BERT based summarizer for Audio Podcast transcripts, combined with Topic modeling

Introduction

- Podcasts are one of the **fastest growing content** available on the internet for consumption.
- Audio podcasts, while entertaining and informative, are usually ignored as text mining content because of the **conversational nature**.
- Our aim is to **summarize the transcripts and model the topics** covered in the transcript.
- This data could be used in recommendation systems, consolidate podcast content from multiple podcasts on the same topic, and other such applications.

Data

Transcript(as text)

- Available in JSON format
- Generated using Google Cloud Platform's Cloud Speech-to-Text API3(GCP-ASR)
- On average, over 6000 words per episode
- Word Error rate of 18.1%
- Format:

[{"words": [{"startTime": "0.900s" , "endTime": "1.4005","word": "Welcome",
"speakerTag":1}, {"startTime": "1.400s", "endTime": "1.5005","word":"to"},
{"startTime": "1.500s","endTime": "1.7005","speakerTag":1},
{startTime":"2.100s","endTime": "2.100s","word": "the", "speakerTag": 1},
....]

Data (continued)

Transcript(as text)

- The compressed audio transcripts are of the size 13 GB
- On decompressing the transcript data is made of 8 subsections
- each subsection upon decompressing is approximately 12 GB
- There is also a test set data that is present in the repository
- There are 1027 transcripts present in this which we us for proof of concept to demonstrate the pipeline
- Due to the lack of computational resources, restricted evaluation and data set generation

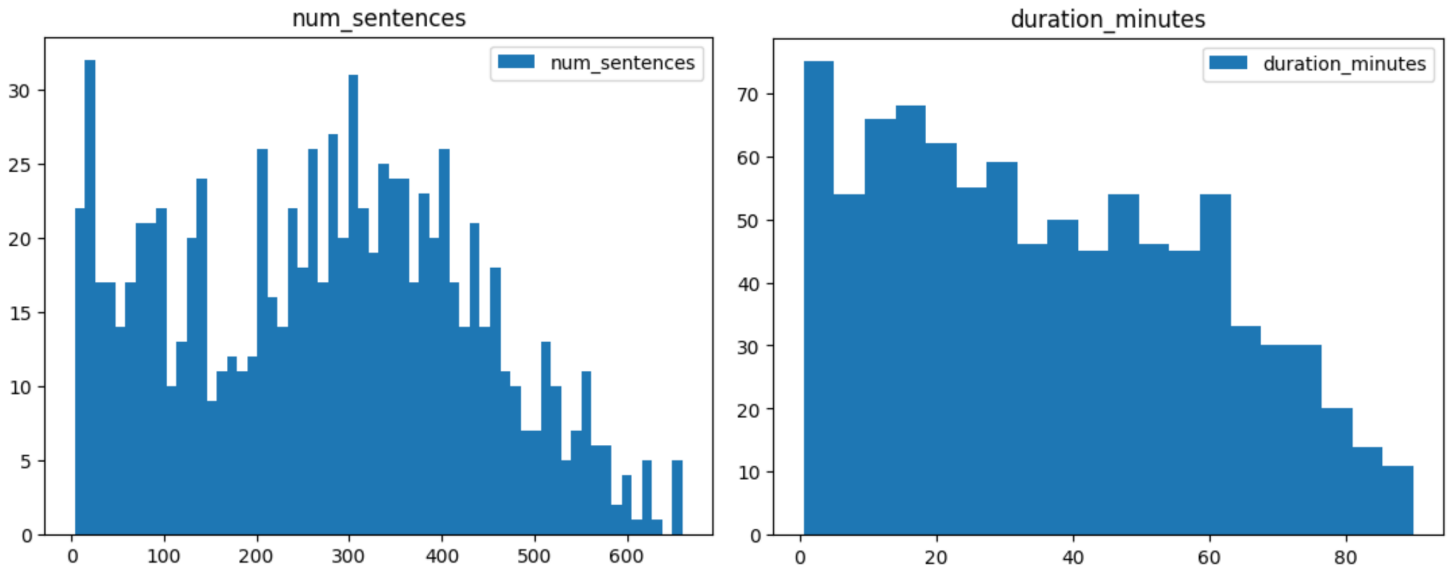
Data (continued)

Transcript(as text)

- A few quick stats about the data:
  - Number of records that are present: 1027
  - Number of records that are useful: 917
  - Mean podcast length (in minutes): 36.305
  - Mean number of sentences per transcript: 274.61
  - Unique vocabulary size: 51906 tokens

