FINE GRAINED SPOKEN DOCUMENT SUMMARIZATION THROUGH TEXT SEGMENTATION

Samantha Kotey[1]*, Rozenn Dahyot*[2] *& Naomi Harte*[1]

[1]ADAPT Centre, School of Engineering, Trinity College Dublin, Ireland
[2]ADAPT Centre, Department of Computer Science, Maynooth University, Ireland

2022 IEEE Spoken Language Technology Workshop (SLT) | 979-8-3503-9690-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/SLT54892.2023.10022829

ABSTRACT

Podcast transcripts are long spoken documents of conversational dialogue. Challenging to summarize, podcasts cover a diverse range of topics, vary in length, and have uniquely different linguistic styles. Previous studies in podcast summarization have generated short, concise dialogue summaries. In contrast, we propose a method to generate long fine-grained summaries, which describe details of sub-topic narratives. Leveraging a readability formula, we curate a data subset to train a long sequence transformer for abstractive summarization. Through text segmentation, we filter the evaluation data and exclude specific segments of text. We apply the model to segmented data, producing different types of fine grained summaries. We show that appropriate filtering creates comparable results on ROUGE and serves as an alternative method to truncation. Experiments show our model outperforms previous studies on the Spotify podcast dataset when tasked with generating longer sequences of text.

***Index Terms*—** spoken document summarization, text segmentation, long sequence transformers, readability formulas, podcast summarization

1. INTRODUCTION

Spoken documents are transcripts of dialogue from sources such as podcasts, interviews, and meetings that are generated by automatic speech recognition systems. Considerable attention has been paid recently to podcast summarization due to the release of the Spotify podcasts dataset [1]. The objective of summarization is to condense these transcripts into shorter text sequences that depict the original conversation. There is indeed a growing need to develop automated tools for dialogue summarization and information access [2]. Podcast dialogue can be hours long [2], and podcast summaries can help listeners to choose which podcast to listen to.

There are two main issues for training a system for podcast summarization: the quality of target (output) summaries and the length of the source (input) transcripts. Summaries in the Spotify podcast dataset are written by creators themselves. Episode summaries are problematic as they contain noisy promotional information and do not conform to a universal language style. Researchers have approached the problem by removing irrelevant material through classifier models [3], while others utilize preprocessing to exclude social media language and repetitive advertising phrases [4].

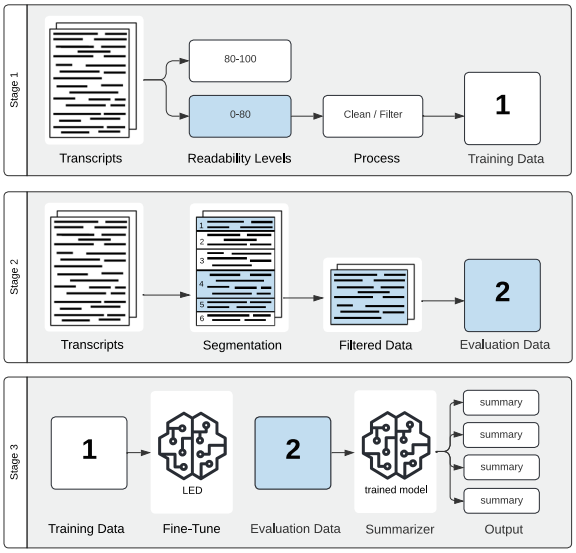

Fig. 1. Overview of our approach to generate different summary types through text segmentation.

To address the length issue, BART transformer models [5] have been adopted to generate short, abstractive meaningful summaries. BART models are limited by an input length of 1024 tokens, and truncate potentially important knowledge after the maximum is exceeded. Adapting to this constraint, sentence selection methods have been proposed such as topic modeling [6], sentence ranking [7], genre labeling [8] and hierarchical frameworks [4]. Models have also been proposed [9] experimenting with longformers, which have an extended input limit. These models have performed reasonably well on short text summarization, however neglected to focus on generating longer sequences of text. Conversations in podcasts can contain multiple story lines, which are difficult to capture in a short summary. Long summaries provide users with an opportunity to consume more interesting content, which may otherwise be missed. Figure 1 shows an overview of our

978-1-6654-7189-3/22/$31.00 ©2023 European Union

proposed approach to generate long fine-grained summaries, which are fluent, coherent and describe sub-topic narratives. In this paper we describe two key contributions:

1. We introduce a method to construct a subset of training data, by measuring how challenging transcripts are to read. Through readability score metrics, we analyze the relationship between lexical qualities and description length. Leveraging a Longformer-Encoder-Decoder (LED) [10], we fine-tune a sequence-to-sequence model with a curated subset of our data.

