Glasgow Representation and Information Learning Lab (GRILL) at TREC 2020 Podcasts Track

Paul Owoicho
School of Computing Science
University of Glasgow
paulowoicho@gmail.com

Jeff Dalton
School of Computing Science
University of Glasgow
jeff.dalton@glasgow.ac.uk

ABSTRACT

In this paper, we discuss our participation in the Summarization Task of the TREC 2020 Podcasts Track. Our submission consists of summaries generated by (i) an abstractive model based on fine-tuning T5 on the Spotify Podcasts Dataset, (ii) an ensemble model where the first 15 sentences from the podcast transcript are extracted and passed as input to a fine-tuned T5 model, and (iii) another ensemble model where we use a SpanBERT and K-Means pipeline to extract the 15 most important sentences from the podcast transcript and pass them as input to a fine-tuned T5 model. Official results demonstrate that out of 179 evaluated summaries, our best performing model (ii) generated 42 good-quality summaries - on par with the average across all other submissions. This provides evidence that focusing on the first part of the podcast episode is a strong baseline for podcast summarization.

1 INTRODUCTION

Although podcasts are a fairly new form of audio media, its popularity and rate of consumption has rapidly grown over the years [4]. To better understand the content within podcasts, the Summarization Task of the TREC 2020 Podcasts Track asked participants to summarise a podcast episode using its audio and/or transcription. Returned summaries were to be grammatically complete and capture the most important parts of the podcast episode. In our participation, we focused on generating summaries for podcast episodes using their transcripts. Our three submissions were generated using summarization models that leveraged a T5 model fine-tuned on the Spotify Podcasts Dataset. For two of our submissions, we first ran each transcript through an extractive pipeline before generating the final summaries with T5. Further details of the task, datasets, baselines, and evaluation methods used are discussed in [1].

2 MODELS AND IMPLEMENTATION

For our experiments, we studied extractive and abstractive summarization methods. In addition to the official track baselines, the models we experimented with were:

BERT Extractive Summarizer

In [3], Miller proposes an approach to extractive summarization where a Transformer model is first used to generate text embeddings from the sentences in a source text. Next, these embeddings are passed as input to a K-Means clustering model. The output summary from this pipeline then consists of sentences closest to the cluster centroids found by the K-Means model. This technique was proposed for use in a lecture summarization service where end-users could summarize the content of their lecture transcripts. As spoken documents, lectures bear similarities with podcasts, supporting our intuition that Miller's approach was viable for the podcasts dataset. In building this model, we closely followed the project's open-sourced code on Github [1] and leveraged its available APIs. Hyperparameters we could change included the number of sentences (also K-means clusters) in the final summary, the Transformer model used for generating text embeddings, and what layer of the Transformer model to generate the embeddings from. We set the model's output to two sentences in order to standardise the comparison between it and the results of the track's baseline approaches. Furthermore, even though the original pipeline was based on the BERT Transformer model, we examined the results of using SpanBERT in our experiments. We generated text embeddings using the second-to-last layer of our Transformer models and aggregated them using a mean pooling method.

T5

In [5], Raffel et al. propose T5 based on the results of a large scale evaluation of transfer learning techniques used for NLP tasks. T5 achieves state-of-the-art performance on many NLP tasks including abstractive summarisation. T5 also represents an improvement over the abstractive baseline model for the Summarization task. Summaries generated by the model on the CNN/DailyMail Dataset can be found in [5].

PEGASUS

Zhang et al. introduce PEGASUS in [6] due to the lack of Transformer based models with an abstractive text summarisation objective. PEGASUS is a large encoder-decoder model where indicative sentences in a piece of text are first removed and then regenerated from the remaining sentences. Despite being fine-tuned on a small number of samples, PEGASUS achieves state-of-the-art performance as measured by ROUGE scores on 12 summarization tasks. Some sample summaries generated with PEGASUS on a wide range of datasets can be found in [6].

