with a transcript, in order to predict the textual human-written summary, but the videos are very short as most of them are shorter than 20 seconds. In [21], the authors introduce a large-scale dataset for Video-based Multimodal Summarization with Multimodal Output comprised of 184 thousand samples. Each sample includes an article, its textual summary and a video with a cover picture. The authors of [3] are using data from a TV series, split into logical story units that can be inserted into Character-oriented video-skim summaries, evaluated with a large scale user study. In [11], the visual features and transcripts of TED Talks records are processed and combined in order to identify and rank semantic segments of educational videos in order to propose a "hot spots" video summary.

### 2.4 Multimodal Multi-Document Summarization

Multi-document summarization datasets can also include multiple modalities. For instance the authors of [20] introduce a multimodal multi-document corpus of 20 documents and 5-10 videos, along with human-written summaries for each of 50 news events. Half of the dataset is in English and half is in Chinese. Some characteristics of the multimodal multi-document summarization task, such as long input and multimodality, are similar to our work. However, the structure of the input is quite different as it is composed of multiple documents of varying length instead of one long video.

## 3 COLLECTION METHODOLOGY

### 3.1 Dataset Collection

The dataset was first assembled by crawling the TIB-AV portal[2] which is a large archive of videos, developed by the German National Library of Science and Technology: *Technische Informationsbibliothek* (TIB).

### 3.2 Data filtering

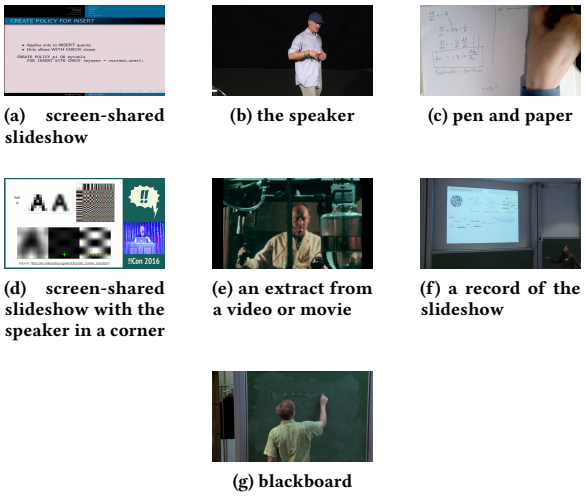
Entries with missing abstracts or abstracts that were too short (less than 30 characters) were filtered out. We also filtered out records for which the abstract or the transcript is in another language than English. To find abstracts in other languages than English, we are using an automatic language detection model: XLM-RoBERTa [6] finetuned on the Language Identification dataset[3], which is the concatenation of XNLI [7], The Multilingual Amazon Reviews Corpus [15] and the Machine translated multilingual STS benchmark dataset[4]. We filter the transcripts in other languages than English by using the language predicted by the multilingual speech recognition system described in Section 4.2. In order to keep the abstracts that are relevant to the associated record, we removed documents if the abstract is the same as the abstract for another video. This allowed to get rid of all the abstracts that were written for a set of records such as conferences, instead of specifically written for a single presentation.

## 4 DATASET COMPOSITION

Each record consist of the following attributes:

- `doi`: digital object identifier (DOI) of the record or the associated paper
- `title`: title of the presentation
- `url`: URL of the record in the TIB archive
- `video_url`: URL of the video file
- `license`: license of the record

[2] https://av.tib.eu/
[3] https://huggingface.co/datasets/papluca/language-identification
[4] https://huggingface.co/datasets/stsb_multi_mt



(a) screen-shared slideshow
(b) the speaker
(c) pen and paper
(d) screen-shared slideshow with the speaker in a corner
(e) an extract from a video or movie
(f) a record of the slideshow
(g) blackboard

**Figure 1: Some example frames from the visual modality**

- `subject`: academic field (*eg* Computer Science, Mathematics, ...)
- `genre`: type of presentation (*eg* Lecture, Conference, ...)
- `release_year`: year the record was released
- `author`: name of the author
- `contributors`: name of the contributors
- `abstract`: the abstract of the presentation, that serve as a target summary
- `transcript`: the automatically extracted transcript
- `transcript_segments`: the automatically extracted transcript with time codes, output of the speech recognition system
- `keyframes`: the automatically extracted key frames time codes

`doi`, `title`, `url`, `video_url`, `license`, `subject`, `genre`, `release_year`, `author`, `contributors` and `abstract` are provided as found in the TIB archive. The length, style, quality and content of the abstract can differ from video to video as it was likely provided by each author. For instance, some abstracts can provide very short title-like summaries, introduction of the conference, the lecture or the speaker, or longer descriptions of the content. We provide examples of transcripts and summaries in Appendix A.

