PoliTO at TREC 2021 Podcast Summarization Track

Lorenzo Vaiani
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy
lorenzo.vaiani@polito.it

Moreno La Quatra
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy
moreno.laquatra@polito.it

Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy
luca.cagliero@polito.it

Paolo Garza
Dipartimento di Automatica e Informatica
Politecnico di Torino, Italy
paolo.garza@polito.it

Abstract

This paper presents the approach proposed by the PoliTO team to accomplish the TREC 2021 podcast summarization task. The purpose is to extract synchronized text/audio segments that convey the most relevant podcast information. The main challenge is to consider is the multimodal nature of the data source, which comprises both textual and acoustic sequences. PoliTO presents a two-stage pipeline that (i) extracts relevant content from multimodal sources and (ii) leverages the extracted content to generate abstractive summaries by using an attention-based Deep Learning architecture. The extractive stage combines the high-dimensional encodings of both textual and audio sources to build a neural network-based regression model. The key idea is to predict the textual similarity between the selected text snippets and the podcast description by also exploiting the underlying information provided by the acoustic features. While audio summaries are obtained by concatenating selected audio samples, summaries in textual form are generated by exploiting the selected information as input of a sequence-to-sequence generative model.

1 Introduction

The Podcasts Track at the Text Retrieval Conference (TREC) is intended to foster research in podcast retrieval and access by researchers from the information retrieval, the NLP, and the speech analysis fields [9]. The research challenge consists of two main tasks: segment retrieval and podcast episode summarization. The former one is focused on retrieving the two-minute video segments that are most pertinent to a given query. The latter task aims at extracting a concise summary of each podcast episode consisting of a shortlist of speech transcription/audio extracts. The system proposed by the PoliTO team addresses the podcast summarization task.

The main challenges that have to be faced by the participants to the 2021 Edition of the podcast summarization task are (i) the *integration of the acoustic features* deeply into the summarization process, and (ii) the ability to process *heterogeneous podcast episodes and shows*. Challenge (i) entails analyzing multimodal content to gain insights into both textual and acoustic features. The resulting summary is expected to include both audio and textual segments. To address the aforesaid issue, the PoliTO approach relies on multimodal Deep Learning architecture. Challenge (ii) is related to the presence in the source dataset of heterogeneous sets of podcast episodes and shows. The need to summarize podcasts with highly variable content and length calls for new, effective approaches to attend relevant information in the raw data. For example, the automatic recognition of advertisements has shown to be particularly helpful while coping with long-lasting podcasts.

PoliTO team's participant paper. Podcast Summarization track at Text Retrieval Conference (TREC 2021).

The PoliTO system first retrieves a selection of speech transcription segments, which can be straightforwardly mapped to the corresponding portions of the original audio track. Then, the output summary in textual form is generated on top of the extracted content by using an abstractive summarization model. For each episode the sentence retrieval step evaluates content relevance according to its similarity with the (creator-provided) episode description.

We propose an architecture composed of multiple components: a text encoder, an acoustic feature aggregator, and a multi-layered regression network whose aim is to extract the multimodal data pairs that will then constitute the input of the audio selector and the abstractive summarizer. The PoliTo system supports the generation of multiple variants of the textual summaries depending on the end-user preferences on number of input sentences and the minimum summary length.

This paper is organized as follows. Section 2 overviews of the data collection and task-related details. Section 3 discusses the methodology adopted for designing the proposed system. Section 4 describes the systems runs submitted for TREC evaluation and discusses the main findings. Finally, Section 5 draws conclusions and enumerates possible future developments.

2 Overview

2.1 Data Collection

A detailed description of the Spotify podcast dataset can be found in [3]. We briefly recap its main characteristics that were useful for the design of our multimodal summarization system. The dataset consists of a collection of more than 100,000 podcasts grouped into approximately 1,800 shows. For each episode, there are several information divided into three main classes: audio, text and metadata. The acoustic information available in the original data collection includes (i) the original audio file, (ii) OpenSmile [5] low-level descriptors and (iii) Yamnet[1] embedding vectors. Each episode comes its transcript extracted from the audio track. It is segmented into textual segments and each of them contains timing information for text-audio alignment. Episode metadata contains additional information such as the podcast creator and the his/her manually-written description for the episode.