Our implementation of T5 and PEGASUS followed a similar approach. We loaded T5 (*t5-base*) and PEGASUS (*google/pegasus-cnn_dailymail*) directly from the HuggingFace library. Both models were fine-tuned on the official training set of 65K podcast episodes using their creator descriptions as training summaries. Important hyperparameters used to train the models were:

- *early stopping*: **True**. This was necessary to prevent the models from overfitting the training summaries.
- *length penalty* **2.0**: This was necessary to discourage the models from generating long summaries.

[1] https://github.com/dmmiller612/bert-extractive-summarizer

- *max length*: **150**: This was a stipulation of the maximum length of our models' summaries. We decided on 150 so that the models could generate meaningful summaries under the official 200-character limit.
- *min length*: **30**: This was a stipulation of the minimum length of our models' summaries. We hypothesised that 30 characters was the minimum necessary to generate a meaningful summary.
- *no repeat ngram size*: **3**: This was necessary to discourage the models from generating summaries where it repeated itself. We hypothesised that a trigram was a reasonable threshold for this.
- *num beams*: **5**: In beam search, this hyperparameter determines the number of steps that are kept track of while a model generates a sequence. Typically, larger values generate better summaries at the expense of speed. We settled on 5 as this is typically used for text generation tasks.
- *learning rate*: **1e-4**: We kept the learning rate low enough to allow our models the chance to converge.
- *epochs*: **3**: We trained our models over 3 epochs. As our models were pre-trained, there was no need to fine-tune them over a longer epochs since we could leverage transfer learning.

We set our hyperparameter values to commonly used defaults without tuning or optimisations. This was because the provided creator summaries varied in quality. Moreover, a single piece of text can be summarised meaningfully in multiple ways. Thus, optimising hyperparameters to make the generated summaries resemble the creator summaries was not worthwhile. We validated our models using the remaining 35K episodes from the podcasts dataset.

Due to resource constraints, we truncated the input transcripts for both models before summary generation. For T5, this threshold was set to 7000 tokens, and for PEGASUS it was set to 1024. The undesired effect of this was that PEGASUS was only able to summarise the first few sentences of the podcast.

Ensemble Models

We experimented with combining our best extractive and abstractive techniques to form an ensemble summariser. These were implemented by first generating a summary with an extractive summariser and passing that as input to an abstractive summariser. Our approach was based on findings in [2].

Our two ensemble models are:

- **First 15 Sentences + T5**: For this model, we extract the first 15 sentences of a podcast transcript and pass that as input to a finetuned T5 model.
- **SpanBert + T5**: For this model, we use SpanBERT in the BERT Extractive Summariser pipeline to select the top 15 sentences from a podcast transcript and pass that as input to a finetuned T5 model.

We use 15 sentences for our experiments with the presumption that it would contain enough context for an abstractive summariser to generate an adequate, meaningful summary. This was due to the results we observed from a simple baseline extractive summariser where we used the first five sentences of a podcast as its summary. [2]

3 METHODOLOGY

Our candidate models were assessed using qualitative and quantitative methods, following which the top three performing models were used to generate summaries on the official evaluation set for our submissions.

For our assessments, we used the models to generate summaries on the same evaluation set of 150 podcasts used by the track coordinators to assess the task's baseline models [1]. Our models' summaries were evaluated using the ROUGE metric with the creator descriptions as reference. We also recruited six volunteers, each of whom rated 25 of the 150 summaries using episode titles, creator descriptions, and other metadata for context. Although it would have been better for podcast transcripts to be used for context, the transcripts were lengthy and volunteers were unwilling to read them. Volunteers' ratings were based on the official EGFB scale and demonstrated how well they thought the generated summary conveyed the gist of the podcast based on its metadata. When we compared our volunteers' ratings with those of the official baseline qualitative ratings for the track's baseline models, we observed a Cohen Kappa coefficient of 0.41, indicating a moderate agreement between the two sets of raters. This was due to the variation in the way the evaluations were carried out.

