AUDIO SUMMARIZATION FOR PODCASTS

Aneesh Vartakavi*, Amanmeet Garg*, and Zafar Rafii
Gracenote
Emeryville, CA, USA
firtname.lastname@nielsen.com

***Abstract*—We propose a novel system to automatically generate audio summaries for podcasts, allowing listeners to quickly preview podcast episodes. The proposed system first transcribes the audio from a podcast using automatic speech recognition (ASR), then summarizes the transcript using extractive text summarization, and finally returns the audio associated with the text summary. Motivated by a lack of relevant datasets for this task, we created our own by transcribing the audio from various podcasts and generating summaries for these transcripts using a manual annotation tool. Using these text summaries, we fine-tuned a recent Transformer-based summarization model to specifically handle podcast summaries. Our system achieves ROUGE-(1/2/L) F-scores of 0.63/0.53/0.63, respectively, showing good performance for podcast summarization. We present some examples of podcast audio summaries here: *https://github.com/aneeshvartakavi/podsumm*.**

***Index Terms*—Podcasts, audio summarization, automatic speech recognition, automatic summarization.**

I. Introduction

The recent surge in popularity of podcasts presents a unique set of challenges to existing content discovery and recommendation systems. Attributes such as the speaker's voice, presentation style, type of humor, and production quality can directly influence the listener's preference. In the case of movies, trailers and teasers would typically allow a viewer to get a preview of a feature film and make a subjective decision as to watch it or not. However, frequent release schedules of new podcast episodes would make the manual production of summaries similar to movie trailers impractical.

We propose a system that can generate podcast audio summaries in an automated manner. We focus on generating extractive summaries, where a set of relevant audio sections are selected from the podcast without altering the original content. Such summaries would inform the listener about the topics of the podcast, as well as subjective attributes such as presentation style and production quality, before deciding to listen to the entire episode. We note here that the purpose is not to give the listener a condensed version of the whole podcast, but rather a preview in a manner similar to a movie trailer.

Podcasts present challenges to automatic summarization systems, as they generally contain free-form speech, overlapping speakers, audio effects, background music, and advertisements. An audio-based summarization system trained with supervised learning would require a varied and large amount of manually-annotated training data to tackle such a complex problem. However, as podcasts largely contain spoken-word content, summarization could be performed in the text domain, on the transcript of an episode.

We therefore present *PodSumm*[1], a novel system for automatically generating audio summaries for podcasts using audio transcription and text summarization. PodSumm first transcribes the audio from a podcast using automatic speech recognition (ASR), then summarizes the transcript using extractive text summarization, and finally returns the audio associated with the text summary. We base our text summarization on PreSumm [2], a recent Transformer model that we fine-tune on summarized podcast transcripts with labels obtained using a manual annotation tool. To the best of our knowledge, PodSumm is the first system to automatically generate audio summaries for podcasts.

The rest of the article is organized as follows. In Section II, we present some prior work. In Section III, we describe the proposed system with the data creation and model training. In Section IV, we show the results of our evaluation. We conclude this article in Section V.

II. Prior Work

Podcast summarization has only recently gained interest as a research problem. As far as we know, our work is the first to tackle the problem of audio summarization for podcasts.

Most prior methods for podcast summarization focus on generating text summaries using abstractive summarization, where the original content has been rephrased [3], [4]. A previous work also made use of ASR and text summarization but to generate short one-sentence summaries in the context of document retrieval for podcast search [5], and not for podcast audio summarization. A dataset has also been recently introduced for podcast search and abstractive text summarization [6]. While text summaries can inform listeners about the topics in a podcast episode, they cannot convey subjective attributes that are typically found in the audio itself. Audio summaries would not only allow that, but they would also let users more conveniently listen to a summary rather than read one, for example, while cooking at home or driving.

Speech summarization, which is the summarization of speech directly in the audio domain, is a complex task that a number of methods have attempted to tackle. Prior solutions to this task approached it as a feature classification

*Equal contribution by the two authors.

[1]An initial version of this work was presented at the Podrecs workshop [1], as part of Recsys 2020, for which no proceedings were published.

problem [7], speech-text co-training problem [8], and graph clustering problem [9]. Neural extractive summarization such as reinforcement learning [10], hierarchical modeling [11] and sequence-to-sequence modeling [12] have shown promising results, though on small curated datasets. Automated speech summarization still has many open research problems, for example, with multi-party, spontaneous, or disfluent speech.