2. We preprocess a subset of evaluation data using a text segmentation method, and apply the trained model to the filtered data. We show that excluding specific segments of text, directs the flow of information to the model and serves as an effective data reduction technique. We demonstrate that text segmentation is an alternative method to truncation, and evaluate our method with ablation studies.

index	sentence	topic
10	I welcome back Derrick Hayes host and creator of the Monsters Among Us podcast as he describes a Cryptid he witnessed as a child that set him on his paranormal Journey.	1
11	... I've been collecting ghost stories for years and marks is truly one of my favorites.	1
12	You definitely don't want to miss it chapter one.	2
13	Alien.	2
14	Big cat when Derek was around 10 years old.	2
15	He is eight-year-old brother and a friend were exploring his dad's property in Southeast, Ohio.	2
54	I'm sure you've heard of Bigfoot the Loch Ness monster Moth Man.	2
55	... alien big cat as a legitimate species scientists are discovering new species every single year 65 of this Earth is still unexplored.	2
56	So to say that we've seen it all and know everything is just wrong.	3
57	And personally I want to know what else were wrong about if you're hungry for more Cryptid encounter.	3
58	Has make sure you check out Derek's podcast Monsters Among Us where you can call in your own unexplainable experiences chapter to the hag.	3

Table 1. Example of text segmentation applied to a transcript.

2. RELATED WORK

Spoken documents are similar to long text documents, with the added complexity of disorganized conversational narratives. Variations of transformer models have proven to be efficient in long document summarization tasks [11], when a combination of multiple attention patterns are employed. Zaheer et al. [12] proposed a sparse attention mechanism by combining random, window and global attention. Similarly, Beltagy et al. [10] combined sliding, dilated and global windows to develop a Longformer-Encoder-Decoder model. For our experiments we adopt an LED framework, as its scalable properties allow for lengthy podcast transcripts to be examined in context. Our goal is to produce long summaries with lengths greater than 100 tokens. Previously, researchers in different domains, have considered tokens ranging from 50-200 to be long, and summary lengths of up to 1000 tokens to be extra long [13]. These extreme lengths are suitable for applications where long content is required such as reports.

2.1. Readability of Spoken Documents

Prior research has studied how media types such as books, TV shows [14], lecture videos [15] and campaign speeches [16] target specific audiences by their receptive language skills. Reddy et al. [17] analyzed lexical properties in transcripts and descriptions in the Spotify dataset. The work explored the relationship between podcasters' linguistic styles and listener engagement levels. It was shown that transcripts of podcasts with a high engagement level had a diverse vocabulary, longer sentences, and a higher Dale-Chall reading grade. In terms of schooling, readability tests such as Dale-Chall [18] and the Flesch–Kincaid grade level [19] determine how difficult text is to read. A higher grade level denotes text is better understood by a university graduate. In contrast, the Flesch Reading Ease allocates scores from 0 (text difficult to read) to 100 (text easy to read). The formula calculates how long a sentence is and the average number of syllables per word [20]. Following Reddy et al. [17], we analyze the difference between transcripts with longer and shorter descriptions (c.f. paragraph 3.1).

2.2. Spoken Document Segmentation

The task of segmenting long documents to define the beginning and end of topic narratives, has been approached by several researchers. Pethe et al. [21] proposed predicting chapter boundaries in long novels by training a supervised BERT-based model on a repository of scanned books. Xiao and Carenini [22] experimented with scientific papers, exploring the idea of guiding extractive summarization by leveraging segmented texts. Popular topic segmentation algorithms include, TextTiling [23], TopicTiling [24], Content Vector Segmentation [25] and TextSplit [26]. In the context of spoken documents, Chen and Yang [27] segmented daily conversations into four global topics with a C99 algorithm developed by Choi [28]. Following this approach, we propose to process the evaluation data with the TextSplit [26] algorithm, as shown in Table 1 and detailed in Section 3.

