Enhancing user experience and engagement through Podcast content Summarization

Saswata Rautray

72, South University Place, Apt.11, Stillwater, Oklahoma

saswata.rautray@okstate.edu

Bharath Raj Muppalla

37, South University Place, Apt.12, Stillwater, Oklahoma

bharathraj.muppalla@okstate.edu

Abstract

The podcast industry is rapidly expanding and garnering tremendous market appeal. As per the reports, podcasting will be a $94.88 billion industry by 2028. However, today, the finding and comprehension of podcast material appear to be less advanced when compared to other sorts of media such as music, movies, and news. Also, a lot of consumers move away from podcast listening due to the sheer length of it. This necessitates the development of more computationally efficient methods for podcast analysis, such as automated summarization and key topics/hashtags identification. With the rapid development in Natural Language Processing techniques, especially the success of attention mechanism and Transformer architecture, the text summarization task has received increasing attention. Different from news, podcasts have a unique characteristic, such as being conversational/dialog format which makes podcast summarization task more challenging. This paper presents an analysis of podcast summarization using the Freakanomics podcasts dataset which we scraped from its website. In this study, we aim to share our results on our analysis to understand current state-of-the-art pre-trained models. We will be using some of the popular extractive and abstractive summarization techniques like Textrank, LSA (Latent semantic analysis), T5 transformers, BART method, Pegasus.

The user experience and engagement will be enhanced through podcast summaries and will play a crucial role in helping end-users make listening decisions and are a crucial feature in podcast recommendation systems as well as many downstream applications.

1. Introduction

Podcasting is a thriving industry. Podcasts have arisen as a hugely consumed kind of internet content, owing to the increased accessibility of production tools and scaled broadcast via big streaming platforms.

Text summarization is a subfield of CS and linguistics that aims to compress a lengthier text into a shorter one, with both the input and output lengths being model variables. To be deemed a good summary, the summarized text must possess certain characteristics, including the absence of duplicate information, the use of appropriate language, the ability to be brief, and, most crucially, the inclusion of all relevant information from the source text being summarized. As the internet has grown in popularity, a tremendous number of text documents have been available, making it increasingly vital to be able to extract and retain the relevant information in a timely and effective manner. Manually summarizing papers is both time consuming and costly for people, which is one of the reasons why automatic text summarization is becoming increasingly significant as the amount of text documents in circulation grows. Podcasts are a rapidly developing audio-only media, and the development of automatic speech-to-text technology has resulted in more and better transcriptions. Current summarizing approaches are challenging to utilize since podcasts as a medium, as well as their transcriptions, differ in terms of category, structure, speakers, duration, and other aspects. Abstractive summarizing, extractive summarization, and a mix of the two are the most popular types of text summary.

Understanding podcasts is becoming increasingly crucial as their popularity grows. One method to comprehend them is to identify the categories that are mentioned. Topics are commonly used by categorization systems and information access technologies to organize or traverse podcast

collections. However, annotating podcasts with subjects remains difficult due to the designated editorial genres being wide, diverse, or deceptive, or due to data problems (e.g. short metadata text, noisy transcripts).

Our objective, by exploring the Transformer-based models, is to filter out the unnecessary content and discover the most valuable information for summary creation.

2. Data Overview

The dataset consists of 93 transcribed podcast episodes from Freakonomics radio's "No Stupid Questions" season encompassing nearly 60 hours of speech. However, using web scraping, we extracted the transcribed textual form of the podcasts. The final features of the dataset include the Podcast Title, Podcast transcribed text, Length of the episode.

Podcasts exist in a range of forms, styles, structures, tempo, and material, posing a slew of new issues in natural language processing, including specialized summarization challenges.

2.1. Samsum Dataset

The SAMSum dataset contains about 16k messenger-like conversations with summaries. Conversations were created and written down by linguists fluent in English. Linguists were asked to create conversations similar to those they write on a daily basis, reflecting the proportion of topics of their real-life messenger conversations. The conversations were annotated with summaries. The SAMSum dataset was prepared by Samsung R&D Institute Poland and is distributed for research purposes.

This is the dataset on which the pre-trained models we used further are fine-tuned on so that the conversational form of contents is learnt. This is important since the podcast data we intend to use is also a conversational dataset.

