A Two-Phase Approach for Abstractive Podcast Summarization

Chujie Zheng
chz@udel.edu
University of Delaware, USA

Kunpeng Zhang
kpzhang@umd.edu
University of Maryland, USA

Harry Jiannan Wang
hjwang@udel.edu
University of Delaware, USA

Ling Fan
lfan@tongji.edu.cn
Tongji University, China

ABSTRACT

Podcast summarization is different from summarization of other data formats, such as news, patents, and scientific papers in that podcasts are often longer, conversational, colloquial, and full of sponsorship and advertising information, which imposes great challenges for existing models. In this paper, we focus on abstractive podcast summarization and propose a two-phase approach: sentence selection and seq2seq learning. Specifically, we first select important sentences from the noisy long podcast transcripts. The selection is based on sentence similarity to the reference to reduce the redundancy and the associated latent topics to preserve semantics. Then the selected sentences are fed into a pre-trained encoder-decoder framework for the summary generation. Our approach achieves promising results regarding both ROUGE-based measures and human evaluations.

1 INTRODUCTION

The summarization task has been well studied in Natural Language Processing (NLP). Especially within the development of deep learning and the introduction of attention mechanism[10], Transformer-based summarization models [7, 9, 11, 12] have achieved remarkable performance. However, these Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length [1, 6]. This bottleneck has brought challenges for podcast summarization since the average length of podcast transcript is much longer than the maximum sequence limitation. When these Transformer-based summarization models can only read the first hundreds of tokens in the episode transcript, how to generate a comprehensive summary covering the important information is a challenge.

Besides the challenge from the length, podcast summarization is a complicated research problem because of the conversational feature. Comparing to the professionally edited texts like news and academic papers, the podcast is colloquial and contains many conversations. According to the number from Spotify, 81.4% of the podcast episodes are conversational[1]. But few studies have focused on how to deal with conversational corpus summarization. It is still a research gap.

This short paper is for the summarization task of TREC 2020 Podcast Track [5]. In this work, we introduce a two-phase approach for podcast abstractive summarization. It is designed based on the unique features of the podcast. Our proposed approach selects the important sentences from the transcript in the first phase and uses the encoder-decoder network to generate the abstractive summary based on the selection. Figure 1 describes the general framework for proposed approach.

The key contribution of this paper focuses on the first phase, which selects the important sentences from the input documents. Building on top of the Transformer-based models, our goal is to filter the irrelevant content and locate the most useful information for the abstractive summary generation. Under this idea, the research question becomes how to define important sentences? We propose two methods to identify the important sentences using the ROUGE score and the topic-level features. Our proposed approach provides some novel insights for podcast summarization and improves the performance of the summarization model.

2 DATASET

The dataset used in this work is the TREC Spotify podcast dataset [3, 4] which has 105,360 podcast episodes from 18,376 shows produced by 17,473 creators. The average duration of a single episode is 30 minutes, while the longest can be over 5 hours and the shortest is only 10 seconds. The TREC Podcast Track organizers form the "Brass Set" by cutting down the dataset to 66,245 podcast episodes using the following rules:

- Remove episodes with descriptions that are too long (> 750 characters) or too short (< 20 characters);
- Remove "duplicate" episodes with similar descriptions (by conducting similarity analysis);
- Remove episodes with descriptions that are similar to the corresponding show descriptions, which means the episode description may not reflect the episode content.

We perform additional data preprocessing on top of the Brass Set as follows:

- Remove episodes with profanity language in the episode or show descriptions as [9].
- Remove episodes with non-English descriptions.
- Remove episodes whose description is less than 10 tokens[2].

After preprocessing, the dataset has 52,140 episodes left (see Table 1 for details). We randomly split the dataset into training, validation, and testing by 80%, 10%, and 10%. This processed dataset is used for our training.

[1] Numbers are reported in Matthew Sharpe's presentation "A Review of Metadata Fields Associated with Podcast RSS Feeds" in RecSys 2020 PodRecs Workshop

[2] We perform some data preprocessing for episode description, including using some rule-based methods to remove the social media link and sponsorship.

