Alma Mater Studiorum · University of Bologna

Abstractive Summarization of Podcast Transcripts with BART using Semantic Self-segmentation [1]

Natural Language Processing

Giuseppe Boezio
Università di Bologna
giuseppe.boezio@studio.unibo.it

Simone Montali
Università di Bologna
simone.montali@studio.unibo.it

Giuseppe Murro
Università di Bologna
giuseppe.murro@studio.unibo.it

June 22, 2022

[1] The code for the project is publicly available on GitHub

Contents

Executive summary

This paper shows the work regarding our abstractive summarizer for the Podcast Summarization Challenge in TREC 2020. The submission focuses on the whole end-to-end process of summarization, and not just on the training of a model. More specifically, several steps were needed to reach good summarization performances, both on the target and the data: a thorough cleaning of the targets (podcast descriptions) using IDF and regular expressions, parallelized with chunk selection techniques for the input data. These chunks were obtained through semantic segmentation, then classified to predict their importance. It is also crucial to underline how the dataset needed a substantial reduction, as lots of datapoints were corrupted by copy-pasted descriptions, speech-to-text errors, and empty targets. These techniques, together with the usage of transfer learning methods exploiting BART (currently the state-of-the-art for summarization techniques), allowed the submission to reach impactful results in the highly complex task of summarization. In fact, most of the metrics obtained increases up to 30% with respect to the baseline considered. This proves that tackling this task is not impossible and the usage of these systems will be seen more and more in the future of podcasts, news, and media.

1 Introduction

Podcasts are a large and growing repository of spoken audio. As an audio format, podcasts are more varied in style and production type than broadcast news, contain more genres than typically studied in video data, and are more varied in style and format than previous corpora of conversations. When transcribed with automatic speech recognition they represent a noisy yet fascinating collection of documents, which can be studied through the lens of natural language processing, information retrieval, and linguistics. Paired with the audio files, they are also a resource for speech processing and the study of paralinguistic, sociolinguistic, and acoustic aspects of the domain. For a faster access to the content of a podcast, it can be useful to produce a summary of it before the user decides to play the podcast.

A particular kind of summarization is abstractive summarization, which consists of producing a summary which is not a subset of sentences of the original transcript (extractive summarization) but a text generated according to a language model. Recently, deep learning methods have shown much success and become the standard approach for various natural language processing (NLP) tasks, including abstractive text summarisation. Especially with the introduction of attention mechanism, pre-training large transformer-based language model (such as BART [5]) before fine-tuning the model on the target task have achieved remarkable performance.

2 Problem definition

2.1 TREC Podcast Track 2020

This project aims at producing a good abstractive summary of podcasts transcripts, obtained from the Spotify Podcast Dataset [1]. It is the first large-scale set of podcasts with transcripts that has been released publicly, with over 100,000 transcribed podcast episodes comprised of raw audio files, their transcripts and metadata. The transcription is provided by Google Cloud Platform's Speech-to-Text API[1].

[1] https://cloud.google.com/speech-to-text

Podcasts as a format, and their transcriptions, are varied with respect to category, structure, speakers, length, etc., and therefore represent a challenge for current summarization models.

The data was initially provided in the context of the TREC Podcast Track 2020 [3], after which they have been made available for more general research use. In the summarization task proposed by the conference, given a podcast episode, its audio, and transcription, the goal is to return a short text snippet which accurately conveys the content of the podcast. While no ground truth summaries were provided for the TREC 2020 task, the episode descriptions written by the podcast creators serve as proxies for summaries, and are used for training the supervised models submitted to the conference.

2.2 Transfer learning with pre-trained models: BART

Most of the papers presented during the TREC Podcast Track 2020 conference involve BART [5] pre-trained models [11] [7] [4] [9] [13]. BART was presented in 2020, and it currently represents the state-of-the-art for many NLP tasks, including text summarization. The model is based on a denoising autoencoder for pretraining sequence-to-sequence models. This is a modification of the standard autoencoder structure, done in order to prevent the network learning the identity function: it often happens that autoencoders just learn the training data, having an output equal to the input, without performing any useful learning. Denoising autoencoders corrupt the data on purpose, adding an arbitrary noising function and learning a model to reconstruct the original text. BART uses a standard transformer-based seq2seq architecture, with a bidirectional encoder and a left-to-right decoder. This means that the encoder's attention mask is fully visible, while the decoder's attention mask is causal. In the original paper, several noising approaches were tried, finally resulting in a double approach to noising: a random shuffling of the order of sentences, and a novel in-filling scheme, where spans of text are replaced with a single mask token.