3. PROPOSED METHODS

Our proposed pipeline approach involves three stages, as illustrated in Figure 1. In **stage 1**, we curate a selection of podcast transcripts from the Spotify podcasts dataset [1], to form the training data. In **stage 2**, we process the evaluation data, by segmenting long text transcripts into smaller segment

chunks. Each episode is divided and numbered into segments, and each segment is defined by a change in topic conversation. We select various segment chunks to form a filtered down version of each transcript, with a reduced input size. Finally, in **stage 3**, we fine-tune a long sequence transformer on the curated dataset and apply the model to the filtered evaluation data. These stages are described in the following sections.

3.1. Training Data Curation (Stage 1)

There are 105,360 podcast episodes in the Spotify podcasts dataset [1], from a cross-section of genres and subject matter. Created both professionally and at amateur level, the dataset includes a corpus of over 60,000 hours of speech transcribed through ASR. However, the dataset is noisy and not all of the episodes are suitable for training. We filter and process the data to obtain transcripts which are difficult to read (analyzed through readability metrics), which also have long clean creator descriptions. This requires removing promotional material, to improve the quality of text.

Although the dataset is in English, a number of non-English descriptions were present, probably due to mislabeling by creators. These outliers were removed using a Naive Bayes model for language detection[1].

Readability scores were computed using the Textstat[2] python library and two groups of data were created (0-80 and 80-100). Transcripts with Flesch scores less than 80 on average have longer descriptions with richer vocabulary, and these are the ones kept for training (cf. Fig. 1). The threshold was chosen because scores over 80 are easier to read and indicative of conversational English. In this group (80-100), the descriptions on average were shorter and contained more noisy irrelevant information. As the objective is to produce long quality training data, this subset was not retained.

To improve the quality of creator descriptions in the group (0-80), we performed a rigorous cleaning process by curating a custom stop word list of uninformative common creator expressions. Phrases such as '*support this podcast*', '*please subscribe*' and '*comment down below*' are frequently repeated in the transcript corpus. Sentences that began with, or contained these stop words were automatically removed, along with hashtags, social media links, URLs and residual emoji's. We tokenize clean descriptions using Python's 'Split' method, separating words by blank spaces, and then count the number of words in each description. The data was filtered as shown in Table 2, with a minimum word count of 60 required for clean creator descriptions, to ensure a large enough sample size.

3.2. Segmentation of Evaluation Data (Stage 2)

We apply the TextSplit library, developed by Christoph Schock [26] to the evaluation data, using a BERT based SentenceTransformer [29] to represent sentences as vectors instead of Word2Vec [30]. The framework uses a similar algorithm described by Alemi and Ginsparg [25] to create text segmentation.

Table 2. Training data sample size of 18585 after pre-processing, for transcripts with Flesch scores less than 80.

Data Filtering Process	No. Episodes
Number of episodes in dataset	105360
After removing non-English	103005
Filtering with Flesch metrics	57390
Filtering 60 min word count	18585

The TextSplit segmentation algorithm [26] iteratively splits the text into coherent segments, divided by sentence boundaries, which are identified using a greedy segmentation approach. We denote V to be a segment of sentences $(S_i, ...S_n)$, and v to represent the sum of the sentence vectors, $v := \Sigma_i s_i$. The cosine similarity of the sentence vector s_i to the segment vector v is computed, with repeated sentences being suppressed by weighted coefficients. This results in coherent segments as similar sentences are attributed to the segment vector. The length of segment vector v is determined by an aggregated score of the cosine similarity of the sentence vectors $(s_i, ...s_n)$. The boundary split position t, in the transcript length L can be noted as:

$$T := (t_0, ..., t_n) \tag{1}$$

$$0 = t_0 < t_i < ... < t_n = L \tag{2}$$

To select the appropriate split position t, the greedy algorithm determines a split position to the left b and right e, of t:

$$b < t < e \tag{3}$$

The sum of norms of segment vectors to the left and right of t, minus the norm of the segment vector v, produces the $gain$ of a split:

$$gain_b^e(t) := \left\| \sum_{i=b}^{t-1} s_i \right\| + \left\| \sum_{i=t}^{e-1} s_i \right\| - \left\| \sum_{i=b}^{e-1} s_i \right\| \tag{4}$$