## 4.1 Modalities

In the video files, there are two modalities: audio and visual. The audio is mostly comprised of speech and the possible music or noise, the visual can be for example the shared screen or a record of the speaker. Figure 1 pictures examples of frames from the visual modality covering most of the cases in the dataset. There is a third modality in the target summary and the metadata which is the textual modality.

Additional features are created via automatic processing of the video. From the audio, we generate a textual transcript, and from the video stream, we select a sparse set of key frames.

## 4.2 Processing the videos

From the raw video we extract automatically the transcript using an automatic speech recognition model, Whisper [26] (specifically the `small` multilingual checkpoint which offers a good accuracy/speed tradeoff for this task). The resulting textual modality will allow us to benchmark common Natural Language Processing abstractive summarization methods.

Because the videos are records of presentations, the visual information is concentrated in a few still frames such as the slides from the slideshow. We provide a noisy estimation of the slides constituting the visual support of the presentation by extracting key frames with an heuristic using semantic hashes of the video frames. In examples (a), (c), (d), (f) and (g) of Figure 1, the sparse information assumption of the video stream stays relevant but does not hold in the examples (b) or (e). Arguably, a content-based key frame extraction method might be more suited for the latter examples, however these visual layouts are not the most usual nor the most informative visual information for the presentations.

More specifically, a semantic hash is a binary code that represents a document such that codes of data points with similar features are close by the Hamming distance. To encode the frames, we use a Discrete Cosine Transform-based semantic hash function described in Algorithm 1, and inspired by a perceptual video hash function introduced in [8]. The Discrete Cosine Transform of type II (DCT) is following the implementation in the `scipy` library [31].

**Data:** $x$ #a grey-scale 64*64 image
**Result:** $h$ #the 64 boolean long-semantic hash binary code for image $x$
**begin**
  $x \leftarrow DCT_1(DCT_0(x))$ #Apply Discrete Cosine Transform on both axes
  $m \leftarrow median(x)$
  **for** $j \leftarrow 0; j < 8; j \leftarrow j + 1$ **do**
    **for** $i \leftarrow 0; i < 8; i \leftarrow i + 1$ **do**
      $h_{i+8j} \leftarrow \begin{cases} 0 \text{ if } x_{i,j} \leq m \\ 1 \text{ else} \end{cases}$
    **end**
  **end**
**end**

**Algorithm 1:** DCT-based semantic hash function

Because slide change usually does not occur multiple times per second, we are downsampling the videostream, in order to keep around 2 frames per second. If there is a camera showing the speaker, it is likely to be shown in a corner or a side of the image, so we are splitting every frame spatially in 9 patches to track changes in the center, sides and corners of the image. Every patch is encoded using the semantic hash function described in Algorithm 1. We compute Hamming distances from the perceptual hash of every patch to the perceptual hash of the same patch at the previous frame. We get 9 values for each frame, and ignore the patches that change too much, since it's likely to be where the camera is located in the shared screen. The threshold for the minimal distance that will be considered as a slide change is computed for every video with hand-made rules fine-tuned on a few examples. The resulting

heuristic is very fast to compute and allows to keep a few dozens still frames from a full video stream, that include most of the slides in the slide show.

## 4.3 Data Statistics

*4.3.1 Data metadata distribution.* Most records come with a set of metadata, such as the year of release, the genre and subject. In order to provide insights about the content of the dataset, we look into the distributions of these metadata.

Figure 2a is a histogram of the release years of the records, that range from 1952 to 2022, with most of the data evenly distributed from 2013 to 2022. In Figure 2b, we plot the duration in second of the records, the average being 2245 seconds, *ie* 37.4 minutes. Figure 2c shows how many key frames were extracted from each record, the average is 45.9 key frames per presentation. Figure 2d is a histogram of the academic subjects of the presentations that shows that most of the records are related to Computer Science. Figure 2e is a histogram of the presentation genres, most of the documents are either conferences, academic talks or lectures.

The metadata shows that a typical record from the dataset is an academic conference or a lecture about Computer Science released between 2013 and 2022, it lasts 37.4 minutes and contains 45.9 slides.

*4.3.2 Tokens statistics.* Following [12], we compute some statistics on the dataset, such as number of documents, extractive fragment coverage, extractive fragment density, compression ratios, and number of tokens in the source and summary texts. The statistics are computed following the algorithms from [12], but using a sub-word Byte-Pair Encoding tokenization from [2], instead of a word tokenization.

From the tokenized source text $x = (x_i)_{i=1}^{N}$ and the tokenized summary text $y = (y_i)_{i=1}^{M}$, [12] defines a set $\mathcal{F}(x, y)$ called the set of extractive fragments, as the set of the largest $n$-grams of $y$ that are in $x$.