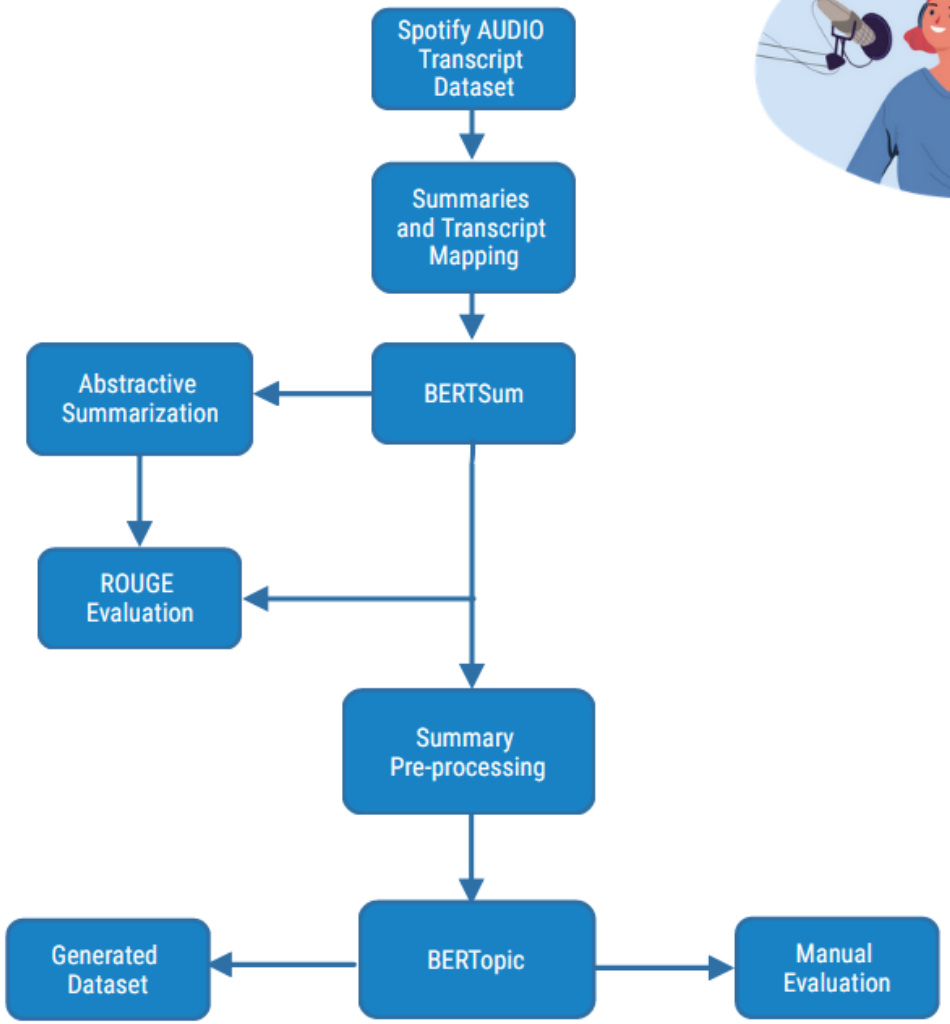
Data (continued)

Transcript(as text)



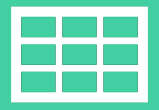
- Histograms showing the distribution of the number of sentences per transcript and the distribution of the length of the podcast(in minutes)