The sum of the $gains$ is used to form a segmentation $score$, formulated as:

$$score(T) := \sum_{i=1}^{n-1} gain_{ti-1}^{ti+1}(t_i) \tag{5}$$

The transcript is split iteratively until the gain of the split is the highest, after subtracting a penalty parameter. In our experiments we select the number of splits as a parameter (e.g. 4, 6, 8, 10), to control the number of segments the transcript is split into. The penalty threshold is adjusted to optimize the required segmentation.

[1] https://pypi.org/project/langdetect/
[2] https://pypi.org/project/textstat/

Ablation Studies for Text Segmentation: We vary the number of segmentation splits in our ablation studies. The maximum splits parameter is set to 4, 6, 8 & 10, with various combinations of topic segments. Each segment split represents a subtopic narrative, which are labeled in sequential numerical order to facilitate segment exclusion for the experiments. The process of removing some of the segments, results in up to half of the original input data being presented to the model for decoding. The splits are not uniform in size, therefore some segments will be larger or smaller than others.

We hypothesize that text segmentation will result in more details being revealed in summaries, as the model is forced to input specific topic segments. We define different types of summaries produced in our experimental results:

Synopsis Summary: The first one or two segments are considered by the summarizer. The content should reflect an introductory style summary, that give users an overall feel for what the show is about. For example, splits **6** (1, 2).

Spoiler Summary: The first segment in each episode is conserved, to give users a contextualized introduction, before disseminating particular spoiler content, from middle or end segments. For example, splits **6** (1, 4, 5), splits **8** (1, 5, 6), or first and last **10** (1, 10).

Walkthrough Summary: Segments include all the conversational texts in each episode, resulting in an outline style summary that walks users through the podcast narrative. E.g, all splits in **6** (1, 2, 3, 4, 5, 6) or **8** (1, 2, 3, 4, 5, 6, 7, 8).

Truncated Summary: Similar to the walkthrough, all splits are used, however the text is truncated after a maximum token length of 1024. This results in a synopsis style summary as the majority of text will come from beginning segments.

The number of splits can be increased to 12, 14, 16 etc, however, a minimum sentence or token length parameter is required to derive a higher number of topic splits. For the evaluation data, a maximum of 10 splits was adequate as this resulted in 29 episodes out of 1027, where topics could not be extracted. 12 splits results in 41 episodes not extracted and 6 splits resulted in only 11 episodes not extracted. In this scenario, all tokens are taken as input. We set the truncation summary length to 1024, to replicate a scenario where a model has a limited input size. This parameter can be set to any number, depending on experiment requirements.

Alternative summaries were considered, by mixing different segment splits, or forcing the summarizer to select a minimum number of sentences per segment. However, in this domain, the first segment must be included as it contains topic and guest introductions, thereby limiting other summary combinations.

3.3. Model Training (Stage 3)

We select a variant of the longformer model, developed with both traditional encoder and decoder architecture, so we can analyze when a full transcript is used as input, compared to a filtered version. The LED model was pre-trained using a Masked Language Modelling (MLM) approach, initialized from the checkpoint weights of BART Base [5]. Able to capture up to 16,384 tokens, we fine-tune an LED base model[3] from the Huggingface transformers library [31]. To set the maximum input length for the model, we compute the token length of each transcript in the full Spotify dataset. Ranging from 8 to 43,504 tokens, we select the third quartile value as an input sequence length of 8,671. This represents the length of up to 75% of transcripts without requiring additional memory capacity. We fine-tune our LED model and train for 3 epochs with a batch size of 1. Applying the model to the evaluation data for summary generation through decoding, we vary the minimum output length to 60 or 90 tokens, depending on the transcripts input length. We set the maximum target output length as 220, based on token sizes in previous mentioned studies, and use a beam search strategy of 3.