The pre-training task has two stages: first, text is corrupted, then a seq2seq model is learned to reconstruct the original text. For this task, the negative log-likelihood of the original document is optimized. The fact that BART is only involved in the pre-training of a model, makes it incredibly flexible and applicable to different tasks. The original paper cites sequence classification tasks, token classification tasks, sequence generation and machine translation. This list is, though, non-exclusive, as shown in this experiment. Additionally, it is shown that BART outperformed the state-of-the-art at the time (BERT-based) by roughly 6.0 points on all ROUGE metrics [6].

2.3 Related work

The Spotify Podcast Dataset presents several challenges [1]:

1. the input documents are automatically transcribed, and thus subject to speech recognition errors
2. disfluencies and redundancies are abundant in spoken text; its information density is often low when compared to written text. The podcasts are of various genres: monologue, interview, conversation, debate, documentary, etc. and transcription is more challenging and noisier
3. the documents are significantly longer than typical summarization data

Despite no participant of the TREC Podcast Track 2020 used the audio data for summary generation, most of them made use of a two-step approach for dealing with longer inputs by extracting a portion

of the transcript that is highly relevant. The extracted portion of the transcripts is the input for an abstractive summarizer.

This first step is necessary because the input exceeds the limit imposed by many neural abstractive models. Indeed a podcast transcript contains on average 80 segments and 5743 tokens, while BART has a limited length of the input sequence up to 1024 tokens, due to the absolute position used by the positional embedding. Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. Manakul P. et al. [7] also tried to expand the positional embedding of the BART model to accommodate sequences longer than 1024, but it has been shown that it is less efficient than the vanilla BART model.

Thus, alternative methods have been investigated to filtering out redundant or less informative sentences in the input transcriptions. The best system that Manakul P. et al. [7] submitted consists of using the hierarchical model to filter transcriptions at both training time and test time. It uses a sentence-level attention score to obtain a measure of the importance of each sentence.

A similar work done by Zheng C. et al. [13] 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. The selection is ROUGE-based, meaning that they calculate the ROUGE score for each sentence with the input document. The intuition is that a sentence with a higher ROUGE score serves as an excellent summary of the input document, which should capture the important information.

Alternatively Song K. e al. [11] investigated a segment-rather than sentence-based extraction to select salient segments from the beginning and end of a transcript. Each segment correspond to 30 seconds of audio. They used 33 segments from the beginning and 7 segments from the end of each transcript to be the candidate segments and they trained a classifier to predict whether the segment is salient or not.

The latter method is interesting, but it may split highly related sentences into different segments. To address this problem a solution could be the semantic self-segmentation (Se3) approach for long document summarization, proposed by Moro G. et al. [2]. Concretely, this method segments long texts into content-wise chunks containing semantically similar sentences.

3 System Description

Our method is inspired by the aforementioned solutions and tries to combine their strengths. We particularly emphasize content selection, in which an extractive module is developed to select salient chunks from the transcript, which serve as the input to an abstractive summarizer. The latter utilizes a BART model, that employs an encoder-decoder architecture. In this way we optimize the usage of podcast transcripts without exploiting their raw audio. An extensive pre-processing on the creator-provided descriptions is performed selecting a subset of the corpus that is suitable for the training supervised model. The Figure 3.1 summarizes the steps involved by our solution. A deeper explanation of the extractive stage is described in the following sections.