Text summarization, on the other hand, has seen more prominent results. Neural models consider this task as a classification problem where an encoder creates a latent representation of the sentences, followed by a classifier scoring the sentences on their importance towards creating a summary [13], [14]. With the rising popularity of deep neural networks, pre-trained language models, particularly Transformers such as BERT [15], have shown promise in a wide range of natural language processing (NLP) tasks. Recent approaches to text summarization such as PreSumm [2] and MatchSum [16] leverage BERT and achieve state-of-the-art performance on a number of benchmark datasets. These present a promising avenue for further development and expansion to other application domains.

III. Methods

A. PodSumm Architecture

Figure 1 shows an overview of the PodSumm system. PodSumm first generates a transcript of the podcast audio using an ASR module and parses the text transcript into individual sentences. It then uses a text summarization model to select relevant sentences, along with their time offsets in the audio, and generates the final audio summary associated with the text summary. Each stage is discussed in detail below.

1) Automatic Speech Recognition: ASR methods perform the task of automatic speech-to-text transcription. As the purpose of this work is not to develop a new ASR system or improve on an existing one, we choose to use a well-known and publicly-available solution for this task, namely AWS Transcribe[2].

2) Text Processing: The transcripts obtained from the ASR module contain the text for the individual words and punctuation marks, their start and end times in the audio, and their confidence scores regarding the prediction. We choose to use an open-source library for NLP, namely spaCy[3], to parse the text into individual sentences with their corresponding start and end times. Additionally, we force a sentence break when a pause of over two seconds between words occurs.

3) Text Summarization: We generate text summaries by selecting relevant sentences from the transcripts, using automatic extractive summarization. We used the recently-proposed PreSumm[4] model [2], which builds upon BERT [15] to obtain a sentence level encoding, and stacks inter-sentence Transformer layers to capture document-level features for summarization. We fine-tune the PreSumm model (which was pre-trained on the CNN/DailyMail dataset [17]) to specifically handle podcasts summaries, by using our own dataset of summarized podcast transcripts, which we describe in the Section III-B.

The extractive PreSumm model performs summarization on a document with sentences $[sent_1, sent_2, \ldots, sent_m]$ by assigning a score $y_i \in [0, 1]$ to each $sent_i$, indicating exclusion from or inclusion in the summary. The model is trained using a binary classification entropy loss to capture difference in prediction $\hat{y_i}$ and ground truth label y_i.

4) Audio Generation: We use the time offsets of the selected sentences in the text summary to identify the corresponding audio sections in the podcast and stitch them together to form the final audio summary.

B. Dataset Creation

Since there is no dataset available for extractive summarization of podcasts, we curated our own to support the development and evaluation of our system. We selected 19 unique podcast series from different genres, with an average (and standard deviation) of 16.3±6.28 episodes per series. The dataset contains 309 different podcast episodes, with an average duration of 36.5±19.8 minutes per episode, for a total of 188 hours of audio.

We built an annotation tool that presented an annotator with the sequence of sentences obtained from the transcript of an episode, as well as the metadata from the podcast feed and the original audio. Each sentence was paired with the respective audio segment using its time offsets. Additionally, the annotation tool dynamically generated audio and text summaries based on the annotator's selection, letting them verify their choices.

The annotator was instructed with the following protocol:

1) Listen to the podcast episode to understand the context and core message.
2) Select the set of sentences that best represent a summary of the episode; the annotators were requested to select continuous sequences of sentences whenever possible, while keeping the total length within 30-120 seconds.
3) Listen to the newly created summary and repeat the above steps if necessary.
4) Submit the annotations when a satisfactory summary is obtained.

Each set of submitted annotations was then associated with a set of sentence indices in the transcript of each podcast episode, corresponding to the most suitable candidate sentences to make a summary.

17 different annotators annotated the 309 podcast episodes. Because of time constraints, each episode was annotated by a single annotator. It took one annotator about 4-9 minutes to annotate a single episode. An average of 14.57±7.01 selected sentences were obtained per summary.