2.2 Podcast summarization task

TREC 2021 launched two different research challenges related to the podcast track: segment retrieval and summarization. The PoliTO team proposes a system tailored to the podcast summarization task. Its objective is to generate/extract a short digest that summarizes the content of the episode in both textual and audio formats. While the former can be either extracted from the text or abstractly generated, the latter must necessarily be an extract of the audio track shorter than one minute (according to the task rules). The evaluation is manually performed by the NIST assessors and the quality of each summary is ranked within a scale of 4 possible values: *Excellent, Good, Fair, Bad*. In addition to the quality evaluation, the assessors also provide answers for nine "yes/no" questions regarding the content and quality of the generated digests:

- 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 end points?

[1] `https://github.com/tensorflow/models/tree/master/research/audioset/yamnet`. Latest access: November 2021.

- Q9: Does the audio clip give a sense of what the podcast sounds like, (as far as you can tell from listening to it)?

3 The PoliTO system

In this section, we present the designed deep learning architecture for multimodal podcast summarization. It is designed to perform simultaneous analyses of audio and text modalities in order to create effective representations of the input data. As first step, for each sentence of the podcast we simultaneously encode its text and aggregate audio information into two separate fixed-size vectors. Then, we combine the two feature vectors to process them with a multimodal regression network. It is composed of a stack of 7 fully connected layers with ReLU activation fuction [6]. The network is aimed at predicting the relevance of each text-audio snippet to the output summary. We select the top-ranked sentences to (i) provide them as input for a the abstractive summarization component that generate the final abstractive summary and (ii) arrange the audio summary that is instrumental for creating a trailer for the podcast episode.

3.1 Description filtering

We apply a preliminary filtering step to the description content aimed at removing advertisements as well as commercial contents. To this aim, the description content is split by exploiting the sentence tokenization provided by spaCy library [7]. Next, similar to [12] it fine-tunes a pre-trained BERT model [4] for binary text classification. The model is trained on a small subset of manually annotated description snippets to automatically recognize advertising content. Advertising content is early pruned from the descriptions before running the summarization process.

3.2 Text encoding

Each sentence in the episode transcripts is encoded using of Transformer-based model trained on sentence similarity tasks [11]. The state-of-the-art approach is used as reference encoder for the podcast speech transcription. In our experiments we use the `paraphrase-mpnet-base-v2` based on MPNet model [13]. It allows a maximum sequence length of 512 tokens and generates 768-dimensional vector for each sentence.

3.3 Acoustic information aggregation

The proposed multimodal architecture also leverages acoustic information. Our model rely on the OpenSmile [5] features provided by dataset authors. The feature vector for each audio sentence is an 88-dimensional vector that represents some acoustic characteristics of the given speech audio segment. Those features are used for sound description and to identify some speaker-related aspects, such as emotion, age, gender, and personality. OpenSmile features extraction occurs with a sampling frequency of one second, so we represent the acoustic information of a sentence aggregating the 88-dimensional extractions that occurred for the corresponding audio snippet. In the aggregation phase we compute means, standard deviations, minimums and maximums values for each descriptor. By concatenating the resulting vectors we get the 352-dimensional vector per audio sample.

3.4 Multi-layered regression network

The vectors obtained by audio and text encoding steps are then fed to multi-layered fully-connected architecture. It aims at processing the multimodal representation, including a mixture of deep learned and hand-crafted features, obtained by concatenating text- and audio-related encodings. The network consists of a stack of fully-connected (FC) layers and it is trained to predict a score that represents the pertinence of the audio, text pair to the episode description. In our experiments, we set the depth of the fusion network to 7, where the width for the first three layers and the last four is set to 1120 and 768 respectively. The network is trained to predict the relevance of the multimodal text-audio segments to the creator-provided episode description. The relevance score for the training set is computed, for each sentence, as the semantic similarity between the sentence itself and the episode's description. To this aim we exploit the Sentence-BERT model trained on Semantic Text Similarity (STS) task [2].