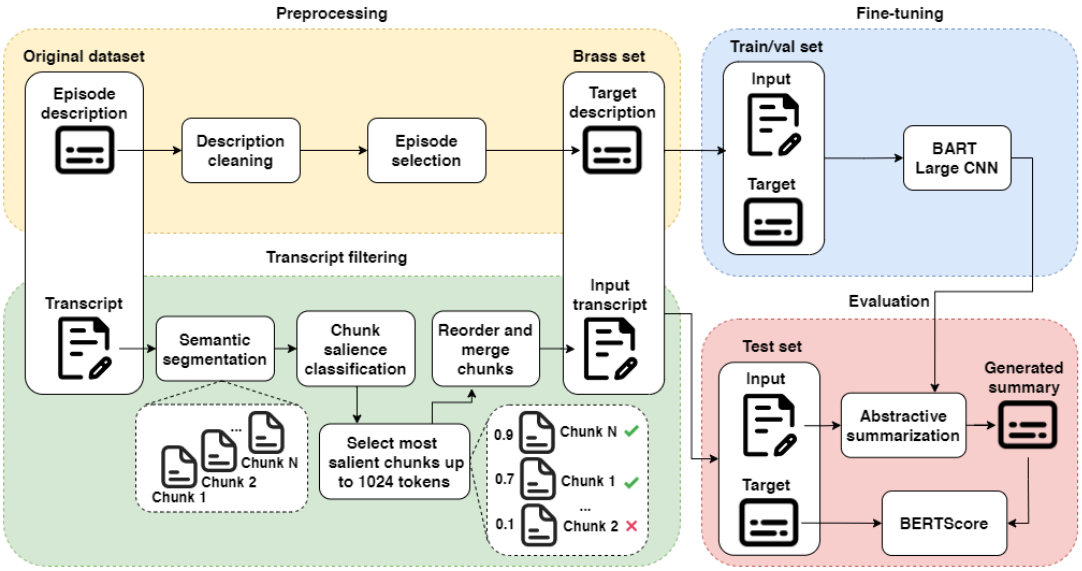

Figure 3.1: Abstract scheme of the proposed solution for the summarization task

3.1 Pre-processing

To train a supervised model on the Spotify Podcast Dataset, we consider the creator-generated descriptions as our reference summaries. However, these descriptions vary widely in quality and are not always intended to act as summaries of the episode content, reflecting the different uses creators have for descriptions and the different genres of podcast in the sample. We first cleaned the episode descriptions in order to filter out useless sentences and symbols, then we established a subset that is more appropriate as a ground truth eliminating some of the episodes from the dataset.

3.1.1 Episode description cleaning

A gold dataset of 150 episodes is also available, being composed by 6 set of summaries for each episode (900 document-summary-grade triplets) that were graded on the Bad/Fair/Good/Excellent scale (0-3). Before starting the cleaning process, we merged this *gold dataset* with the dataset we are going to clean, and the best summary of each episode will be considered. Then we strive to enhance the quality of creator descriptions using heuristics to identify sentences that contain improper content. The steps are the following:

1. removing the content after "—" that usually is a sponsorship or a boilerplate (e.g. *"— This episode is sponsored by ..."*, *"— Send in a voice message"*)
2. remove sentences that contain URLs, @mentions and email addresses in the episode descriptions
3. remove tokens corresponding to emojis
4. remove the least significant sentences

The first three steps are easily achievable using regular expressions to remove content that is more schematic (i.e. links, usernames, emojis or hashtags). The latter requires a more flexible approach,

computing a salience score for each sentence of the description by summing over word IDF scores, as done in [11]. A low IDF score indicates the word frequently appears in other episodes, and thus is uninformative. We remove sentences if their salience scores are lower than an empirical threshold (σ = 3.6). This allows to filter out most of the confusing parts of the descriptions, which, being often repeated, highly limited the performances. Examples of the application of the cleaning steps on some episode descriptions are shown in Table A.1 of Appendix A.

3.1.2 Episode selection

In order to select a subset of the training set that is proper to achieve good summarization results, we filtered out some descriptions using three heuristics shown in the original paper [3]. We removed descriptions that are (a) too long (greater than 750 characters) or short (less than 20 characters); (b) too similar to its show description (no new info); (c) too similar to other descriptions in the same show (cut-and-paste or template). The podcast episodes should be restricted to the English language, but they cover a range of geographical regions and there is a number of non-English podcasts in the dataset. Therefore they have been removed too. The remaining episodes represent the so called **Brass set**. The organizers find that about a third of the entire set contains less useful descriptions, but since we cleaned them before, the selection stage results in 77,041 training examples, which are more than the ones obtained by the organizers (66,245). The number of episode removed after applying each selection heuristic and some examples of removed descriptions are shown in Tables A.2, A.3, A.4 of Appendix A.