4. EXPERIMENTS

4.1. Evaluation Strategy

To evaluate our work, we perform an ablation study to compare the segment splits to each other, through metrics and qualitative analysis. We also compare scores with participant submissions to the summarization track of the TREC 2020 competition [32, 8, 9]. However, we note the objective of the competition was to produce short summaries from podcasts that can fit onto a smartphone screen. Manakul and Gales [32] explored ensemble models with sentence filtering and hierarchical attention, receiving the highest human and automatic evaluation scores. Rezapour et al. [8] generated summaries to reflect the same category style as the podcast and utilized named entities in their model. Karlbom and Clifton [9] proposed a Longformer model with an input size of 4096, a similar approach to our method.

Entries were evaluated using ROUGE-L scores [33], measuring the longest common subsequence (LCS) between creator descriptions and generated summaries. We also compute F1-scores, to provide a balance between the two metrics, precision and recall. In addition to ROUGE, we use METEOR [34] metrics, shown to have better correlation with human judgment.

There were 1027 episodes in the 2020 evaluation testset with varying description lengths. As ROUGE metrics measure n-gram overlapping, long descriptions are required to fairly evaluate long summaries. To address this issue, we selected all the episodes with Flesch scores under 80 and descriptions lengths over 100 words, acknowledging that promotional material is present in that word count. This resulted in 160 episodes of evaluation data, suitable for text segmentation for our long text summarizer model.

[3] https://huggingface.co/allenai/led-base-16384

Table 3. Performance of our model applied to eval data filtered by numerical topic splits. Results are measured by ROUGE-L Precision, ROUGE-L Recall, ROUGE-L F1 and Meteor scores. '# Av.W' denotes the average number of tokens generated in the summary. '# Input' denotes the average number of tokens in transcripts available for decoding and comparative percentages.

Evaluation Subset 160/1027								
Type	**Splits**	↑ **Precision**	↑ **Recall**	↑ **F1-Score**	↑ **Meteor**	↑ **# Av.W**	↓ **#Input**	↓ **#%**
walkthrough	**0**	0.324	**0.191**	**0.229**	**0.225**	**115**	6553	100%
truncated	**0**	**0.334**	0.178	0.221	0.211	105	**1024**	**16%**
synopsis	**4** (1)	**0.315**	0.163	0.206	0.197	99	**819**	**12%**
synopsis	**4** (1, 2)	0.311	**0.175**	**0.214**	**0.210**	**108**	1914	29%
spoiler	**4** (1, 3)	0.307	0.168	0.206	0.203	106	1911	29%
spoiler	**4** (1, 4)	0.310	0.168	0.207	0.203	**108**	2917	45%
synopsis	**6** (1, 2)	**0.321**	**0.176**	**0.216**	**0.208**	105	**1287**	**20%**
spoiler	**6** (1, 4)	0.310	0.166	0.207	0.198	104	1375	21%
spoiler	**6** (1, 6)	0.298	0.164	0.200	0.197	107	1816	28%
spoiler	**6** (1, 4, 5)	0.298	0.165	0.204	0.201	**109**	2235	34%
middle	**6** (2, 3, 4)	0.216	0.118	0.145	0.159	107	2430	37%
synopsis	**8** (1, 2)	**0.315**	0.170	**0.212**	0.204	103	**972**	**15%**
spoiler	**8** (1, 6)	0.309	0.164	0.205	0.195	102	1037	16%
spoiler	**8** (1, 8)	0.309	0.165	0.203	0.198	103	1386	21%
spoiler	**8** (1, 5, 6)	0.308	**0.174**	**0.212**	**0.207**	**110**	1713	26%
spoiler	**8** (1, 7, 8)	0.294	0.165	0.199	0.199	109	1956	30%
synopsis	**10** (1, 2)	**0.319**	**0.168**	**0.211**	**0.202**	100	**771**	**12%**
spoiler	**10** (1, 10)	0.298	0.156	0.194	0.191	102	1163	18%
spoiler	**10** (1, 6, 7)	0.309	0.165	0.206	0.197	106	1410	22%
spoiler	**10** (1, 8, 9)	0.286	0.158	0.194	0.193	108	1254	19%
middle	**10** (5, 6, 7)	0.185	0.109	0.130	0.152	**113**	1590	24%

Table 4. Model's performance on 160 episodes in the data subset, compared TREC 2020 participants.