Using $\mathcal{F}(x, y)$, we compute the extractive fragment coverage and the extractive fragment density.

The coverage is defined as the sum of the lengths of the elements in $\mathcal{F}(x, y)$ divided by the length of $y$,

$$Coverage(x, y) = \frac{1}{|y|} \sum_{f \in \mathcal{F}(x,y)} |f|. \tag{1}$$

This value is the percentage of words in the tokens that are also in the source. A lower value means that the summary is using novel tokens. This value only accounts for the vocabulary extraction, but does not take into account the order of the tokens.

The density is defined as the average length of the extractive fragments of $\mathcal{F}(x, y)$,

$$Density(x, y) = \frac{1}{|y|} \sum_{f \in \mathcal{F}(x,y)} |f|^2. \tag{2}$$

The compression ratio is the token ratio between the source and the summary texts,

$$Compression(x, y) = \frac{|x|}{|y|}. \tag{3}$$

The statistics computed on TIB and on other summarization datasets are reported in Table 1.

# 5 EXPERIMENTS AND BASELINES

We provide experiments and baselines for abstractive summarization using only the text modality derived from the audio modality, via the automatically extracted transcript. The goal is to generate a text that summarizes the document, with the abstract being the reference "gold" summary.

The visual and speech modality were not used in the models tested in this work, as to the best of our knowledge, no existing model is designed to do abstractive textual summarization of videos this long.

## 5.1 Evaluation metrics

We are using the ROUGE metric [22] and the BERT Score [34] for evaluation, via the F-measure of ROUGE-$n$ (R$n_{f1}$), longest common subsequence-based ROUGE (RL$_{f1}$) and BERT Score (BS$_{f1}$).

ROUGE score computes the overlap of $n$-grams between the candidate and the reference summary. BERT Score computes pairwise cosine similarity between BERT embeddings of the reference and candidate summaries. ROUGE is a commonly used metrics for summarization evaluation. BERT Score has been shown to correlate well with human judgment for summarization tasks.

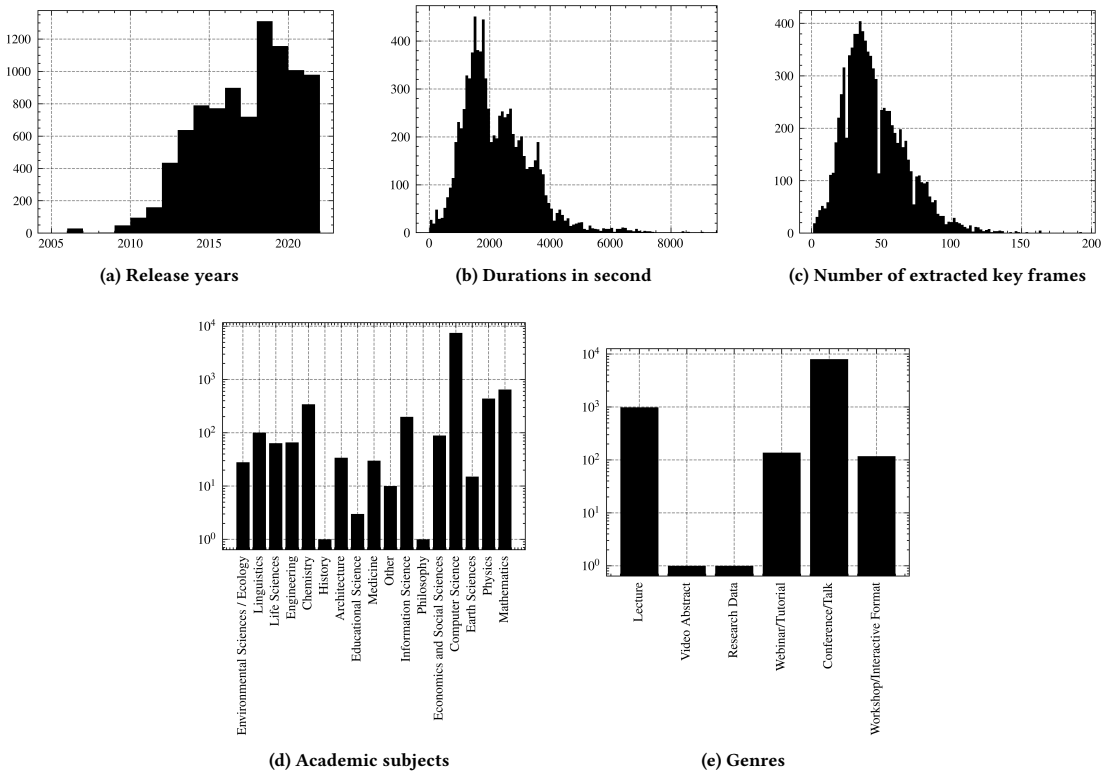
We are using the metrics implementations provided by the `evaluate` library[5], with default settings.