3.5 Summary generation

The PoliTO system provides end-users with a multimodal summary of the podcast episode including both audio and text. To produce textual summary it selects the top-scored sentences according to the model prediction. Those sentences are concatenated and fed as input of the abstractive summarizer. The abstractive summarization system relies on a transformer-based encoder-decoder architecture that can process long text sequences, namely Longformer [1]. The Longformer model is based on transformer architecture [14] while introducing relevant modification to the original attention mechanism. While the original self-attention scales quadratically with the sequence length, thus hindering the processing of long text sequences, Longformer introduces a windowed attention mechanism that scales linearly with sequence length, enabling efficient long documents processing. LED (Longformer Encoder-Decoder) is the sequence-to-sequence variant of the architecture that support long document-based generative tasks. LED reduces the limitations on the number of sentences that need to be selected to contribute to the generation of the abstractive summary. Indeed, the number of selected sentences is one of the two parameters that characterize our submissions to the competition, in addition to the threshold on the minimum length of the generated summary. The sequence-to-sequence model is fine-tuned for three epochs to generate summaries as close as possible to the author's provided description.

The generation of audio summaries entails the selection of the K audio samples associated to the top-scored multimodal pairs. We choose K as the minimum number of pairs such that the total duration of the audio snippets exceeds the threshold set for the Podcast Summarization Track (i.e., 60 seconds). The selected audio snippets are (i) re-ranked according to their ascending order of appearance in the original podcast, (i) concatenated and (iii) trimmed to avoid exceeding the maximum duration (i.e., 60 seconds).

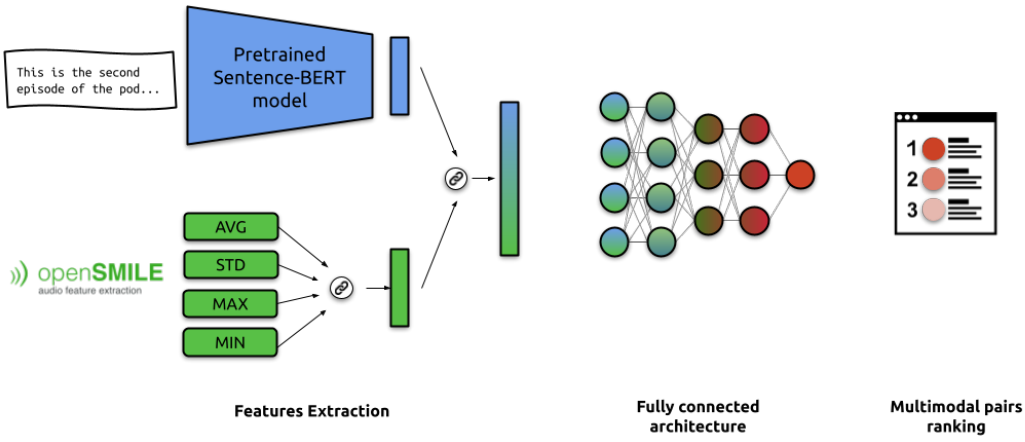

Figure 1: Sketch of the PoliTO architecture.

4 Experimental Results

4.1 Submitted runs

TREC organizers allowed participants to submit up to 4 different runs. PoliTO applied the approach described in the previous section to generate 4 different textual outputs. They differ in the minimum length of the generated summaries and the number of selected sentences from the original transcript. Each textual summary is paired with an audio trailer extracted using the same approach for all the runs. Both parameters characterizing each submission are related to the training and generation procedures of the abstractive summarization model.