3. Methods Plan/ Overview

The first step is Preprocessing, which is the process of changing raw data into a form that can be used by the model. Preprocessing has become a critical step in machine learning to guarantee that the data quality is acceptable enough for learning and analysis since real-world data is frequently noisy, inconsistent, and incomplete.

In Extractive text summarization, the algorithm selects the most representative sentences from the paragraph that can effectively summarize it. Unlike Abstractive summarization, it does not paraphrase the text. Models like Textrank, LSA (Latent semantic analysis), are extractive in nature, while T5 transformers, BART method, Pegasus are abstractive models

The experiments are designed by initially using a set of extractive baseline models that is for general summarization, and then progressing to more complex models by starting from various pre-trained checkpoints and finetuned on the conversational dataset. The following are some of the popular methods we plan to employ:

3.1. TextRank

TextRank: It is a graph-based model that may be used as an unsupervised approach to extract sentences from a text, and it is classified as an extractive summarization method.

It is an unsupervised algorithm based on weighted graphs from a paper by Mihalcea et al. It was added by another incubator student Olavur Mortensen. TextRank Summarization ranks sentences using the PageRank algorithm that Google used for ranking webpages, where "votes" or "in-links" are represented by words shared between sentences.

PyTextRank/PyTLDR is a python implementation of the original TextRank algorithm with a few enhancements like using lemmatization instead of stemming, incorporating Part-Of-Speech tagging and Named Entity Resolution, extracting key phrases from the article and extracting summary sentences based on them.

3.2. Latent Semantic Analysis (LSA)

LSA works by projecting the data into a lower dimensional space without any significant loss of information. One way to interpret this spatial decomposition operation is that singular vectors can capture and represent word combination patterns which are recurring in the corpus. The magnitude of the singular value indicates the importance of the pattern in a document.

3.3. BART (large-sized model), fine-tuned on CNN dailymail and SAMSum dataset

BART uses both BERT (bidirectional encoder) and GPT (left to the right decoder) architecture with seq2seq translation. BART achieves the state-of-the-art results in the summarization task.

BART model pre-trained on English language, and fine-tuned on CNN Daily Mail, a large collection of text-summary pairs. It was introduced in the paper BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Lewis et al. and first released in (https://github.com/pytorch/fairseq/tree/master/examples/bart). BART is particularly effective when fine-tuned for text generation (e.g. summarization, translation) but also works well for comprehension tasks (e.g. text classification, question answering).

bart-large-cnn-samsum: This model was obtained by fine-tuning facebook/bart-large-cnn on Samsum dataset. Building on the bart-cnn pre-trained checkpoint, the bart-large-cnn-samsum model takes the same starting point and is then finetuned on the another dataset, with the hope that the model will profit from the already learned job of summarization from the pre-trained checkpoint.

3.4. T5 Large Transformer, fine-tuned on C4, Wiki, etc. and SAMSum dataset

T5, or Text-to-Text Transfer Transformer, is a Transformer based architecture that uses a text-to-text approach. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text. This model was fine-tuned on the SAMSum corpus from t5-large checkpoint.

3.5. Pegasus Transformer, fine-tuned on CNN and SAMSum dataset

Building upon earlier breakthroughs in natural language processing (NLP) field, Google's PEGASUS further improved the state-of-the-art results for abstractive summarization, in particular with low resources. On a high level, PEGASUS uses an encoder-decoder model for sequence-to-sequence learning. This model is a fine-tuned version of google/pegasus-cnn_dailymail on the samsum dataset.

4. Results

The results of the research will be provided per evaluation measure, with the score for each model published in a single table. The metrics are initially provided at an overview level, such as mean scores, and then a visualization of the score distribution for each model per measure is displayed.

4.1. ROUGE score

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations. It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced). ROUGE-1 refers to overlap of unigrams between the system summary and reference summary. ROUGE-2 refers to the overlap of bigrams between the system and reference summaries.