C. Model Training

We begin with the PreSumm model [2] (which was pre-trained on the CNN/DailyMail dataset [17]). We fine-tune the model on our podcast dataset for 10,000 steps, as described

[2]https://aws.amazon.com/transcribe/
[3]https://spacy.io/usage/linguistic-features#sbd
[4]https://github.com/nlpyang/PreSumm

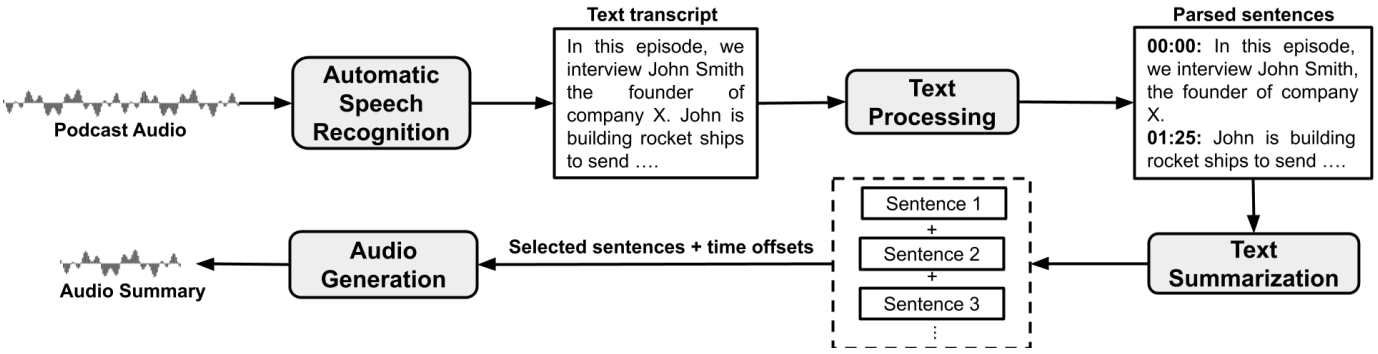

Fig. 1. Overview of the PodSumm system.

in [2]. We use a learning rate of 10^{-4} and keep all the other parameters to their default values. The position embeddings of length 512 were deemed sufficient for our application, as the annotations in our dataset were contained within the first 512 tokens, even for longer episodes. Model checkpoints are saved and evaluated on the test set every 1,000 steps. The best performing checkpoint parameters are used to report the performance of our system. We select the top-*k* sentences from the rank-ordered candidates to create the final summary.

We report the F-score for the ROUGE-(1/2/L) metric [18]. ROUGE-*n* measures the overlap of *n*-grams (contiguous sequence of *n* words) between the predicted and the reference summaries. ROUGE-1 and ROUGE-2 thus refer to the overlap of unigram and bigrams, respectively. ROUGE-L measures the longest common subsequence between predicted and reference summaries. The scores range from 0 to 1, where larger scores represent better performance.

D. Cross Validation

Our current dataset is small in comparison to other datasets [17], [19]. To mitigate the effect of sampling bias, we report the mean and standard deviation of the ROUGE F-scores for a 5-fold cross-validation. The model is trained on the training split (80% or 247 episodes) and the performance is reported on the test split (20% or 62 episodes). The process is repeated for each fold.

E. Data Augmentation

Podcasts can contain advertisements and announcements, which usually repeat across episodes from the same or different series. We find that the fine-tuned model often selects sentences from these repetitive segments, which is undesirable. We propose to augment our dataset by generating new episodes where we replace these repetitive segments with randomly selected repetitive segments from other episodes, and thus, increase the generalization ability of our model.

F. Ablation Studies

1) Effect of number of candidate sentences: Similar to PreSumm, we select the top-k sentences with the highest scores as our predictions. We study the effect of varying the number of sentences selected to represent the summary from the rank-ordered candidates in the model prediction. In our experiment, $k \in (5, 9, 12, 15)$ was varied and the ROUGE-(1/2/L) F-scores are reported.