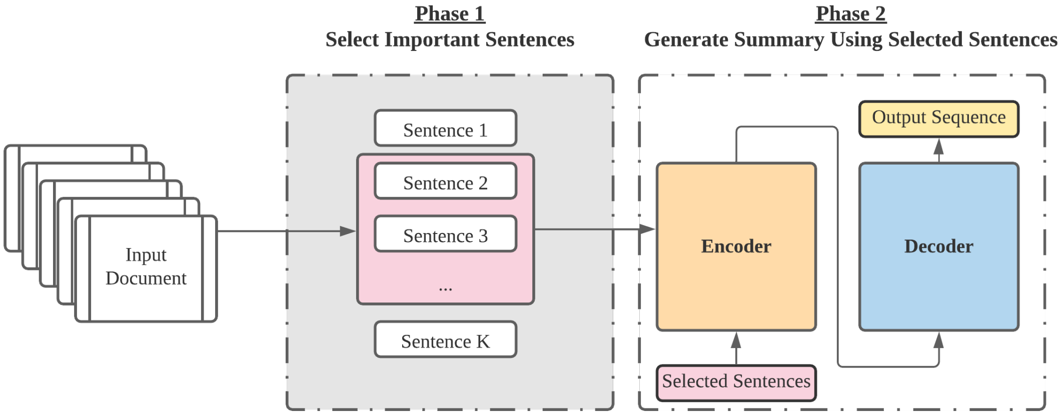

Figure 1: Framework for Two-Phase Approach

Dataset Preprocessing	# of Episodes
TREC Spotify Podcasts Dataset	105,360
After filtering by the TREC organizer (Brass Set)	66,245
After removing episodes with profanity language	55,799
After removing episodes with non-English descriptions	55,383
After removing episodes whose description is too short	52,140

Table 1: Data Preprocessing and the Number of Episodes

3 TWO-PHASE APPROACH

In this section, we introduce our **Two-Phase Approach** for podcast summarization. The key motivation is to select important sentences from the input document and use it as the input for the abstractive summarization model. In this work, we don't design a new model for Phase 2, so we choose to use the BART framework, which performs the best in the preliminary experiment [13]. In practical implementation, we use distilBART [3] provided by HuggingFace. Our focus is on how to define an efficient metric to select sentences contained important information. So the research problem turns to how to define the important sentences.

We propose two sentence selection methods to identify the sentences with important information. The importance is determined based on whether the sentence covers the key information of the article or whether it reflects the key topic of the article.

3.1 ROUGE-based Approach

Inspired by [12], we propose a ROUGE-based approach to identify the sentence who covers the key information of the input document. We calculate the ROUGE score [8] for each sentence with the input document, which considers as the relevance score for this sentence. ROUGE is the most common evaluation metric for the summarization task, which focuses on the overlapping between the reference summary and the model output. The intuition for the **ROUGE-based Approach** is a sentence with a higher ROUGE score serves as an excellent summary of the input document, which should capture the important information.

In practical implementation, since there are three types of ROUGE scores commonly used: ROUGE-1, ROUGE-2, and ROUGE-L, we choose the average of three scores as the measurement to find the most relevant sentences. We introduce our detailed implementation **Sliding Window ROUGE-based Approach** and **Novelty-Enhanced ROUGE-based Approach** in the following.

3.1.1 Sliding Window ROUGE-based Approach. We select the sentences in a window size as the candidate and compute the ROUGE score, as Figure 2. This implementation is based on two reasons. First, it is very time consuming to calculate the score for every single sentences, especially for the long document. In addition, we believe selecting single sentence may lead to the redundancy in the information. In other words, we are using ROUGE score as the measurement to selecting sentences $\mathbf{x}$ in the window size w, as Equation 1. s_k represents the k-th sentence in the input document, which contains N sentences.

$$\mathbf{x} = \text{argmax}_{s_{i:i+w}} \text{ROUGE}(s_{i:i+w}, s_{1:N}) \tag{1}$$

Here w is chosen based on the consideration that the average length of the selected sentences should have approximately 1024 tokens, which is the maximum sequence limit we set for BART, followed the instructions provided by HuggingFace [4]. In our implementation, we set $w = 40$ since the average number of tokens is 1018 tokens, which satisfied our constraints.

3.1.2 Novelty-Enhanced ROUGE-based Approach. From the implementation of the Sliding Window ROUGE-based Approach, it brings