4.1.1. BART

	rouge-1	rouge-2	rouge-l
r	0.332579	0.144872	0.312217
p	0.472669	0.203604	0.443730
f	0.390438	0.169288	0.366534

4.1.2. Pegasus

	rouge-1	rouge-2	rouge-l
r	0.319005	0.139744	0.307692
p	0.507194	0.246050	0.489209
f	0.391667	0.178250	0.377778

4.1.3. T5

	rouge-1	rouge-2	rouge-l
r	0.334842	0.165385	0.319005
p	0.649123	0.375000	0.618421
f	0.441791	0.229537	0.420896

4.1.4. TextRank

	rouge-1	rouge-2	rouge-l
r	0.414027	0.198718	0.373303
p	0.376543	0.152709	0.339506
f	0.394397	0.172702	0.355603

4.1.5. LSA

	rouge-1	rouge-2	rouge-l
r	0.457014	0.241026	0.409502
p	0.382576	0.175537	0.342803
f	0.416495	0.203133	0.373196

4.1.6. RelevanceSummarizer

	rouge-1	rouge-2	rouge-l
r	0.454751	0.223077	0.411765
p	0.328969	0.148973	0.297872
f	0.381766	0.178645	0.345679

- ROUGE-n recall=40% means that 40% of the n-grams in the reference summary are also present in the generated summary.
- ROUGE-n precision=40% means that 40% of the n-grams in the generated summary are also present in the reference summary.
- ROUGE-n F1-score = It gives us a reliable measure of our model performance that relies not only on the model capturing as many words as possible (recall) but doing so without outputting irrelevant words (precision)

ROUGE-L measures the longest common subsequence (LCS) between our model output and reference. All this means is that we count the longest sequence of tokens that is shared between both. The idea here is that a longer shared sequence would indicate more similarity between the two sequences.

Considering F1 scores of Rouge-L, we can see that

T5(42%) < Pegasus(37.7%) < LSA(37.3%) < BART(36.6%) < TextRank(35.4%) < Relevance Summarizer(34.5%)

Therefore, the abstractive models tend to perform better than the extractive methods employed.

4.2. Human Evaluation

We also conducted a human evaluation of the dialogue summary to assess its informativeness, conciseness and coverage. Informativeness measures how well the summary includes key information. Conciseness measures how well the summary discards the redundant information. Coverage measures how well the summary covers each part of the dialogue. In order to reduce variance caused by humans, we have 2 external human evaluators, and they were asked to rate each summary on a scale of 1 to 5 (higher is better) for each metric.

Models	Informativeness	Coverage	Conciseness	Average
BART	4.5	4	3	3.83
T5	4	4.5	4.5	4.33
Pegasus	4	4	3	3.67
TextRank	3	3.5	3	3.17
LSA	3.5	3.5	3	3.33
Relevance Summarizer	3	3	3	3.00

From the above average scores, the abstractive models tend to perform better than the extractive methods.

5. Conclusion and Future Scope

Given long documents to read, our natural preference is to not read, or at least, to scan just the main points. So having a summary would always be great to save us time and brain processing power.

This paper describes our approaches to produce summaries for podcast episodes using the transcribed texts as input. We explored how well abstractive and extractive summarization strategies perform. Abstractive methods are effective at capturing the entirety of the long input sequences.

The future work would involve the Fine tuning of the models to the specific dataset of freakanomics dataset. The availability of compute resources would be required. Also, the availability of existing summaries to train would help. There is a lack of existing tags for evaluation and measuring success of the tags. The further work could involve determining the successful hashtags for the podcast materials.

6. References

"Series Full - Freakonomics." *Freakonomics*, 30 Dec. 2021, freakonomics.com/series-full/nsq/. Accessed 12 Apr. 2022.

"Samsum · Datasets at Hugging Face." *Huggingface.co*, 13 Dec. 2021, huggingface.co/datasets/samsum. Accessed 12 Apr. 2022.

Feng, Xiachong, et al. *Language Model as an Annotator: Exploring DialoGPT for Dialogue Summarization*. 2021.

Karlbom, Hannes, and Ann Spotify. *Abstractive Podcast Summarization Using BART with Longformer Attention*.

Weng, Jiahao. "How to Perform Abstractive Summarization with PEGASUS." *Medium*, Towards Data Science, 4 Feb. 2021, towardsdatascience.com/how-to-perform-abstractive-summarization-with-pegasus-3dd74e48bafb. Accessed 12 Apr. 2022.

Rocket Mortgage Technology. "Conversational Summarization with Natural Language Processing." *Medium*, Rocket Mortgage Technology Blog, 24 Nov. 2020, medium.com/rocket-mortgage-technology-blog/conversational-summarization-with-natural-language-processing-c073a6bcaa3a. Accessed 12 Apr. 2022.

7. Appendix

The link to python codes and sample summaries by the models outputs: https://github.com/saswatarautray/podcast-summarizarion