Metric	R-1 F1	R-2 F1	R-L F1
LEAD-*k* (baseline)			
$k = 5$	0.28 (0.02)	0.17 (0.03)	0.27(0.02)
$k = 9$	0.40 (0.03)	0.26 (0.04)	0.39(0.03)
$k = 12$	0.47 (0.03)	0.32 (0.03)	0.46(0.02)
$k = 15$	**0.52 (0.03)**	**0.39 (0.04)**	**0.51(0.03)**
PreSumm ($k = 12$)			
No FT	0.53(0.02)	0.38(0.02)	0.52 (0.02)
FT	0.63(0.03)	0.51 (0.03)	0.62(0.03)
FT + Aug	**0.64 (0.02)**	**0.53 (0.03)**	**0.63 (0.02)**
PreSumm (FT + Aug)			
$k = 5$	0.56 (0.03)	0.46 (0.04)	0.55 (0.03)
$k = 9$	0.63 (0.02)	0.52 (0.03)	0.62 (0.02)
$k = 12$	**0.64 (0.02)**	**0.53 (0.03)**	**0.63 (0.02)**
$k = 15$	0.63 (0.02)	0.53 (0.03)	0.62 (0.02)

TABLE I
MEAN (AND STANDARD DEVIATION) OF THE ROUGE-(1/2/L) F-SCORES, FOR THE BASELINE, PRESUMM WITH k = 12, AND PRESUMM (FT + AUG) WITH $k \in$ (5, 9, 12, 15), FOR A 5-FOLD CROSS VALIDATION. HIGHER SCORES ARE BETTER. BOLD VALUES ARE THE HIGHEST.

2) Effect of data augmentation: The data augmentation applied during training alters the repetitive content preceding the sentences relevant to the summary. To test the effect of the data augmentation scheme on the model performance, we performed a fine-tuning experiment with and without data augmentation, and report the system performance metrics.

IV. RESULTS

We summarize the results of our evaluation in Table I and report the ROUGE F-scores for the 5-fold cross validation. Similar to prior work [2], [13], [16], we use a simple baseline for comparison, namely LEAD-*k*, where we select the first *k* sentences from the transcript as a summary, with $k \in$ (5, 9, 12, 15). We note that we are unaware of other competitive methods designed for automatic extractive summarization of podcasts.

We find that LEAD-15 performs well, only slightly worse than a pretrained PreSumm model on the CNN/DailyMail dataset. After fine-tuning on our dataset (FT), we find significant improvement for all the ROUGE F-scores. The model with additional data augmentation (FT + Aug) further im-

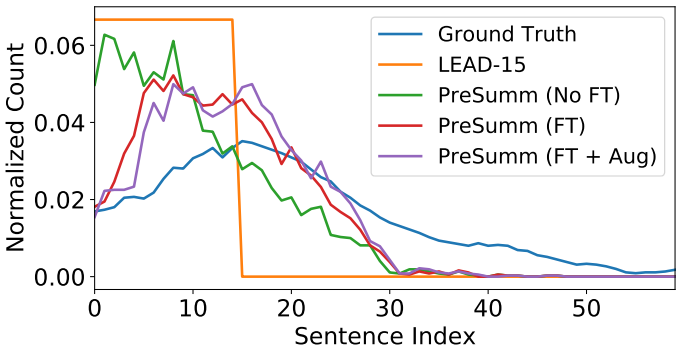

Fig. 2. Distribution of the sentence indices corresponding to the best summary candidates, over all the podcast episodes in our dataset, for the different methods. As the proposed model improves, the ROUGE F-score increases, and the distribution shifts closer towards the ground truth.

proves the performance. We find that selecting the top-12 sentences produces the best results for PreSumm (FT + Aug).

Figure 2 shows the distribution of the sentence indices corresponding to the best summary candidates, over all the podcast episodes in our dataset, for the different methods. The ground truth distribution shows that the very first sentences in a podcast transcript are not necessarily the best candidates for a summary. We can see that the model without fine-tuning, PreSumm (No FT), is relatively biased to select sentences from the beginning of the transcript, which is likely to be a property of the CNN/DailyMail dataset. The models with fine-tuning, PreSumm (FT), and with further data augmentation, PreSumm (FT + Aug), improve the ROUGE F-scores, which can be seen here as a shift in distribution towards the ground truth distribution.

Figure 3 shows an example of podcast transcript with the summary predictions obtained from PreSumm (FT + Aug) with $k = 12$. We can see 9 true positives in green (correctly predicted summary sentences), 1 false positive in blue (incorrectly predicted summary sentence), and 5 false negatives in red (incorrectly ignored summary sentences). We also have 9 correctly predicted repetitive sentences in magenta and 2 incorrectly predicted repetitive sentences in cyan. The sentences in black are true negatives (correctly ignored sentences). This example illustrates that the proposed method is able to correctly identify important sentences from the podcast transcript. The transcript itself shows some errors that have accumulated through the system, such as variations in spoken words (e.g., *.org* incorrectly transcribed as *dot org's*), incorrect sentence segmentation (e.g., between *It is 7:30 p.m.* and *On January, 30th*), etc.