[3] https://huggingface.co/sshleifer/distilbart-cnn-12-6

[4] https://github.com/huggingface/transformers/tree/master/examples/seq2seq

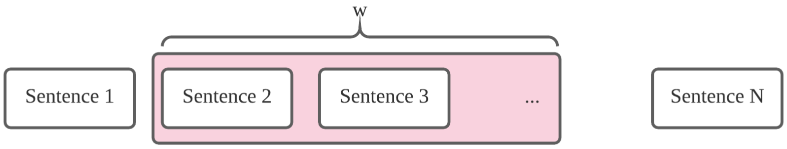

Figure 2: Framework for ROUGE-based Approach

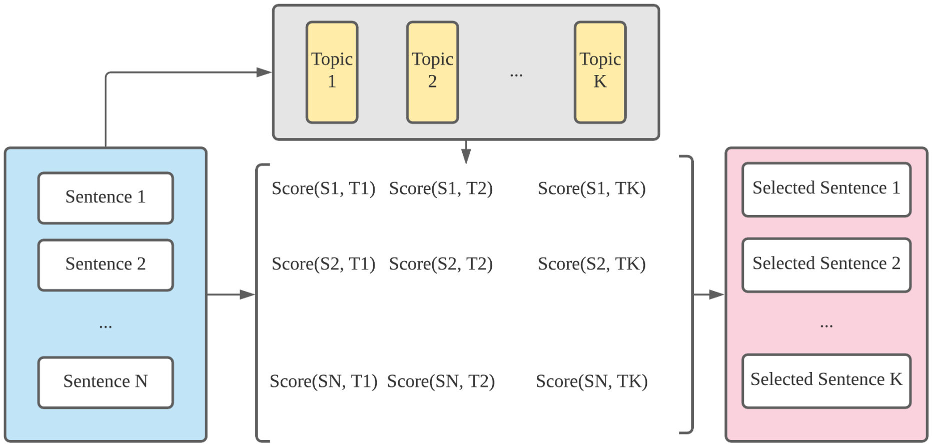

Figure 3: Framework for Topic-Enhanced Approach

us another question: since we are selecting the sentences in consecutive order, is it possible that we may miss some single important sentences? To solve this question, we propose a new approach: besides finding the sentences $s_{i:i+w}$ in the window size w, we also calculate the ROUGE score for each sentence s_j and check if it is selected already in the window. If not, we will combine it with the selected window as the summary. This approach aims to enhance the performance of the ROUGE-based approach in case we miss any important single sentence.

We choose the top $k = 5$ sentences with the highest ROUGE score and check if these sentences are in the selected sentences under the window size $w = 25$. These two parameters are chosen based on the requirement that the average length of the final selection satisfies the maximum length limit for the Transformer.

3.2 Topic-Enhanced Approach

The second proposed approach for important sentence selection is identifying the sentence captures the main topic of the input document. The intuition is to learn the main topics at first and find the sentences that are most relevant to the key topics.

This approach is built on the assumption for topic modeling: documents are represented as random mixtures over topics and each topic is characterized by a distribution over words [2]. As Figure 3 describes, we identify the latent topics and calculate the relevance score for each sentence with the topic embedding. The sentences with the highest relevance score are selected. Using topic modeling, we generate the key topics from the input document. We use Gibbs sampling to learn the topic distribution $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_k)$. For each latent topic $\boldsymbol{\theta}_i$, we calculate the relevance score between each sentence and $\boldsymbol{\theta}_i$, then we select the more relevant sentence as the representation for this topic. These selected sentences are combined as the output in the first phase.

4 EVALUATION

In this section, we summarize the evaluation result for our submitted results. Totally we've submitted 4 runs this year. The detail for each submitted run is summarized here:

- **Method 1:** Using the BART framework without a sentence selection approach to fine-tune the model and generate the results.
- **Method 2:** Using the BART framework with sentences selected by **Sliding Window ROUGE-based Approach** to fine-tune the model and generate the results.
- **Method 3:** Using the BART framework with sentences selected by **Novelty-Enhanced ROUGE-based Approach** to fine-tune the model and generate the results.
- **Method 4:** Using the BART framework with sentences selected by **Topic-Enhanced Approach** to fine-tune the model and generate the results.

4.1 ROUGE Score

The organizers reported the ROUGE-L score between the generated summary and the episode description, which is written by creators. As the most common evaluation metric for summarization, the ROUGE score focuses on the overlapping between the generated summary and the reference summary. Considering the fact that the quality of the episode description varies, it may not be the perfect

	ROUGE-L P	ROUGE-L R	ROUGE-L F
Method 1	**23.87**	16.07	**16.30**
Method 2	20.84	16.01	15.29
Method 3	20.16	**16.83**	15.49
Method 4	18.44	13.90	13.29

Table 2: ROUGE-L Scores for Submitted Summaries

Average	Method1	Method2	Method3	Method4
0.98	**1.12**	1.03	1.08	0.72

Table 3: Average Score for Quality Rating

	%(Excellent)	%(Good)	%(Fair)	%(Bad)
Method 1	7.26%	**21.79%**	46.37%	24.58%
Method 2	5.59%	19.00%	**48.04%**	27.37%
Method 3	**7.82%**	**21.79%**	41.34%	29.05%
Method 4	3.91%	11.73%	36.87%	**47.49%**

Table 4: Percentage of Quality Rating

reference summary. For example, there are many social media links and sponsorships in the episode description, which are irrelevant content. Here the ROUGE score reported in Table 2 is sharing to offer a standard summarization evaluation viewpoint.