## 5.2 Models and heuristics

The models that are being benchmarked are text-only models for abstractive or extractive summarization. Most of them have an input size limit in term of tokens, in those cases the input is truncated to the size limit. For all the deep learning models, we are using the HuggingFace transformers [32] and Pytorch [24] implementations.

- Lead-3 returns the first 3 sentences of the transcript as a summary. It is a common extractive baseline for text summarization tasks.
- Extractive Oracle is an upper-ceiling limit for extractive summarization approaches, as it selects the sentences of the transcript that maximise the ROUGE score with respect to the target abstractive summary. We are using the implementation from the `extoracle_summarization` library[6].
- BART [19] is a Transformer encoder-decoder architecture designed for sequence-to-sequence Natural Language Generation tasks such as abstractive summarization, it can handle inputs of up to 1,024 tokens. We are using the `bart-base` checkpoint that has 140M parameters, finetuned on either the ArXiv dataset or the CNN/Daily Mail dataset.
- PEGASUS [33] is an abstractive summarization model trained with self-supervised learning, it can handle inputs of up to 512 tokens. We are using the 570M parameters `pegasus` checkpoint, finetuned on either the ArXiv dataset or the CNN/Daily Mail dataset.
- PEGASUS-X [25] is an extension of the PEGASUS model to handle inputs of up to 16,384 tokens by using staggered block-local attention. We are initializing the model from the 272M parameters `pegasus-x-base` checkpoint, and then fine-tune on the TIB train split.

[5] https://github.com/huggingface/evaluate
[6] https://github.com/pltrdy/extoracle_summarization




(a) Release years (b) Durations in second (c) Number of extracted key frames

(d) Academic subjects (e) Genres

**Figure 2: Histograms of metadata**

**Table 1: Statistics of abstractive summarization datasets.**

| Dataset | Available modalities[1] | #Docs | Coverage | Density | Comp. Ratio | #Tokens | |
|---|---|---|---|---|---|---|---|
| | | | | | | Source | Summary |
| CNN/Daily Mail [23] | T | 311,971 | 0.78 | 11.27 | 13.93 | 868.63 | 68.19 |
| Billsum [17] | T | 22,218 | 0.86 | 2.76 | 24.59 | 3668.43 | 227.35 |
| BIGPATENT [29] | T | 1,341,362 | 0.86 | 2.31 | 35.54 | 3950.17 | 128.47 |
| ArXiv [5] | T | 215,913 | 0.91 | 3.14 | 45.76 | 8967.91 | 371.17 |
| PubMed [5] | T | 133,215 | 0.88 | 7.31 | 16.19 | 4039.90 | 272.18 |
| BookSum Chapter [18] | T | 12,515 | 0.76 | 1.87 | 16.38 | 5966.13 | 463.03 |
| How2 2000h [27] | A+$T^2$ | 72,983 | 0.58 | 0.97 | 8.37 | 305.21 | 38.91 |
| How2 300h [27] | A+V+$T^2$ | 11,458 | 0.60 | 1.21 | 9.03 | 333.43 | 39.40 |
| **TIB (ours)** | A+V+$I^3$ +$T^2$ | 9,103 | 0.71 | 1.42 | 63.49 | 6309.96 | 184.29 |

[1]A: Audio, V: Video, I: Image, T: Text
[2]Automatically transcribed from audio
[3]Automatically extracted from video

- Longformer Encoder-Decoder [2] follows BART's architecture, with a chunked self-attention mechanism designed to handle longer inputs, up to 16,384 tokens. We are using the `led-base-16384` checkpoint, that is initialized from `bart-base` and has 161M parameters, the model is fine-tuned on the TIB train split.

The PEGASUS-X and Longformer Encoder-Decoder models are fine-tuned on the TIB dataset train split for 10,000 steps and validated on the validation split, that represents respectively 80% and