3.2 Transcript filtering

3.2.1 Semantic segmentation

Transcripts are very long compared to other kind of documents to summarize; therefore, as discussed in [2] it seems reasonable to split them in different sequences of semantically related sentences called chunks. This semantic self-segmentation (Se3) approach turns out to be very useful in the following steps to reduce the length of the transcripts, only keeping the ones which are the most promising to provide relevant information for the summarization. To compute the similarity, Se3 first creates the sentence embeddings using a fine-tuned sentence transformer called Sentence-BERT [8]. Afterward, the semantic similarity is calculated between the current sentence and each sentence within the previous chunk and the next chuck. The similarities are averaged per chunk and compared.

3.2.2 Chunk salience classification

To reduce the size of the transcripts we have decided to only keep the most salient chunks using a classifier.

3.2.2.1 Chunks classifier training

Ideally a salient chunk should have a relatively high similarity score with the creator-provided description. The main problem is that, at runtime, the description is not available. To tackle this problem we have trained a deep neural network to discriminate between useful and useless chunk setting a binary classification problem. As training set, we used a cleaned version of the *gold dataset*, composed by 141 examples which had descriptions graded as Good/Excellent. The input features for the classifier

are generated extracting the chunks from the transcript and creating a representation of each chunk averaging over the sentence embeddings yielded with Sentence-BERT [8]. The ground-truth segment labels are derived by comparing segments with creator descriptions. We calculate the ROUGE-L f1-score for a chunk against the creator description. A chunk is labelled as positive if the score is greater than a threshold (τ = 0.2), otherwise negative, and done in [11]. The positive-to-negative ratio is 1:10 among candidate chunks.

3.2.2.2 Salient chunks selection

At inference time, we use the chunk classifier to assign a *salience score* to each chunk. The chunks are sorted in a decreasing order according to the score and only the first ones whose overall number of tokens does not exceed 1024 are taken. Chunks are considered as basic atomic units, therefore they are used with all their sentences or they are not used. In this case tokens are not words but subsequences of them using Byte Pair Encoding. This is important to provide a compatibility with the BART Transformer which uses this kind of tokenization technique.

3.2.2.3 Reorder and merge chunks

To exploit on the positions of tokens, the selected chunks are sorted according to their initial position and then concatenated. At this point the *brass set* is composed by smaller and more meaningful version of the transcripts and cleaned version of their corresponding descriptions.

4 Experimental setup and results

4.1 Sentence boundary disambiguation during pre-processing

Some of pre-processing steps described in 3.1.1 and 3.2.1 require the division of text into sentences. It might, at first, seem like a trivial task, but it hides a number of pitfalls that might turn out to be dangerous. For example, a naïve sentence tokenizer based on a regular expression, might wrongly tokenize the sentence *"Mr. Burns didn't care. Today's injury is n.95 at the power plant."*, recognizing the dots as sentence terminators. During the experiments shown in this report, a state-of-the-art sentence tokenizer has been chosen. PySBD [10] has been shown to outperform most open-source libraries, such as spaCy, stanza or NLTK. Its API is standard and simple to use, only requiring the instantiation of a pysbd.Segmenter and its usage.

4.2 Fine-tuning parameters

The initial BART model's starting point was `bart-large-cnn` [1], pre-trained on English language, and fine-tuned on CNN Daily Mail, a large collection of text-summary pairs. The training has been performed on a rented cloud machine [2] with 4 x NVIDIA RTX 3090 GPU of 24GB memory and 128 GB of RAM. We split the filtered brass set into train/dev sets of 69,336/7,705 episodes. The BART baseline was then fine-tuned for 3 epochs on filtered transcripts as input (rather than the first 1024 tokens of the transcript). The parameters used for the training are `batch_size=4`, and an Adam optimizer with `learning_rate=2e-5`

[1] https://huggingface.co/facebook/bart-large-cnn
[2] https://vast.ai/console/create/

and `weight_decay_rate=0.01`. The optimization is performed on the internal loss computed by BART. The final model, that we call `bart-large-finetuned-filtered-spotify-podcast-summ` (and from now on we name `bart-large-finetuned-filtered` for brevity), has been uploaded on the Huggingface hub [3].