4.2 Human Evaluation

As one of the important parts of the evaluation, this year the organizers provide the qualitative judgments for the generated summary. They selected 179 out of 1000 episodes from the submitted set. An assessor first quickly skimmed the episode, and then made judgments for each summary for that episode, in random order. Therefore, for each episode, the assessor will give a score from 0-3 quality according to its quality, where a higher score indicates higher quality. Table 3 reports the performance of our submitted results. We compare our performance with the average score for all participants, and it turns that 3 out of 4 methods perform better than the average. Compared with the average score, Method 1 with the highest quality rating scores 14.3% higher.

We also analyze the percentage for each category in Table 4 for all submitted runs. Method 3 performs the best using this measurement because of its outstanding performance in the high-quality rating categories: 7.82% of the generated summary is rated as "Excellent" and 21.79% of the output is rated as "Good".

For every episode, the assessor is asked eight yes-or-no questions regarding the quality of the summary, and “1" indicates that the answer is “yes”. These questions include:

- **Q1**: Does the summary include names of the main people (hosts, guests, characters) involved or mentioned in the podcast?
- **Q2**: Does the summary give any additional information about the people mentioned (such as their job titles, biographies, personal background, etc)?

	Average	Method1	Method2	Method3	Method4
Q1	0.44	**0.60**	0.52	0.56	0.26
Q2	0.28	**0.34**	0.31	0.28	0.18
Q3	0.58	**0.70**	0.66	0.68	0.50
Q4	0.45	0.51	0.52	**0.63**	0.50
Q5	0.52	0.54	0.58	**0.62**	0.42
Q6	**0.09**	0.05	**0.09**	0.04	0.07
Q7	0.62	0.74	0.63	**0.75**	0.74
Q8	0.45	0.50	0.46	0.51	**0.55**

Table 5: Average Score for Human Evaluation Questions

- **Q3**: Does the summary include the main topic(s) of the podcast?
- **Q4**: Does the summary tell you anything about the format of the podcast; e.g. whether it’s an interview, whether it’s a chat between friends, a monologue, etc?
- **Q5**: Does the summary give you mode context on the title of the podcast?
- **Q6**: Does the summary contain redundant information?
- **Q7**: Is the summary written in good English?
- **Q8**: Are the start and end of the summary good sentence and paragraph start and end points?

We compare the performance of our submitted runs to the average scores for all participants. Except for Q6, a higher score represents better performance. According to Table 5, our submitted runs have achieved better performance than the average for all eight questions. Our models are better at capturing important information and semantically more fluent.

Table 3, 4 and 5 leave us a lot of room for further discussion.

- Firstly, based on the fact that the average rating is only 0.98 and the best performance for our models is 1.12, there is a lot of room for improvement for podcast summarization. Considering that the score is from 0 to 3, the average performance is only "Fair". Especially in comparison with the excellent performance of news summarization, we have a lot of thoughts: What makes podcast summarization so challenging? We are looking forward to this year’s Podcast Track and improve our performance.
- We also raise some concerns about the prepared eight yes-or-no questions. These questions reflect the quality level of the generated summary, which is also considered as the feature for a good podcast summary. Comparing to other summaries, it emphasizes several aspects including additional information like people’s job titles or the format of the podcast. These features definitely play a very important role in helping people familiar with the episode. How to capture these features leads to several directions for future research.

5 CONCLUSION

In this year’s podcast track, we develop a two-phase approach to handle the podcast summarization task. Using the transcript generated by Google ASR, we design a pipeline to select the important sentences which cover the key information of the input document and generate the abstractive summary using the selected sentences.

Our main contribution is that we propose two approaches to select sentences including the ROUGE-based approach and the topic-enhanced approach. These approaches provide a novel definition for important sentences and improve the performance of the podcast summarization model.

REFERENCES

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Ann Clifton, Aasish Pappu, Sravana Reddy, Yongze Yu, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020. The Spotify Podcasts Dataset. *arXiv preprint arXiv:2004.04270* (2020).

[4] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth J. F. Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020. 100,000 Podcasts: A Spoken English Document Corpus. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

[5] Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth Jones, Jussi Karlgren, Aasish Pappu, Sravana Reddy, and Yongze Yu. 2020. Overview of the TREC 2020 Podcasts Track. In *The 29th Text Retrieval Conference (TREC 2020) notebook*. NIST.

[6] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451* (2020).

[7] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[8] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[11] Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063* (2020).

[12] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777* (2019).

[13] Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan. 2020. A Baseline Analysis for Podcast Abstractive Summarization. *arXiv preprint arXiv:2008.10648* (2020).