4 EXPERIMENTAL RESULTS

We observed that our participants preferred T5's summaries more often than those of the other models. Furthermore, we noticed that the extractive pipeline had the fewest good-quality summaries of all models, showing that abstractive methods outperform extractive methods for podcast summary generation. For the ensemble models, our assessors preferred the summaries generated by the First 15 Sentences + T5 model over those generated by the SpanBERT + T5 model. While this may be representative of the fact that the first parts of a podcast are its most important for summary generation, it also shows that an effective way to combine extractive and abstractive podcast summarisation methods is by first retrieving indicative sentences from the beginning of a podcast.

5 SUBMITTED RUNS

Our submitted runs were based on summaries generated by the following models:

- *T5*: This model generates a summary using the T5 model fine-tuned on the podcasts dataset.
- *First 15 Sentences + T5*: This model generates a summary of the first 15 sentences of a podcast using the T5 model fine-tuned on the podcasts dataset.
- *SpanBERT + T5*: This model generates a summary of the 15 most important sentences in a podcast (determined by the SpanBERT + KMeans pipeline) using the T5 model fine-tuned on the podcasts dataset.

[2] https://castos.com/podcast-structure/

Rating	**All Submissions (Avg. Count)**	**Finetuned T5**	**First 15 Sentences + Finetuned T5**	**SpanBERT + T5**
Bad	72	62	69	73
Fair	57	78	68	84
Good	**31**	27	29	14
Excellent	**19**	12	13	8

Table 1: Count of summaries from all submissions that received each rating

Question	**All Submissions (Avg. Count)**	**Finetuned T5**	**First 15 Sentences + Finetuned T5**	**SpanBERT + T5**
Q1	80	62	**85**	47
Q2	**49**	33	47	28
Q3	104	100	**109**	89
Q4	80	**96**	94	78
Q5	93	**95**	82	83
Q6	17	**7**	24	13
Q7	112	**152**	135	151
Q8	80	**106**	91	99

Table 2: Count of summaries from all submissions that received a ‘yes’ for each question

6 NIST EVALUATIONS

We focus our discussions on the results of our qualitative evaluations since the Summarisation Task was about summary quality and not affinity to the provided creator descriptions.

NIST evaluated participants’ returned summaries on the official EGFB scale and on 8 yes/no questions discussed in [1] and itemised below:

- **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)?*
- **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 endpoints?*

Over all evaluated summaries, all submissions seemed to struggle with generating good-quality ("Excellent" and "Good" ratings) summaries as shown in Table 1. Our SpanBERT + T5 model generally performed worse than the average of all submissions, while the First 15 Sentences + T5 model was our best performing submission.

On the 8 yes/no questions, all submissions seemed to perform reasonably well in generating well-structured, grammatically correct summaries and omitting redundant information as shown in Table 2. Our SpanBERT + T5 model however struggled with entity recognition/identification more often than other models, indicating that the 15 important sentences extracted with the extractive pipeline may have not always contained that information. Overall, while all models were fairly able to recognise the topic and format of podcasts, these are areas that need to be explored further in the future.

6.1 Behaviour Analysis

We also examined how our models performed on episodes where most submissions excelled and struggled in. We found that:

For episode *04Llj5QjDot46TEDGj7lbI*, 59% of submissions received an ’Excellent’ rating. Our T5 and SpanBERT + T5 models received a ’Fair’ rating, while the First 15 Sentences + T5 model received an ’Excellent’ rating. The summaries generated by all our models are shown in Table 3.

Of the three models, the First 15 Sentences + T5 model generates the most detailed summary that captures the most important parts of the episode. Unsurprisingly, this summary was also given a ’yes’ on all questions asides question 6. This provides evidence that effective summary generation for a podcast can be done just using the beginning of the podcast. This may also explain the discourse structure of podcasts in which all entities and topics in the podcast are introduced at the start.

For episode *45hMVOlYDHoX0E92qKjKTW*, over 93% of submissions were rated "Bad". Our models also struggled with this episode as they were also rated ’Bad’ and received a ’no’ on all questions that submissions were evaluated on. The summaries generated by our models are shown in Table 4.