**Table 2: Performance of baseline text-only models on the TIB dataset**

| Model | Type | #Parameters | $R1_{f1} \uparrow$ | $R2_{f1} \uparrow$ | $RL_{f1} \uparrow$ | $BS_{f1} \uparrow$ |
|---|---|---|---|---|---|---|
| Lead-3 | Heuristic | 0 | 14.10 | 2.09 | 9.05 | 0.81 |
| Extractive Oracle | Heuristic | 0 | 40.29 | 11.14 | 19.83 | 0.83 |
| BART [19] CNN/Daily Mail zero shot | Abstractive | 140M | 19.45 | 3.34 | 12.08 | 0.83 |
| BART [19] ArXiv zero shot | Abstractive | 140M | 22.83 | 4.74 | 13.75 | 0.83 |
| PEGASUS [33] CNN/Daily Mail zero shot | Abstractive | 570M | 16.24 | 3.41 | 11.04 | 0.83 |
| PEGASUS [33] ArXiv zero shot | Abstractive | 570M | 21.55 | 3.04 | 12.49 | 0.82 |
| PEGASUS-X [25] fine-tuned | Abstractive | 272M | 20.62 | 5.71 | 14.14 | 0.84 |
| Longformer Encoder-Decoder [2] fine-tuned | Abstractive | 161M | 27.75 | 7.39 | 17.36 | 0.85 |

10% of the dataset. All the models are tested on the TIB dataset test split, that is 10% of the dataset. Model output were decoded using beam search with 2 beams and repeated 3-gram blocking. Results of the experiments are reported in Table 2.

### 5.3 Evaluation

The Lead-3 heuristic offers low performance, which is not suprising since the first sentences are likely to contain greetings and presentation of the speaker that will not appear in the target summary.

The extractive oracle heuristic is the upper limit of extractive approaches. Its medium performance is a consequence of the low coverage and density statistics of the TIB dataset, as reported in Table 1.

The CNN/Daily Mail and ArXiv datasets are both abstractive summarization datasets. The latter is more similar to the TIB dataset in both its statistics such as density, source and summary lengths, as shown in Table 1, and in its content which is scientific articles, while the former is newspaper articles. The models we tested on a zero shot setting were trained on these datasets, and as expected, the models trained on ArXiv perform better on the TIB test set than the same models trained on CNN/Daily Mail.

We fine-tuned two models designed for long document abstractive summarization on the TIB dataset: PEGASUS-X [25] and Longformer Encoder-Decoder [2]. Both of these models are transformer models that use both local attention (staggered block-local attention for PEGASUS-X and sliding-window attention for Longformer) and global attention. The main difference between Longformer and PEGASUS-X is the use of learned absolute position embeddings for the former, and sinusoidal position embeddings for the latter. In a videoconference record, the position of a token is informative of the part of the presentation and the ability to learn the position of the most relevant parts might explain why Longformer outperforms PEGASUS-X in this setting.

Examples of decoded outputs using the fine-tuned models are provided in Appendix A along with extracts from the input transcripts and the reference summaries.

## 6 CONCLUSION AND FUTURE WORK

In this work, we introduce TIB, a dataset for abstractive summarization of long multimodal documents. Compared to other available summarization datasets, the length of documents in TIB is much longer than multimodal datasets, and comparable to long-range textual datasets. We tested existing textual summarization models and state-of-the-art models for long range text summarization, as a baseline for future work, such as multimodal methods. We hope this dataset will contribute to the study of automatic videoconference summarization by giving access to multimodal data for abstractive summarization of long documents.

The methods we have benchmarked are textual-only baselines. However, slides usually provide information that is not always redundant in the speech, as well as relational inductive biases that help to understand the structure of a presentation. Future work on this dataset includes studying summarization models that take the whole multimodal input to predict the abstract.

## REFERENCES


[1] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, and Ioannis Patras. 2021. Video Summarization Using Deep Neural Networks: A Survey. *Proc. IEEE* 109, 11 (Nov. 2021), 1838–1863. https://doi.org/10.1109/JPROC.2021.3117472 Conference Name: Proceedings of the IEEE.

[2] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. http://arxiv.org/abs/2004.05150 arXiv:2004.05150 [cs] version: 2.

[3] Xavier Bost, Serigne Gueye, Vincent Labatut, Martha Larson, Georges Linarès, Damien Malinas, and Raphaël Roth. 2019. Remembering winter was coming. *Multimedia Tools and Applications* 78, 24 (Dec. 2019), 35373–35399. https://doi.org/10.1007/s11042-019-07969-4

[4] Francesca Bottanelli, Bruno Cadot, Felix Campelo, Scott Curran, Patricia M. Davidson, Gautam Dey, Ishier Raote, Anne Straube, and Matthew P. Swaffer. 2020. Science during lockdown – from virtual seminars to sustainable online communities. *Journal of Cell Science* 133, 15 (Aug. 2020). https://doi.org/10.1242/jcs.249607

[5] 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).* Association for Computational Linguistics, New Orleans, Louisiana, 615–621. https://doi.org/10.18653/v1/N18-2097

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[7] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Brussels, Belgium, 2475–2485. https://doi.org/10.18653/v1/D18-1269

[8] B. Coskun and B. Sankur. 2004. Robust video hash extraction. In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference, 2004.* 292–295. https://doi.org/10.1109/SIU.2004.1338317

[9] Dialpad. 2022. *State of Video Conferencing 2022.* Technical Report.