Each run submitted by our team is described below:

- *25_32-128*: this run includes abstractive text-based summaries with a length that ranges between 32 and 128 words. The input of the abstractive model consists of the 25 top-ranked sentences.

- *50_32-128*: this run includes abstractive text-based summaries with a length that ranges between 32 and 128 words. The input of the abstractive model consists of the 50 top-ranked sentences.

- *50_64-128*: this run includes abstractive text-based summaries with a length that ranges between 64 and 128 words. The input of the abstractive model consists of the 50 top-ranked sentences. The choice of increasing the length of the generated summaries is due to the analysis of the summaries generated in the previous runs. In those cases, the output summaries tend to be very short with a token length close to the lower limit.

- *100_32-128*: this run includes abstractive text-based summaries with a length that ranges between 32 and 128 words. The input of the abstractive model consists of the 100 top-ranked sentences. This parametrization of the system aims at reducing the impact of extractive sentence selection. The model acts as a filter only for very long episodes.

4.2 Human evaluation

NIST evaluators judged 193 episodes randomly selected from the full test set consisting of approximately 1000 episodes. Task organizers also provides a baseline summary, for each podcast in the test set, based on the content of the first minute of the episode. Table 1 shows the results of our submissions comparing them with the proposed baseline. The run identifier is reported in the first column, the average quality (between 0 and 3) in the second one and the percentage of "yes" given by the assessors for each remaining question in the rest of the columns. The best value for each column is written in boldface. Note that, for question 6, the smallest value correspond to best performances since the "yes" answer has a negative meaning.

Analyzing the results, the human evaluation shows that our summaries outperforms the baseline in the majority of cases. On average the highest quality score (e.g., Quality) is obtained by setting the minimum length to the highest value (e.g., 64 tokens). Our runs get a better rating for the majority of other questions as well. Grouping the questions according to their semantic meaning,

- Q1 and Q2 are the only two questions for which the baseline gets a better evaluation. Both questions are related to the recognition of names, titles and personal information from the original podcast. The lower scores associated to our submissions are probably due to the use of LED model. Its peculiar attention mechanism is prone to hallucinations [8] thus could happen that the model fabricate or alter content that is not present in the input data.

- Q3, Q4 and Q5 are content-related questions and for two of them (Q3 and Q5) the best score is obtained by the submission *50_64-128* (which achieves a good rating also for Q4, slightly lower than the top-scored one). Those scores are mainly related to the amount of information contained in the summaries. Our top-scored submission generate longer summaries compared with others since the parameter for minimum number of words is doubled.

- Q6 is related to the redundancy of the generated summary. The best results is obtained by the submission that is provided with the longer input data (e.g., *100_32-128*). This result demonstrates that allowing the model to process an higher number of input sentences helps reducing the redundant content in the output summaries.

- Q7 and Q8 are syntactic- and linguistic-related questions. In both cases the best performing submission is *50_32-128*. It turned out to be the best model to balance the amount of input and output information to generate fluent summaries.

- Q9 is the only audio-related question. Our submissions obtain slightly different evaluations even if the output audio snippets are shared among all submission. This can happen when the audio summaries for the same episode are manually evaluated from different assessors. In all cases, however, our model achieves higher rating if compared with the baseline.

When compared with other system submissions for the Podcast Summarization task, our best run (e.g., *50_64-128*) ranked first and second-best according to text and audio quality respectively [10].

	Quality	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
baseline	0.772	**0.549**	**0.326**	0.606	0.427	0.536	0.451	0.456	0.187	0.957
25_32-138	0.974	0.354	0.255	0.645	**0.513**	0.523	0.183	0.806	0.594	0.978
50_32-138	0.860	0.323	0.234	0.615	0.437	0.500	0.204	**0.811**	**0.615**	0.989
50_64-128	**1.010**	0.378	0.285	**0.682**	0.503	**0.562**	0.292	0.715	0.536	0.983
100_32-128	0.917	0.333	0.256	0.606	0.446	0.497	**0.182**	0.776	0.601	**0.994**

Table 1: NIST evaluation results.