Figure 4 shows another example of a podcast summary created by our trained model pipeline. The interested reader can find some example of audio summaries generated by PodSumm here: https://github.com/aneeshvartakavi/podsumm.

V. Conclusion

We presented PodSumm, a novel system for automatically generating audio summaries for podcasts, using ASR and extractive text summarization. Instead of directly working in the audio domain, we proposed to work in the text domain by leveraging existing state-of-the-art ASR and automatic summarization tools. We fine-tuned a recent extractive text summarization model based on a Transformer with manually-annotated transcripts to specifically handle podcast summaries.

Our evaluation showed that the sentences selected by PodSumm largely agree with the ground truth. These extractive text summaries can then easily be used to produce the final audio summaries without altering the original content of the podcast, retaining subjective attributes such as production quality and presentation style. This is a key difference and benefit of our system in comparison with other recent works on podcast summarization which can only provide text summaries and will alter the original content [3], [4].

Our final trained model showed the best performance (ROUGE-L F-score of 0.64) in comparison with the baseline method (ROUGE-L F-score of 0.52). The podcast-specific data augmentation proposed in this work was able to improve the robustness of the generated summaries, leading to higher ROUGE F-scores. We believe tht the relatively poor performance of the pre-trained PreSumm model (in comparison to LEAD-15) could be attributed to the differences between the task it was originally trained for (news summarization) and podcast summarization. Additionally, the podcast transcripts contain advertisements, transcription errors, and speaker disfluencies, which are not present in the CNN/DailyMail news dataset.

This work demonstrates that it is possible to generate meaningful audio summaries for podcasts with the help of extractive text summarization mechanisms. Given the complex nature of such problem, we believe there is plenty of room for improvements.

Future works should include the creation of larger datasets for extractive summarization to help with the development of more accurate models. Given the subjective nature of audio summaries, gathering multiple annotations from different annotators for the same episode could further help to improve performance. More robust and podcast data-specific ASR and automatic summarization tools and/or proper post-processing mechanisms could correct potential transcription and summarization errors. Additional techniques such as speech/music classification and speaker diarization could also be beneficial as podcasts generally have background music and multiple speakers.

As this is a relatively new field of research, we hope that this work was able to provide some useful insights and we are looking forward to newer methods emerging from the research community leading to an improved listener experience.

References

[1] Aneesh Vartakavi and Amanmeet Garg, "PodSumm: Podcast audio summarization," PodRecs: The Workshop on Podcast Recommendations, September 25 2020.

[2] Yang Liu and Mirella Lapata, "Text summarization with pretrained encoders," in *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, November 3–7 2019.

Hey there real quick before we start the show. California. We are coming your way for a live show next month on February 19th we are super excited to finally get to come to Southern California. We will be in 1000 Oaks with K C. L. You talking about the 2020 race and more to get your tickets, head over to NPR prisons dot org's. Oh, and if you're in Iowa, we have a show there tomorrow night Friday night, and there are still a few tickets available. Okay, here's the show. Hey there, it's the NPR politics podcast. It is 7:30 p.m. On January 30th. I'm Tamara Keith. I cover the White House. I'm Aisha Roscoe. I also cover the White House and I'm Susan Davis. I cover Congress. Senate will convene as a court of impeachment today. The Senate impeachment trial is continuing with more questions and answers, senators asking questions, the House managers and the president's legal team answering those questions. And in fact, as we take this, the Q and A is still going on, so things could happen. That's why we do a time stamp. Um, Aisha, I'm wondering what stood out to you about today? Well, a lot of what the questions seem to be about was getting at this idea of. Is there a limit to what a president can do to get re elected? Because one of the president's lawyers representing him, Alan Dershowitz, made this argument that most presidents think their re election is in the public interest. And therefore, if they take actions to kind of help their reelection as long as it's not illegal, it's OK. And it really seemed like the senators were probing the limits of how far that argument can go on. At one point, there was a question from Senator Susan Collins from Maine, a Republican, and and a few other Republicans, including Senators Crepeau, Blunt and Rubio. And remember, all of the questions are submitted in writing to the chief justice, who then reads them aloud.