[10] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165 (March 2021), 113679. https://doi.org/10.1016/j.eswa.2020.113679

[11] José García, M. Sabatino, Pasquale Lisena, and Raphaël Troncy. 2014. Detecting hot spots in web videos. *CEUR Workshop Proceedings* 1272 (Oct. 2014), 141–144.

[12] 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 708–719. https://doi.org/10.18653/v1/N18-1065

[13] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating Summaries from User Videos. In *Computer Vision – ECCV 2014 (Lecture Notes in Computer Science)*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 505–520. https://doi.org/10.1007/978-3-319-10584-0_33

[14] 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*. Association for Computational Linguistics, Online, 1419–1436. https://doi.org/10.18653/v1/2021.naacl-main.112

[15] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The Multilingual Amazon Reviews Corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4563–4568. https://doi.org/10.18653/v1/2020.emnlp-main.369

[16] Kyong-Jee Kim, Seo Rin Kim, Jangwook Lee, Ju-Young Moon, Sang-Ho Lee, and Sung Joon Shin. 2022. Virtual conference participant's perceptions of its effectiveness and future projections. *BMC Medical Education* 22, 1 (Jan. 2022), 10. https://doi.org/10.1186/s12909-021-03040-9

[17] Anastassia Kornilova and Vlad Eidelman. 2019. BillSum: A Corpus for Automatic Summarization of US Legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. 48–56. https://doi.org/10.18653/v1/D19-5406 arXiv:1910.00523 [cs].

[18] Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BookSum: A Collection of Datasets for Long-form Narrative Summarization. http://arxiv.org/abs/2105.08209 arXiv:2105.08209 [cs].

[19] 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*. Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[20] Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2019. Read, Watch, Listen, and Summarize: Multi-Modal Summarization for Asynchronous Text, Image, Audio and Video. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (May 2019), 996–1009. https://doi.org/10.1109/TKDE.2018.2848260 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[21] Mingzhe Li, Xiuying Chen, Shen Gao, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2020. VMSMO: Learning to Generate Multimodal Summary for Video-based News Articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9360–9369. https://doi.org/10.18653/v1/2020.emnlp-main.752

[22] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[23] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Berlin, Germany, 280–290. https://doi.org/10.18653/v1/K16-1028

[24] 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 *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[25] Jason Phang, Yao Zhao, and Peter J. Liu. 2022. Investigating Efficiently Extending Transformers for Long Input Summarization. http://arxiv.org/abs/2208.04347 arXiv:2208.04347 [cs].

[26] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. https://doi.org/10.48550/arXiv.2212.04356 arXiv:2212.04356 [cs, eess].

[27] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A Large-scale Dataset for Multimodal Language Understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. https://hal.science/hal-02431947

[28] Gunnar Schwarz, Davide Bleiner, and Detlef Günther. 2022. On video lectures during remote teaching and beyond. *Analytical and Bioanalytical Chemistry* 414, 11 (May 2022), 3301–3309. https://doi.org/10.1007/s00216-022-03983-y

[29] Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2204–2213. https://doi.org/10.18653/v1/P19-1212

[30] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. TVSum: Summarizing web videos using titles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5179–5187. https://doi.org/10.1109/CVPR.2015.7299154 ISSN: 1063-6919.

[31] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17, 3 (March 2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2 Number: 3 Publisher: Nature Publishing Group.

[32] 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. HuggingFace's Transformers: State-of-the-art Natural Language Processing. http://arxiv.org/abs/1910.03771 arXiv:1910.03771 [cs].

[33] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, 11328–11339.

[34] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. https://iclr.cc/virtual_2020/poster_SkeHuCVFDr.html

# A EXAMPLES OF DECODED SUMMARIES

Received 12 April 2023; revised 6 July 2023

**Table 3: Example of decoded summaries using the models fine-tuned on TIB - document 0 of the TIB test set (random document)**