4.3 Training history

We used the ROUGE F1-scores metric to check on the validation set the progress of the model during the training. The Figure 4.1 shows the evolution of the loss and the metric results at each epoch. Although the loss on the validation set decreases very slowly, the ROUGE chart , which refers only to validation data, indicate that the model is effectively improving.

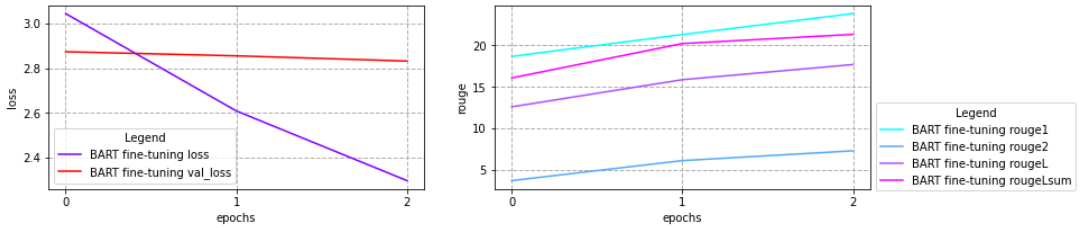

Figure 4.1: Evolution of loss and ROUGE F1-score metrics during the training

4.4 Evaluation

The test set consists of 1,027 episodes. It is provided in the Spotify Podcast Dataset separately from the training set, so we can use it to evaluate our model. Only 1025 have been used because two of them did not contain an episode description. The pipeline to follow is the same of the training phase. The hyperparameters used by BART during the generation are the same defined in [11]:

- `length_penalty=2.0`: it penalizes the log probability of a summary by its length, promotes the generation of long summaries
- `num_beams=4`: number of predicted tokens instead of a single token greedy strategy
- `no_repeat_ngram_size=3`: a trigram cannot occur more than once in a summary
- `min_length=39`: minimum length of the summary
- `max_length=250`: maximum length of the summary

To evaluate the results two metrics have been chosen: ROUGE and BERTscore. ROUGE was used also in the TREC 2020 task to compare the generated summaries against the creator descriptions as the reference. It points out to be a meaningful metric for automated evaluation [1] even if it is a syntactic metric. It relies only on matching n-grams between the generated text and the reference text. This means that if the words in the n-gram do not exactly match then they are not counted towards the final score even though the word might mean something very similar in that context. Abstractive summarization models aim to capture the salient parts of the input text and generate a summary, where a strength of the model is that

[3] https://huggingface.co/gmurro/bart-large-finetuned-filtered-spotify-podcast-summ

it can formulate or phrase the summarized text in a different way than the reference without losing the meaning of the text. In these cases ROUGE score unfairly disadvantages such summaries.

For this reason, BERTScore [12] has been chosen as semantic metric to evaluate the results. It computes similarity score between each token in candidate text and the reference text by using a contextual word embedding. The original BERTScore paper showed that BERTScore correlates well with human judgment on sentence-level and system-level evaluation, but this depends on the model and language pair selected. Previous work on similarity measures demonstrated that rare words can be more indicative for sentence similarity than common word. Therefore we have tried a variant of BERTscore which uses a weighting based on IDF scores computed on reference sentences. The IDF scores remain the same for all systems evaluated on a specific test set and this is the reason which allow us to use it on predictions of different models.

Calculating the BERTScore metric involves downloading the transformer-based model that is used to compute the contextual word embeddings. The state-of-the-art model for the English language is `roberta-large` and we use it.

4.4.1 Results

ROUGE has been computed using the three standard measure precision, recall, and F1 score. Results about ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-L Sum are summarized in the table 4.1. BERTScore has been computed about precision, recall and f1 score using both the standard and the IDF weighting version, as shown in table 4.2. We use those results to compare our model with the pretrained version of BART (`bart-large-cnn`) to quantify how well our model performs with respect to one of the baseline defined in [1]. The BERTscore is computed for each pair of candidate and reference candidate. For this reason the metric shown in the table corresponds to the mean of the aforementioned scores.