PODCAST SUMMARY GENERATION AND TOPIC MODELLING



Preprocessing

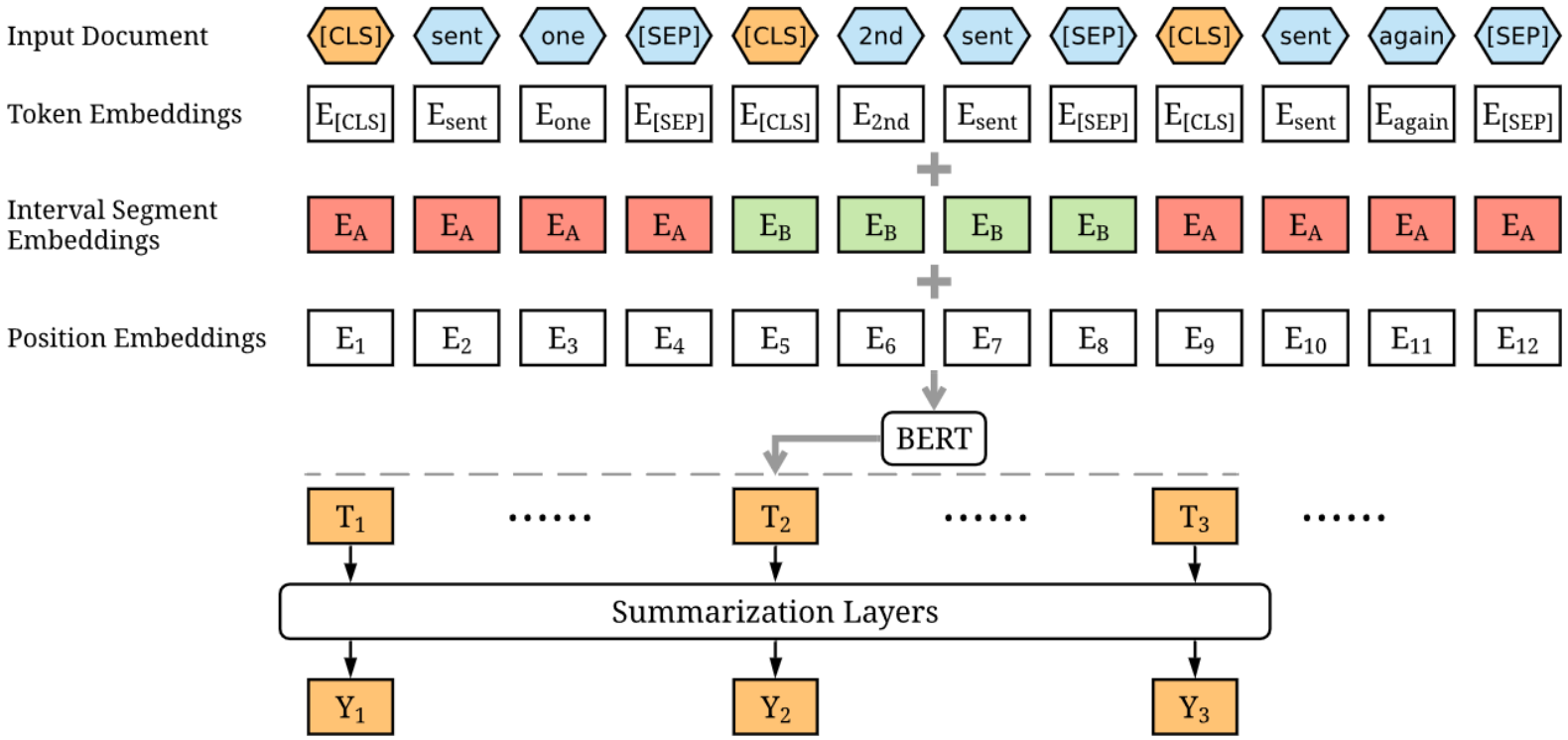
The provided data set has JSON files for transcripts and an XML file for summaries. To map the summary to the transcript, the episode title was used since the episode key was not in the XML.



Additional information in the JSON files was not useful, so it was discarded and the remaining data was aggregated into a single Excel sheet for all experiments and evaluations.

Columns used for processing is as follows: XML file path, Title, Summary, Description, Total time duration in secs, Total time duration in mins, Show filename prefix, Episode filename prefix, JSON file path, JSON transcript

BERTSum


Input Document
[CLS] sent one [SEP] [CLS] 2nd sent [SEP] [CLS] sent again [SEP]
Token Embeddings
E[CLS] Esent Eone E[SEP] E[CLS] E2nd Esent E[SEP] E[CLS] Esent Eagain E[SEP]
Interval Segment Embeddings
EA EA EA EA EB EB EB EB EA EA EA EA
Position Embeddings
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12
BERT
T1 T2 T3
Summarization Layers
Y1 Y2 Y3

BERTSum

- BERTSum algorithm used summarizer package, with a pre-trained model of size 1.34 GB.
- To generate summaries, the entire transcript is passed along with the ratio variable representing the summary size compared to the entire transcript.
- Three ratios used were 0.25, 0.3, 0.35. The chosen ratios are reasonable estimates for generating detailed summaries without being too vague or too large.
- Generating a single summary using a 4GB GPU took approximately 7.5 seconds.