| **Transcript** |
| --- |
| Yes, thank you for coming. I know it's late and you're all tired. So my name is Martin. I work for Amazon for a prime video and I want to talk a little about the profiler that we've done at Amazon. So before I start talking about my profiler, just do a quick introduction to profilers in general. Probably some of you never used one, so I just do a quick introduction. Then I explain why we decided to do another profiler even though there is a bunch of profilers already. Then I talk a little about the Hocktracer features, do a demo and hopefully we'll have time for questions at the end. So a profiler basically is a tool that allows you to measure performance of the application. You can basically split existing profilers into two groups. It's a sample-based profiler that runs periodically and checks some information from your application. So the simple profiler, simple sample-based profiler can be like that. You have a while loop and you check, for example, the call stack of a specific process, based on process ID. Later on, you can gather those data and generate some statistics `[...]` |
| **Reference summary** |
| HawkTracer is low-overhead instrumentation-based profiler built at Amazon Video for platforms with limited capabilities. It's highly extensible (at compile time) and portable so it can be run on almost any embedded device. In this talk I'll briefly talk about history of the profiler (why did I start it, and how did I manage to publish it to open source world), introduce the architecture of the profiler, present it's advantages and limitations, show how can you instrument the code and demonstrate the profiler in action by running it with an example project. |
| **Predicted summary: Longformer Encoder-Decoder** |
| Hocktracer is an instrumentation-based profiling tool that can be used for measuring memory usage and CPU usage of an embedded system. In this talk, I will show you how we created a custom implementation of HockTracer and how we instrumented the data in it. I will also explain why we decided to do another one. Hock Tracer is a profiler for user-space profiling on embedded systems. It can be implemented as a standalone library or as a full-fledged Hocktrace module. In addition to the built-in Hock-Tracer library, it can be extended to run in any C++, Python or Rust language. |
| **Predicted summary: PEGASUS-X** |
| The Hock Tracer is an instrumentation-based profiler that allows developers to measure performance of their application. It is a tool that allows you to split existing profilers into two groups. It runs periodically and checks some information from your application. The tool can be split into 2 groups. The simple profiler can be like that. You have a while loop and you check, for example, the call stack of a specific process, based on process ID. Later on, you can gather those data and generate some statistics. The accuracy depends on the sampling frequency. It's a good method to find why your application is in general slow. But if you want to have very detailed data, it's good to do an instrumentation based profiler. It can be used to measure the time we spent in the full function by saving the timer before calling the function and after calling the functions. The Hocktracer features are not very accurate because the accuracy depends to the sampling frequencies. It needs to be embedded to the application itself. |

**Table 4: Example of decoded summaries using the models fine-tuned on TIB - document 361 of the TIB test set (worst $RL_{f1}$ score for Longformer Encoder-Decoder)**

| **Transcript** |
|---|
| Yeah, I'm Andrew Harvey. I'm here to talk about a side project that I worked on just kind of on the weekends that I was interested in. It's called Beyond Tracks. It's a website, beyondtracks.com. In the next like two weeks or so, we're going to launch a new version of it. It's just been so busy with this conference. I haven't had a chance to do that yet. But, so I joined OpenStream app in 2010. And unlike many other maps, it actually showed a lot of outdoor walking tracks. I looked at other maps and either they were a topo map from 20 years ago, or it didn't have the detail, or it was another map that just didn't have the content there. But OpenStream app had a lot of these trails, which I thought was pretty cool. And the other thing is, I could actually go and edit it. So if I noticed that stuff was missing, I could go and edit that and fix it up, add in new tracks. So that was also really good. And you can download all the data. So you can use it for other purposes, apart from just kind of looking at the map that's on OpenStreetMap.org. So I kind of had all those criteria met. And I was really interested in how can I take OpenStreetMap data and the information that many people contribute to it. So not just where paths are, but what the surface of that is like and all kinds of other information about that that isn't really exposed on this map. So how can I make that more useful? So I thought, OK, I'll take, instead of using the OpenStreetMap map, I'll take the data and start showing it in a different way. So showing information about how long walks are, what the climb is, photos [. . . ] |
| **Reference summary** |
| BeyondTracks: an OpenStreetMap case study |
| **Predicted summary: Longformer Encoder-Decoder** |
| Andrew's talk was the second talk in the "Beyond Tiles" session at FOSS4G SotM Oceania 2019, organised by OSGeo Oceano and held at The National Library in Wellington, New Zealand from November 12-15 2019. FOSS3G SOTM OCEania is the coming together of Oceo and Oceana's geospatial open source and open data community - with four days of workshops, presentations, a community sprint and social events. |
| **Predicted summary: PEGASUS-X** |
| BeyondTracks is an open source project that uses the OpenStreetMap data to create a website that allows users to search for and access walks on top of OpenStreetMap data. |