Metric	Model	Precision	Recall	F1 Score
ROUGE-1	bart-large-cnn	0.2357	0.2125	0.1917
	bart-large-finetuned	0.3250	0.2315	0.2370
ROUGE-2	bart-large-cnn	0.0449	0.0447	0.0379
	bart-large-finetuned	0.0884	0.0677	0.0670
ROUGE-L	bart-large-cnn	0.1467	0.1408	0.1223
	bart-large-finetuned	0.2160	0.1634	0.1621
ROUGE-L Sum	bart-large-cnn	0.2088	0.1855	0.1684
	bart-large-finetuned	0.2846	0.2003	0.2058

Table 4.1: Comparison of scores among models with ROUGE metrics

Model	Precision		Recall		F1 Score	
IDF weighting	Yes	No	Yes	No	Yes	No
bart-large-cnn	0.8103	0.8317	0.7941	0.8121	0.8018	0.8214
bart-large-finetuned	0.8401	0.8631	0.8093	0.8279	0.8240	0.8447

Table 4.2: Comparison of models performance with BERTScore

5 Analysis of results

Analysing the results illustrated in section 4.4.1, we can argue that all the outcomes produced by ROUGE metrics computed on our fine-tuned model are higher than the ones on the baseline. Generally speaking, the ROUGE-2 (which compares matching bi-grams) is really low with respect to the others. This can be justified by the fact that creator-provided description sometimes are even less accurate than prediction in summarizing content and that it's likely to have different bi-grams between descriptions and predictions even if they are semantically similar.

On the other hand, the results obtained with BERTScore demonstrate that the meaning of the predictions are highly correlated with the description context. Considering the F1 measure, which is the harmonic mean of precision and recall, our model reach a score of 82.40% with the IDF weighting, with an absolute increase of +2.22 over the result gained with the `bart-large-cnn`.

As illustrative example we show a good and a bad summary comparing creator descriptions and predictions. As we can see from Table B.1 of Appendix B, the summaries generated by the `bart-large-finetuned` turn out to be meaningful because they do not contain advertisement or non readable information such as URLs and mails contained in the creator descriptions. At first glance, it is clear that, because of the length constraints, the predicted summary tends to be too short compared to the amount of information contained in the original summary. This is, on the one hand an advantage because it allows to focus on important aspects of the speech. Indeed the objective for the task was "to provide a short text summary that a user might read when deciding whether to listen to a podcast. The summary should accurately convey the content of the podcast, be human-readable, and be short enough to be quickly read on a smartphone screen". Moreover our predictions point out to be better than the `bart-large-cnn` model ones because they use a more formal language and indirect speech instead of using just a part of it as contained in the transcript. In the second example, our predicted summary is not so good because the discourse is not fluent, there are repeated expressions like "In this episode" and "we discuss".

An important step of the pipeline used in our system that could dramatically affect the generation of good summaries is the transcript filtering. Indeed the chunk salience classifier is empowered of filter out chunks, and sometimes some of them could be meaningful. Without them the abstractive summarizer is not able to reconstruct that information in the summary.

The performance of the summaries generation are also related to the model used. We rely on `bart-large` which is highly effective but at same time very complex to train due to the high number of parameters (400 millions). With our limited computation resources, we was able to perform only 3 epochs, but maybe a further training could increase the robustness of the model.

6 Conclusion and future works

Extractive models suffer from errors caused by speech recognition or the natural disfluency of spoken language, whereas abstractive models, like BART (the model we used), seem to be more able to generalize over these errors and generate relatively fluent written language.

Further, we caution that it is challenging to generate abstractive summaries that are fully accurate to the content of the podcast. Even humans may not be able to spot some subtle errors without a careful comparison of system abstracts and transcripts. Our method is effective at capturing the entirety of the long input sequences and sometimes outperforms the descriptions made by the podcast creators themselves.