Evaluation Subset 160/1027					
Model	**Precision**	**Recall**	**F1-Score**	**Meteor**	**# Av.W**
cued_speechUniv2 [32]	0.388	0.142	0.199	0.161	66
category-aware2 [8]	0.387	0.139	0.183	0.155	74
hkuupodcast1 [9]	**0.419**	0.131	0.186	0.144	57
walkthrough	0.324	**0.191**	**0.229**	**0.225**	**115**

Table 5. Model's performance on 1027 episodes, compared to TREC 2020 participants.

Evaluation Data 1027/1027					
Model	**Precision**	**Recall**	**F1-Score**	**Meteor**	**# Av.W**
cued_speechUniv2* [32]	0.235	0.224	**0.197**	0.216	57
category-aware2* [8]	0.258	0.199	0.184	0.191	58
hkuupodcast1* [9]	**0.265**	0.190	0.192	0.193	45
walkthrough	0.153	**0.242**	0.158	**0.219**	**108**

4.2. Experimental Results

In Table 3, we carry out extensive ablation studies to compare different types of summaries, produced with data filtered by text segmentation splits. We analyze the percentage of data presented to the model, the average word count and metric scores. In the walkthrough summaries, the full transcript is used as input to the model. In truncated summaries, only the first 1024 tokens are considered with the remainder being discarded. Walkthrough and truncated are considered as baselines for comparative analysis. The best performing synopsis style summary against the baselines was splits **6** (1, 2). Considering on average only 20% of the segments were presented for decoding, a high F1-score of 0.216 is achieved. When the first two segments are taken from 8 and 10 splits, the percentage of input data is reduced, and F1-scores are accordingly lower. This suggests that more segments of text are required when the number of topic splits increases.

When the first and last segments are taken, i.e. **6** (1, 6), **8** (1, 8) and **10** (1, 10), lower recall, F1, and meteor scores are recorded. This is because creator descriptions are unlikely to contain information from the end of the podcast, and often contain promotional material. Segment splits without the first topic, i.e. **6** (2, 3, 4) had the lowest scores, as the summarizer is unable to maintain structure without the first segment. Summaries would therefore be prone to hallucinating name entities and other details. Overall the results demonstrate it is unnecessary to make use of the entire transcript, as appropriate filtering can create comparable results on ROUGE.

Table 4 presents the results of our walkthrough summaries compared to TREC 2020 participants. Overall the model sig-

Table 6. Examples of generated podcast episode summaries, highlighting fine-grained details from different topic segment splits from episode: 1fE39oSODbzEQt3u2mSC3K.