Model	Quality score	Summary
Creator-provided	/	Welcome! In this episode of A Dam PJO Podcast Caleigh and Izzy discuss The Titan's Curse, as well as 2006 trends, godly interventions, and strongly worded opinions on Bianca di Angelo. Hope you'll join us! Come talk to us! Instagram, run by Caleigh https://www.instagram.com/dampjocast/ Twitter, run by Izzy https://twitter.com/dampjocast E-Mail dampjocast@gmail.com
25_32-128	2	In this episode, we talk about Percy and the hunters, and we also talk about the Bianca and Talia encounter. We also get a glimpse into the mind of the Riptide, and a little bit of the mythology. —
50_32-128	2	In this episode, we talk about The Titan's Curse and the quest. We also talk about Bianca and her scar, and how she feels about the Greek gods.
50_64-128	3	We're back with another episode! This time we're talking about the Titans of Olympus and the trials of Apollo. We're also talking about Bianca's death and how she feels about her relationship with her mother. We also talk about the Riptides and the Ocarina of Time. We hope you enjoy it!
100_64-128	2	In this episode we talk about the Titans of Olympus and the trials of Apollo. We also talk about Bianca's accent and how she is a monster.

Table 2: Qualitative examples for the podcast episode: `6iRBuqS8OxEdShKp85uXQv`

4.3 Qualitative results

In Table 2 we report a qualitative comparison between summaries generated for each submission and the original podcast description. For each run we also report the quality score assigned by manual evaluation. All abstractive summaries are written in fluent English and use similar opening sentence, that is commonly used in episode descriptions. All of them discuss the same topic by using different level of details. Considering names and titles of people involved into the discussion, our submitted summaries focus on the same fictional character (e.g., Bianca). However, different submissions focus on different peculiar aspects of the character while mentioning concepts related to similar topics (e.g., Olympus, gods, and mythology). It is worth noting that, most of the people's names mentioned in the description do not emerge in automatically generated summaries.

5 Conclusion and future work

In this paper we present the PoliTO system designed for multimodal summarization of podcast episodes. A deep learning architecture is proposed to effectively combine text encoding and acoustic descriptors using a multimodal regression network.

The summarization performance achieved by our submissions is superior to that of the baseline model in terms of quality and for 7 binary questions out of 9. However, there is no submission that

has the highest rating for all characteristics assessed by human evaluation. Different parameters' configurations have different strengths and weaknesses. Analyzing the the manual evaluation, we can conclude that the use of multimodality is beneficial for the identification of key phrases of a podcast. However, our solution has some limitations due to the phenomenon of hallucination that hinders handling specific information (e.g., people's names or titles).

As future work, we plan to replace the acoustic descriptors exploiting ad-hoc speech embedding models. Furthermore, we aim at proposing an end-to-end architecture including feature extractors models into the training process, for both text and audio.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[2] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, 2017.

[3] Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 100,000 podcasts: A spoken English document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1459–1462, New York, NY, USA, 2010. Association for Computing Machinery.

[6] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[7] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[8] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online, June 2021. Association for Computational Linguistics.

[9] Rosie Jones, Ben Carterette, Ann Clifton, Jussi Karlgren, Aasish Pappu, Sravana Reddy, Yongze Yu, Maria Eskevich, and Gareth J. F. Jones. TREC 2020 podcasts track overview. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020.

[10] Jussi Karlgren, Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J. F. Jones, Sravana Reddy, and Edgar Tanaka. Trec 2021 podcasts track overview. 2022.

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[12] Rezvaneh Rezapour, Sravana Reddy, Ann Clifton, and Rosie Jones. Spotify at TREC 2020: Genre-aware abstractive podcast summarization. *CoRR*, abs/2104.03343, 2021.

[13] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*, 2020.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.