**Table 5: Example of decoded summaries using the models fine-tuned on TIB - document 617 of the TIB test set (best $RL_{f1}$ score for Longformer Encoder-Decoder)**

| **Transcript** |
|---|
| Classes in Polymer Dynamic. Based on George Philly's book, Phenomenology of Polymer Solution Dynamics, Cambridge University Press, 2011. And today, this lecture is lecture 9, More Dielectric Relaxation. I'm Professor Philly's, and this is part 2 of my classes on dielectric increment in polymer solutions, based on my book, Phenomenology of Polymer Dynamics. Okay, so what we are going to do today is to continue with our reading of chapter 7. I'll have homework again next time. The issue is as follows, dielectric spectroscopy is an enormously effective tool that gives us a wide variety of different measurements about a polymer solution. In particular, if we have a polymer chain in the solution, it has a few features. It has an end-to-end vector, that is, a vector that starts at one end of the chain and points straight to the other end. And that's a measurement of how big the polymer is in the solution. It's not the only measure, but it is a measure. Furthermore, we can use dielectric relaxation spectroscopy to characterize the relaxation of this vector. That is, [. . . ] |
| **Reference summary** |
| Lecture 9 - dielectric relaxation, part 2. George Phillies lectures on polymer dynamics, based on his book "Phenomenology of Polymer Solution Dynamics". |
| **Predicted summary: Longformer Encoder-Decoder** |
| Lecture 9 - More Dielectric Relaxation. George Phillies lectures on polymer dynamics based on his book "Phenomenology of Polymer Solution Dynamics". |
| **Predicted summary: PEGASUS-X** |
| Lecture 9 - More Dielectric Relaxation. George Phillies lectures on dielectric increment in polymer solutions. |

**Table 6: Example of decoded summaries using the models fine-tuned on TIB - document 275 of the TIB test set (worst $RL_{f1}$ score for PEGASUS-X)**

| **Transcript** |
|---|
| Well, then we have like six minutes. Speak faster. Your role will be to push buttons. Well, 2005, there was an idea. On the way back from FOSSTEM, a couple of guys said, well, FOSSTEM is nice, but the buildings are like a little bit old, crappy, and the rooms are overheated. And the other problem with FOSSTEM is it's in the winter. I don't know if you've been there. It's freezing cold in Brussels. So we decided, well, we have this new modern university here we could use. And let's do it in summer so we can do a barbecue afterwards. This is a touchscreen, by the way. Now you know. Don't touch it. But even back then, Frostconn had the features we are used to, like, for example, a conference. And even the three wise men showed up. And a bouncy castle. And over the years, we had a selection of bouncy castles. Then there was a merchandise booth even back then. And you can buy this year's t-shirts, by the way, or add the merchandise booth. There was a social event. And we had a, like, I think five years ago, we had a show and a glowing frog. And there was barbecue. By the way, do you recognize this guy? We all were much younger back then. We had lots of volunteers over the years. One guy even showed up from Ireland. David Dolphin. He now works somewhere in the valley, I believe, Flickr or something, `[...]` |
| **Reference summary** |
| Short introduction and last minute changes are announced here |
| **Predicted summary: Longformer Encoder-Decoder** |
| Frostconn is a conference that started in 2005 in Brussels. It has grown to become a regular event and a growing international conference. In this talk, the dean of the department of computer science, Mr. Hayden, will give the tenth edition of the conference. |
| **Predicted summary: PEGASUS-X** |
| The 10th Frostconn will take place in Brussels from November 12-15 2019. |

**Table 7: Example of decoded summaries using the models fine-tuned on TIB - document 541 of the TIB test set (best $RL_{f1}$ score for PEGASUS-X)**

| **Transcript** |
|---|
| Alright. And we are live. Alright. We are here in another QA session with Jake LaBelle. Talking about ZOS and surrogate chains. Jake, could you please talk a little bit about the QA session. And how you got into mainframe hacking. Hi, I'm Jake. So I guess how I got into it. My company does some jobs in it. I was looking at some of the reports they wrote and I went, Alex pretty cool. Looked at some pictures of mainframes and was like, yeah, that looks like my type of thing. And yeah, just jumped right into it. It's, yeah, it's. I know I think it looks cool and so just went for it. That's awesome. And I believe this is your first time presenting at Defcon, correct. Yeah, first time. Yeah. So we have a tradition here at Defcon. Whenever you do your first talk, you can do a lot of stuff. Defcon, whenever you do your first talk, we welcome you on to the big stage with a drink. This is for contributing content back to the community and answer taking the time to answer questions. So, cheers to you, Jake. Welcome to Defcon. I'm going to be telling us we're filling up my drink though, so. All right. Cheers. Yeah. So we've already got a few questions that have been going through the chat. You kind `[...]` |
| **Reference summary** |
| Question and answer session for Jake Labelle - Getting Shells on zOS with Surrogat Chains |
| **Predicted summary: Longformer Encoder-Decoder** |
| Question and Answer session for Jake LaBelle - Mainframe Hacking is Deal Long Live Mainframe Security. During this Q<br>A session you will be able to answer questions about ZOS, surrogate chains, mainframe security, and other topics relevant to you. |
| **Predicted summary: PEGASUS-X** |
| Question and Answer session for Jake LaBelle - ZOS and surrogate chains |