Creator Description	Topic Splits Summaries
Just when I think I've heard it all, I am always **pleasantly thrilled when a story comes along that surprises me**. That's what today's episode is about - **people who have witnessed something** that is difficult to categorize. **Unexplainable things**, that you may or may not heard of, that are possibly not of our world. First, I welcome back Derek Hayes, host and creator of the Monsters Among Us podcast, as he describes a **cryptid he witnessed as a child that set him on his paranormal journey**. Then I chat with Susan Slaughter, paranormal investigator, as she describes her **nocturnal visit from a hag**. And finally, I speak with Mark Alan Miller, former Vice President of **Seraphim**, Clive Barker's production company. I've been collecting ghost stories for years, and Mark's is truly one of my favorites. You definitely don't want to miss it. **Chapter 1: Alien Big Cat**, with Derek Hayes, Monsters Among Us **Chapter 2: The Hag**, with Susan Slaughter **Chapter 3: The Man Whose Features Were Crawling On His Face**, with Mark Alan Miller, Encyclopacalypse Publications.	**6 (1, 2)** I welcome back Derrick Hayes, host and creator of The Monsters Among Us podcast, as he describes a **cryptid he witnessed as a child** that set him on his **paranormal journey**. Ian's company has been collecting ghost stories for years, ... truly one of my favorites. You definitely don't want to miss it. **Chapter 1: Alien: Big Cat.** When **Derek was around 10 years old** ... **6 (1, 6)** **Chapter 1: Alien** ... Can I Help You? What Do You Need Silence? I guess I use the term **"face loosely"** because it is a **blank slate with features crawling around his face.** The eyes and mouth and nose are walking around this **blank Slate of a face** and when he **stands up to scream to shout** whatever he was going to do in the moment, **the figure is gone**. And that is the point where **he starts texting me**. He is **freaking out**, he is not having a good time and **I text him back.**" . **6 (1, 4, 5)** I've been collecting ghost stories for years, and this is truly one of mine favorites. You definitely don't want to miss it. **Chapter 1: Alien**: I am Mark Alan Miller ... work for famed horror writer and artist Clive Barker. This is maybe **10-11 years ago** then and I was new to the company... Clive operated out of his **sprawling mansion in Los Angeles** ... He lives in this **enormous 3-story 3 Wing mansion, in Beverly Hills**. All these insane things happened there when it was first built, **he wrote a book about it called "Cold Heart Canyon".** It was the **book inspired by his experiences** living in the house ... " **0 (walkthrough)** ... I am always **pleasantly thrilled when a story comes along that surprises me**, ... That's what this episode is about, **people who have witnessed something** that is difficult to categorize, **unexplainable things** ... I welcome back Derrick Hayes, host and creator of the Monsters Among Us podcast, as he describes a **cryptid he witnessed as a child that set him on his paranormal journey**. Then I chat with Susan Slaughter, a paranormal investigator as she describes her **nocturnal visit from a hag** and finally I speak with Mark Allen Miller, former vice president of **seraphim** Clive Barker's ..." " **0 (truncated)** ... spoken with a lot of people about the supernatural experiences ... I am always **pleasantly thrilled** when a story ... **people who have witnessed something** that is difficult to categorize, **unexplainable things** ... I welcome back Derrick Hayes, host and creator of the Monsters Among Us podcast, as he describes a **cryptid he witnessed as a child** that set him on his **paranormal journey**. Then I chat with Susan Slaughter, a **paranormal investigator** as she describes her **nocturnal visit from a hag** and finally I speak with Mark Allen Miller, former vice president of seraphim, Clive Barker's ..."

nificantly outperforms the existing systems [32, 8, 9] yielding higher recall, F1 and meteor scores. While precision was lower than [9], the balance between accuracy and quality is proportionate, in contrast to methods scoring high on precision and low on recall. We attribute this performance to the decoding strategy, setting a high minimum word length and length penalty parameter.

In Table 5, we evaluate our model on all 1027 episodes of evaluation data, without segmentation (walkthrough). The results follow a pattern where recall scores are high, as longer summaries are generated. However, precision and F1-scores are lower as the majority of descriptions computed against ROUGE-L are short. As expected, the models ability to generate shorter summaries was constrained, due to restrictive training parameters. Meteor scores were on par with summaries from [32], which received high scores from human judgment evaluation in the TREC competition. Comparing length, we achieved an average of 108, while other models produced just under 57.

Looking at the summaries from a qualitative perspective, we compare topic segment splits, as shown in Table 6. The truncated, walkthrough and synopsis style summaries (1, 2) have higher ROUGE scores and contain similar information as text is taken from the beginning of the podcast. However, when the summarizer is presented with text filtered from the middle or end, in (1, 4, 5) and (1, 6), it is evident that more details are revealed. The richness and diversity of these types of long detailed summaries, are not captured with high ROUGE scores in the experiments. Therefore, additional qualitative examples [4] are available in the supplementary material.

5. CONCLUSIONS

We have proposed a method using a long sequence transformer to generate fine grained summaries of podcast conversations. We investigated the effectiveness of pre-processing and curating training data reflective of reading difficulty levels. We experimented with text segmentation as an alternative method to truncating data after the input limit is reached. Our approach encouraged the model to produce longer summaries, able to capture more fine-grained details by excluding specific segments of text. Our model demonstrates an improved performance, compared to previous work, when tasked with generating longer sequences of text. Future research will extend this work to other domains, to reduce long text for models with a limited token input size, and provide an increased level of sub-topic details.

6. ACKNOWLEDGMENTS

This research was conducted with the financial support of Irish Research Council (IRC) under Grant Agreement GOIPG/2019/2353 and the ADAPT SFI Research Centre under Grant No. 13/RC/2106_P2.

[4] https://github.com/sigmedia/spoken-documents

7. REFERENCES

[1] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones, "100,000 podcasts: A spoken english document corpus," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5903–5917.