Evidently, the summaries our models generated are incoherent and repetitive (in the case of the SpanBERT + T5 model), and, being a short episode, we also observed that the summaries generated

Model	Summary
T5	This episode is from a conversation I had with Vanessa. We talk about her life and how she was able to get out of the house without being seen again. You can find Vanessa on Instagram at @VanessaJass, youtube.com/VanessaJass or email me at findingperspectivepodcast@gmail.com.
First 15 Sentences + T5	This episode we have Vanessa Jazz. Vanessa is a toronto based family and real estate lawyer, motivational speaker and upcoming author. Vanessa has recently launched her own charity called Survivors Unleashed. Vanessa shares her incredible story of being coerced into the sex trafficking world in Canada. She also discusses how she was forced into the sex trafficking world as a teenager. Vanessa's website: www.survivorsunleashed.com/ Website: www.survivorsunleashed.com Facebook: www.facebook.com/groups/findingperspectivepodcast/ Instagram: @findingperspect.
SpanBERT + T5	This is a story of how I was coerced into the sex trafficking world and what it was like to be in the sex industry. It's not about money, it's about being happy. Listen to hear my story!

Table 3: Summaries generated for episode *04Llj5QjDot46TEDGj7lbI*

Model	Summary
T5	This episode is for you if you haven't already. If you haven't already, be sure to download the Spotify app and search for DKM/H or browse podcasts in your library tab. Enjoy!
First 15 Sentences + T5	This episode is for you if you haven't already. If you haven't already, be sure to download the Spotify app and search for DKM/H or browse podcast in your library tab. Enjoy!
SpanBERT + T5	This episode is for you if you haven't done already. Be sure to download the Spotify app and search for DKM/H or browse podcasts in your library tab. You can also share this episode with your friends on Instagram. If you haven't done already, be sure to download the Spotify app and search for DKM/H or browse podcasts in your library tab.

Table 4: Summaries generated for episode *45hMVOlYDHoX0E92qKjKTW*

by the T5 and the First 15 Sentences + T5 models are the same. We believe this episode may have been challenging because no clear entities or topics were introduced. Furthermore, the GCP-ASR transcription reduced the episode's clarity - even for human readers. This would have negatively impacted the effectiveness of any summarisation model. The transcript for the episode is provided below:

> *Hey there. Thanks for listening to the decaying major podcast. Make sure you follow your favorite podcast. So you never miss an episode like this one or if you become a premium user you can download the episode so you can listen to them offline like I do when you're on a plane or wherever you're traveling and also you can share this with your friends on Instagram. If you haven't done so already be sure to download the Spotify app and search for D. Km/h or browse podcast in your library tab. Control control is the thing wherein I'll catch the conscience of the king control is the Whirlwind wrapped in a puzzle words are the pieces and your mouth the muzzle bumping. Seeped in thick and I see my blood cold heart dicey. That's sodium vapor your heart seemed safer - thin like paper. rolls, like this old sharpened like a fossil whittled like a stick rhubarb crumble sweet and thick Mercury disc bite like allergen opaque like a curtain speckled like a freckle tested like a medal. You had five masks ten percent polyester use your best friend as a tester 10 linen. You kept this one locked away hidden and Rib knit 7 to 10. Just a kid get out quick 60 cotton weighted used this at the very bottom, but that last 10% when that's all you sick. What do you hook with pigment died and fries fried that is rayon and crayons your OCD banded your Rainbow of coded sized and alphabetized. Legs Pistons coal powered steam engine freight train that clicker clacker of refrain is so hip jugular hit the lens flipped to admit of is our fish bowl shape distorted Vape the smoke cloud and moved and changed its shape make a defensive Pact. with a*

> *world map trace the curvy line across the small of your back.*

7 CONCLUSIONS AND FUTURE WORK

In this paper, we discussed our participation in the Summarisation Task of the TREC 2020 Podcasts Track. Our results provide evidence that effective podcast summarisation can be done by only focusing on the beginning parts of the podcast episode. However, areas such as entity recognition and discourse analysis should be investigated for future work.

REFERENCES

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

[2] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2020.

[3] Derek Miller. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165, 2019.

[4] Office of Communication. Audio on demand: the rise of podcasts, Sep 2019.

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

[6] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.