Evaluation of BERTSum

- First, Gold values i.e, the summary from the given dataset was used for evaluation of the generated summaries.
- The quality of the generated summaries is evaluated using ROUGE metrics, including R1 (unigrams), R2 (bigrams), and RL (longest common subsequences).

Summary Ratio	Mean R1	Mean RL
0.25	0.365320	0.337778
0.30	0.380252	0.354347
0.35	0.394339	0.369475

- However, since the gold summaries are shorter than the generated summaries, ROUGE may not be a foolproof method for determining the best summary ratio. Therefore, another metric may be necessary.

Abstractive VS Extractive Summarization

- Extractive summarization selects and condenses important sentences from the original text. This retains the wording and meaning of the original text, but can result in a summary lacking coherence and readability.
- Abstractive summarization generates a new, concise summary that captures the essence of the original text in natural language. It is more challenging but can create summaries that are more coherent, readable, and grammatically correct

Cont.

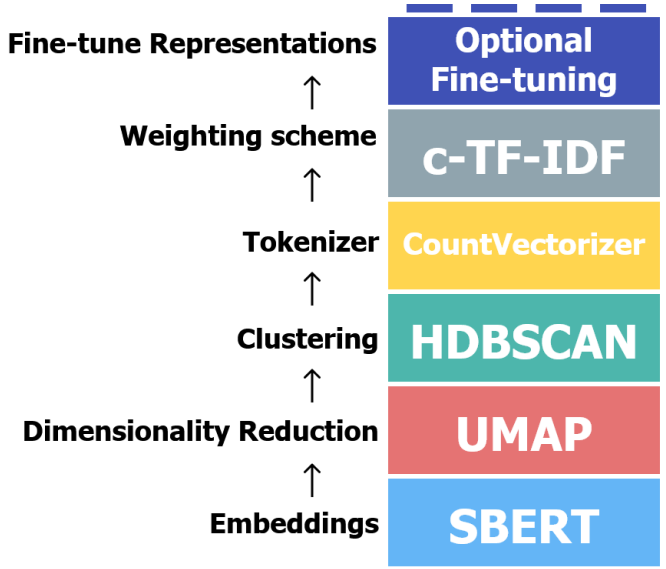
- Second evaluation is implemented using an abstractive summarizer (facebook/bart-large-xsum) to compare the information in the summary with the information in the entire transcript.

- For abstractive summarization, the preprocessed content is sent to Transformers library is used with the Facebook model to generate summaries for both the entire transcript and the extractive summaries.

Summary Ratio	Mean R1	Mean RL
0.25	0.286904	0.263419
0.30	0.314593	0.289201
0.35	0.339918	0.312618

BERTopic

- BERTopic is a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters.
- To generate the topics, the summary is preprocessed to remove non alpha-numeric letters and stopwords
- Pre-processed Summaries are fed into the BERTopic model and a separate model is generated for each podcast episode.
- 5 best topics are extracted based on the probability and evaluated.



Evaluation of BERTopic

- First, 2 Random Sets each containing 5% of the Total Data is extracted and prepared for Evaluation.
- Each Data Point in the Sets are manually evaluated based on the Podcast Transcript content, the summary of the podcast and the relevancy of the Topics.
- Data Points are categorized into Relevant/Non-Relevant categories based on the top 5 set of Topics generated for each summary.

Sets	Data Count	Relevancy (%)
Set 1	47	54
Set 2	48	60

- We get an average relevancy of 57% out of the 2 random sets generated.

Conclusion

- Both the pipelines together successfully provide a proof of concept to generate new data set
- The highest ROUGE score is 0.39
- Because of the obvious reduction in size of text from transcript to summary
- The need for additional abstractive summarization.
- In the topic modeling, a lot of random garbage present in every transcript
- This is due to the conversational nature of the podcasts
- Explains the mediocre relevance of 57% in the test set.

Future work

- With the availability of higher computational resources, both models can be fine-tuned to achieve higher accuracies.
- Build the dataset completely for all parts of the transcripts.
- Deploy the pipelines as APIs that can be used on not just on a bunch transcripts but single transcripts as well.
- Integrate summarization module with a Speech-to-text system to build summaries in real time.

References

- https://arxiv.org/pdf/2004.04270.pdf - Spotfiy Podcast Dataset
- https://arxiv.org/pdf/2203.05794.pdf - BERTopic: Neural topic modeling with a class-based TF-IDF procedure
- https://arxiv.org/pdf/1903.10318.pdf - Fine-tune BERT for Extractive Summarization

THANK YOU!