[2] Rosie Jones, Hamed Zamani, Markus Schedl, Ching-Wei Chen, Sravana Reddy, Ann Clifton, Jussi Karlgren, Helia Hashemi, Aasish Pappu, Zahra Nazari, Longqi Yang, Oguz Semerci, Hugues Bouchard, and Ben Carterette, "Current challenges and future directions in podcast information access," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

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

[4] Potsawee Manakul and Mark Gales, "Long-Span summarization via local attention and content selection," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6026–6041.

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, "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*, 2020, pp. 7871–7880.

[6] Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, and Ling Fan, "A Two-Phase approach for abstractive podcast summarization," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. 2020, NIST.

[7] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J F Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu, "TREC 2020 podcasts track overview," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. 2020, NIST.

[8] Rezvaneh Rezapour, Sravana Reddy, Ann Clifton, and Rosie Jones, "Spotify at TREC 2020: Genre-Aware abstractive podcast summarization," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. 2020, NIST.

[9] Hannes Karlbom and Ann Clifton, "Abstract podcast summarization using BART with longformer attention," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. 2020, NIST.

[10] Iz Beltagy, Matthew E Peters, and Arman Cohan, "Longformer: The Long-Document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[11] Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan, and Dragomir Radev, "An exploratory study on long dialogue summarization: What works and what's next," in *Findings of the Association for Computational Linguistics: EMNLP*, 2021, pp. 4426–4433.

[12] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Others, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17283–17297, 2020.

[13] Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu, "Progressive generation of long text with pretrained language models," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 4313–4324.

[14] Sowmya Vajjala and Detmar Meurers, "Exploring measures of "readability" for spoken language: Analyzing linguistic features of subtitles to identify age-specific tv programs," in *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, 2014, pp. 21–29.

[15] May Kristine Jonson Carlon, Nopphon Keerativoranan, and Jeffrey S Cross, "Content type distribution and readability of MOOCs," in *Proceedings of the Seventh ACM Conference on Learning@ Scale*, 2020, pp. 401–404.

[16] Elliot Schumacher and Maxine Eskenazi, "A readability analysis of campaign speeches from the 2016 US presidential campaign," *arXiv preprint arXiv:1603.05739*, 2016.

[17] Sravana Reddy, Mariya Lazarova, Yongze Yu, and Rosie Jones, "Modeling language usage and listener engagement in podcasts," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 632–643.

[18] Edgar Dale and Jeanne S Chall, "A formula for predicting readability: Instructions," *Educational Research Bulletin*, vol. 27, no. 2, pp. 37–54, 1948.

[19] J Peter Kincaid, Robert P Fishburne, Jr, Richard L Rogers, and Brad S Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," Tech. Rep., Naval Technical Training Command Millington TN Research Branch, 1975.

[20] R Flesch, "A new readability yardstick," *J. Appl. Psychol.*, vol. 32, no. 3, pp. 221–233, June 1948.

[21] Charuta Pethe, Allen Kim, and Steve Skiena, "Chapter captor: Text segmentation in novels," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8373–8383.

[22] Wen Xiao and Giuseppe Carenini, "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)*, 2019, pp. 3011–3021.

[23] Marti A Hearst, "Text tiling: Segmenting text into multi-paragraph subtopic passages," *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[24] Martin Riedl and Chris Biemann, "Text segmentation with topic models," *Journal for Language Technology and Computational Linguistics*, vol. 27, no. 1, pp. 47–69, 2012.

[25] Alexander A Alemi and Paul Ginsparg, "Text segmentation based on semantic word embeddings," *arXiv preprint arXiv:1503.05543*, 2015.

[26] Christoph Schock, "TextSplit," https://github.com/chschock/textsplit, 2020, Accessed: 2021-11-1.

[27] Jiaao Chen and Diyi Yang, "Multi-View Sequence-to-Sequence models with conversational structure for abstractive dialogue summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4106–4118.

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

[29] Nils Reimers and Iryna Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[31] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush, "Transformers: State-of-the-Art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[32] Potsawee Manakul and Mark Gales, "CUED_SPEECH at TREC 2020 podcast summarisation track," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. 2020, NIST.

[33] Chin-Yew Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, July 2004, pp. 74–81.

[34] Satanjeev Banerjee and Alon Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.