A possible future development could be an ablation study to understand whether the transcript filtering increase the performance of our model. In order to do that, we could fine-tune `bart-large-cnn` on the first 1024 tokens from the transcripts an use is as new baseline. Moreover, it could be useful try other chunk salience classifier using a hierarchical attention mechanism on all chunks.

In conclusion, areas such as entity recognition and discourse analysis should be investigated for future work to extract semantic information which could be useful to detect relevant parts in the transcript.

Bibliography

[1] Ann Clifton et al. "100,000 Podcasts: A Spoken English Document Corpus". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5903–5917. DOI: 10.18653/v1/2020.coling-main.519. URL: https://aclanthology.org/2020.coling-main.519.

[2] Luca Ragazzi Gianluca Moro. "Semantic Self-segmentation for Abstractive Summarization of Long Legal Documents in Low-resource Regimes". In: *AAAI* (2022).

[3] 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: *CoRR* abs/2103.15953 (2021). arXiv: 2103.15953. URL: https://arxiv.org/abs/2103.15953.

[4] Hannes Karlbom and Ann Clifton. "Abstractive Podcast Summarization using BART with Longformer attention". In: ().

[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*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://aclanthology.org/2020.acl-main.703.

[6] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[7] Potsawee Manakul and Mark Gales. "CUED_speech at TREC 2020 Podcast Summarisation Track". In: *arXiv preprint arXiv:2012.02535* (2020).

[8] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. DOI: 10.48550/ARXIV.1908.10084. URL: https://arxiv.org/abs/1908.10084.

[9] Rezvaneh Rezapour, Sravana Reddy, Ann Clifton, and Rosie Jones. "Spotify at TREC 2020: Genre-Aware Abstractive Podcast Summarization". In: *arXiv preprint arXiv:2104.03343* (2021).

[10] Nipun Sadvilkar and Mark Neumann. *PySBD: Pragmatic Sentence Boundary Disambiguation*. 2020. DOI: 10.48550/ARXIV.2010.09657. URL: https://arxiv.org/abs/2010.09657.

[11] Kaiqiang Song, Chen Li, Xiaoyang Wang, Dong Yu, and Fei Liu. "Automatic summarization of open-domain podcast episodes". In: *arXiv preprint arXiv:2011.04132* (2020).

[12] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: *CoRR* abs/1904.09675 (2019). arXiv: 1904.09675. URL: http://arxiv.org/abs/1904.09675.

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

A Appendix: Pre-processing

	Episode description
Original	Danielle and Jessi could talk your ears off when it comes to this topic. Episode 004 is all about their skincare routines, products they love, and tips and tricks for feeling radiant and confident in your own skin. Follow them @basicallyorganicpodcast (and @jessimechler @itsdaniellebridges) for tags of all the brands they're currently loving! Rate and subscribe!! — Support this podcast: https://anchor.fm/basicallyorganicpodcast/support
After cleaning step 1, 2, 3	Danielle and Jessi could talk your ears off when it comes to this topic. Episode 004 is all about their skincare routines, products they love, and tips and tricks for feeling radiant and confident in your own skin. Rate and subscribe!!
After cleaning step 4	Danielle and Jessi could talk your ears off when it comes to this topic. Episode 004 is all about their skincare routines, products they love, and tips and tricks for feeling radiant and confident in your own skin.

	Episode description
Original	If you like ASMR you will love this White Noise Machine on Amazon! Tap here to check it out! If you enjoyed this make sure to give us a 5 star rating! — This episode is sponsored by · Anchor: The easiest way to make a podcast. https://anchor.fm/app Support this podcast: https://anchor.fm/AdamDino/support
After cleaning step 1, 2, 3	If you like ASMR you will love this White Noise Machine on Amazon! Tap here to check it out! If you enjoyed this make sure to give us a 5 star rating!
After cleaning step 4	If you like ASMR you will love this White Noise Machine on Amazon! Tap here to check it out!