Fig. 3. Summary predictions from PreSumm (FT + Aug) with $k = 12$. True positive sentences are in green, false negatives in red, and false positive in blue. Correctly and incorrectly predicted repetitive sentences are in magenta and cyan, respectively. Sentences in black are true negatives.

Hey, it's guy here. So by now, you might have heard that Jake Burton Carpenter, the founder of Burton Snowboards, died this past week. Jake was 65 and in his honor, we thought we would release this interview I did with him back in July of 2017 was actually an incredible experience. Jake flew down from Vermont for this interview, and he brought me a bottle of Vermont maple syrup. But what he really brought was himself. Um, and as you can hear in this interview, he was really vulnerable and really human. And this is a guy who helped to elevate snowboarding and turn it into an international sport. Anyway, all of us who got a chance to meet him feel really lucky that we did. And if you haven't heard this interview or even if you have, it's really worth hearing. So take a listen in honor of Jake. I mean, I was like Willy Loman and I was a traveling salesman and I would load up My car was a Volvo wagon at the time, but I remember once going out with 38 still boards, and I drove around New York State and visited dealers and I went out with 38. I came home with 40 40 snowboards because one guy given me to back that he'd bought and said, This is a joke . from NPR It's how I built this show about innovators, entrepreneurs and idealists and stories behind the movements. I'm Guy Raz and on Today show how Jake Carpenter turned a childhood novelty toy into one of the biggest winter sports in the world from it built Burton snowboards. Eso imagine being on an airplane in 1977. You sit down, you strike up a conversation with the guy sitting next to you.

Fig. 4. Summary example with correct predictions in green, false negatives in red and false positive sentences in blue.

[3] Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan, "A baseline analysis for podcast abstractive summarization," PodRecs: The Workshop on Podcast Recommendations, September 25 2020.

[4] Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, and Ling Fan, "A two-phase approach for abstractive podcast summarization," *arXiv preprint arXiv:2011.08291*, 2020.

[5] Damiano Spina, Johanne R. Trippas, Lawrence Cavedon, and Mark Sanderson, "Extracting audio summaries to support effective spoken document search," *Journal of the Association for Information Science and Technology*, September 2017.

[6] Rosie Jones, "The new TREC track on podcast search and summarization," in *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xi'an, China, July 25-30 2020.

[7] Sadaoki Furui, T Kikuichi, Yousuke Shinnaka, and Chiori Hori, "Speech-to-speech and speech to text summarization," in *First International workshop on Language Understanding and Agents for Real World Interaction*, Sapporo, Japan, July 13 2003.

[8] Shasha Xie, Hui Lin, and Yang Liu, "Semi-supervised extractive speech summarization via co-training algorithm," in *11th Annual Conference of the International Speech Communication Association*, Makuhari, Japan, September 26-30 2010.

[9] Nikhil Garg, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani-Tür, "Clusterrank: a graph based method for meeting summarization," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[10] Yuxiang Wu and Baotian Hu, "Learning to extract coherent summary via deep reinforcement learning," in *32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2-7 2018.

[11] Tzu-En Liu, Shih-Hung Liu, and Berlin Chen, "A hierarchical neural summarization framework for spoken documents," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, United Kingdom, May 12-17 2019.

[12] Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, July 2019.

[13] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *31st AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 4-9 2017.

[14] Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou, "Neural latent extractive document summarization," in *2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, October 31–November 4 2018.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, June 2-7 2019.

[16] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang, "Extractive summarization as text matching," in *58th Annual Meeting of the Association for Computational Linguistics*, Seattle, WA, USA, July 5-10 2020.

[17] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching machines to read and comprehend," in *28th International Conference on Neural Information Processing Systems - Volume 1*, Montreal, QC, Canada, December 7-12 2015.

[18] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *Workshop on Text Summarization Branches Out*, Barcelona, Spain, July 25-26 2004.

[19] Ann Clifton, Aasish Pappu, Sravana Reddy, Yongze Yu, Jussi Karlgren, Ben Carterette, and Rosie Jones, "The Spotify podcasts dataset," 2020.