Table A.1: Example of cleaning applied to two episode description

Selection heuristics	Dataset size	Cumulative reduction rate
Original	105,153	0%
Length	88,045	16.27%
Similarity with show description	86,448	17,79%
Similarity with other descriptions	77,415	24.86%
Non-english descriptions	77,041	25.19%

Table A.2: Number of episode removed from the Spotify Podcast Dataset after applying each selection heuristic

Episode description	Show description	Similarity score
Life and fashion all packed into a panini	Life and fashion all packed into a panini	99%
Today, three of the worlds straightest males have gathered to talk about the straightest things.	On the Wearings-Socks Podcast, three of the worlds straightest males have gathered to talk about the straightest things.	87%
We look back at the case of the Maryland Court vs Adnan Syed and tell you what we think really happened on that fateful day in 1999	We're out here doing a podcast about the Serial Podcast that is based off of the State of Maryland v. Adnan Syed case that happened back in 1999.	52%

Table A.3: Some examples of episodes removed from the Spotify Podcast Dataset since too similar to the show description

Episode description	Other episode description	Similarity score
A real banger, one for the ages	This is a banger, one for the ages	84%
In this exercise, drift off to sleep while listening to nature sounds.	In the full version of this exercise, focus on muscle relaxation while listening to nature sounds.	72%
In this episode, Shawna and Larry talk with Jason Lobmeyer about helpful tips on how to be a great CCV kids coach.	In this episode, Shawna and Larry talk with George Mang about helpful tips on how deal with behavioral issues in a kids experience.	71%

Table A.4: Some examples of episodes removed from the Spotify Podcast Dataset since too similar to others episode descriptions in the same show

B Appendix: Generated summaries

Creator description	Your host Susie Roloff sits down to chat with Katie Backa a well known tournament angler from Texas. Patreon- https://www.patreon.com/paddlenfin Podcast Website- www.paddlenfin.com YouTube- https://www.youtube.com/paddlenfin Email- paddlenfin@gmail.com Social Media- @paddlenfin Rocktown paddlesports - rocktownadventures.com Loveland Canoe Kayak- https://www.lovelandcanoe.com Hammered Lures- https://hammered-lures.myshopify.com Fish Mob Lures-https://www.facebook.com/officialfishmoblures/ TRC Covers-https://trccovers.com JigMasters Jigs- https://jigmasters.com Ketch Products-https://ketchproducts.com Recycled Plastics Recycling Program - Mail to: 316 Pinewood Dr. Camp Hill,PA 17011 — This episode is sponsored by · Anchor: The easiest way to make a podcast. https://anchor.fm/app
Predicted summary bart-large-finetuned	In this episode we talk about women in the sport of fishing. We talk about what it's like to be a female angler and how it's different than being a male angler, and how to deal with the expectations that come with being a woman angler.
Predicted summary bart-large-cnn	It should be an expected thing. Why is there this perception that like, oh, yeah, you're a woman you have to have kids once you get married. I think I'm still figuring, you know myself out and where I'm going to go with everything but I've definitely got a better idea. Sometimes it's hard to not want to always go with the flow.

Creator description	'Capitalism has placed a crushing burden on women's shoulders.' In this piece by one of the towering figures of the working-women's movement, Kollontai shows how the old family unit is no longer useful and how communism provides the necessities to lessen the burden of women and we discuss how 100 years later working women are still staggered by the triple load of being a mother, housewife and worker as well as how the capitalist world is still playing catch up to the collossal social changes introduced in the Soviet Union. This episode is part of our series on Marxist feminists. As Thomas Sankara said, "Women hold up the other half of the sky." Red Book Club recognizes that including the voices of nonmen in our studies is not a niche activity, but is in fact an essential step in gaining the most comprehensive view of the material conditions of the past and present as we possibly can; so we've planned this series to amplify the ideas that nonmen have been bringing to the conversation for centuries. From Federici's analysis of women as a means of primitive accumulation to Luxemburg's essay of the benefits of revolution vs the impossibility of reform
Predicted summary bart-large-finetuned	In this episode, we discuss The Reformation of the Family Unit and the Idea of the Traditional Family Unit. In this episode we discuss the ideas of the family unit, the role of women in society, and the reasons why the old family unit social model is no longer useful.
Predicted summary bart-large-cnn	Natalie: I'm really excited to do this work today. And the Reformation of the idea of the family unit. Our first section here is women's role in reproduction, its effect upon the family.

Table B.1: Example of good and bad predicted summaries