TREMA-UNH at TREC 2020

Sumanta Kashyapi, Laura Dietz
sk1105@wildcats.unh.edu, dietz@cs.unh.edu
TREMA lab, University of New Hampshire, U.S.A

Abstract

This notebook describes the submissions of team TREMA-UNH to the TREC Podcasts track. We participate in the summarization task of the track. We focus on combining extractive and generative summarization technique. The extractive model is used to detect salient parts of the input text and the generative model is used to generate summaries of only the selected segments.

1 Introduction

This year, team TREMA-UNH from the University of New Hampshire, USA, participated in the summarization task of TREC Podcasts track. The task is to generate summaries of long-form audio transcripts from various podcast episodes, describing the central theme of the episode. Details of the task, motivation and dataset are described in the TREC Podcasts overview papers [1, 2].

2 Approach

It has been shown that for short text snippets, Generative Adversarial Networks (GAN) are capable of producing summaries that are comparable to human performance. In our approach, each podcast transcript is first segmented into fixed size blocks. Then for each block, a recent implementation of GAN model [3] is used to generate a summary. It is observed that increasing the size of the input block degrades the readability and overall quality of the generated summaries. Hence, we kept the input block size relatively small, leading to high number of input blocks and their corresponding summaries per transcript. Our main contribution is to extract important segments from the podcast transcript that captures salient topics of the episode. We generate summaries of only those segments to construct the final summary.

Our overarching summarization approach consists of three steps as follows:

- **Step 1.** To extract the salient segments from a transcript, we train a model to generate a single topical embedding out of all input segments, that captures all salient topics across the segments. Precisely,

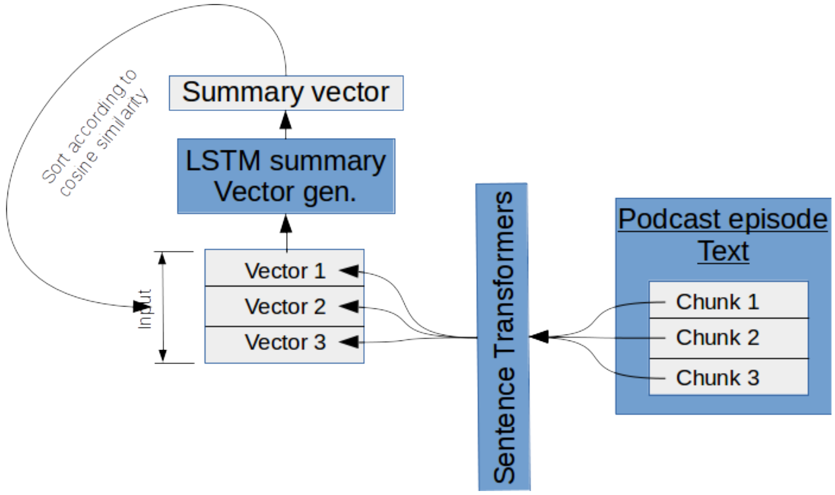

Figure 1: Overarching approach of podcast summary generation of a single episode

the model takes the topical embeddings of all input segments and generates a single embedding vector capturing salient topic information. We envision the generated topic vector as the representation of the summary in the latent embedding space.

- **Step 2.** Then the input segments are ranked according to the similarity between it's own topical embedding vector and the generated salient topic vector.
- **Step 3.** From the ranking, we pick top k segments, generate summaries of each segment using the GAN model and concatenate all of them to obtain the final summary of the transcript.

This steps are depicted in Figure 1.

Following section describes the first step of generating salient topic vector in detail.

2.1 Salient topic vector generation

We train a model that given a collection of text segments, emits a single topic vector, representing the summary of the input segments. We represent each text segment as a fixed-length vector using Sentence-BERT embedding [4]. Hence, the model learns to generate the topic vectors in the same embedding space. As the training dataset, we use the `benchmarkY1 train` split of TREC CAR year 1 dataset [5]. The ground truth in the dataset specifies relevant Wikipedia passages for different sections of Wikipedia articles. We leverage this hierarchical information into constructing our training samples. Figure 2 depicts the process of training data generation. More specifically, given relevant passages for a section, we try to predict a summary vector that is close to the Sentence-BERT

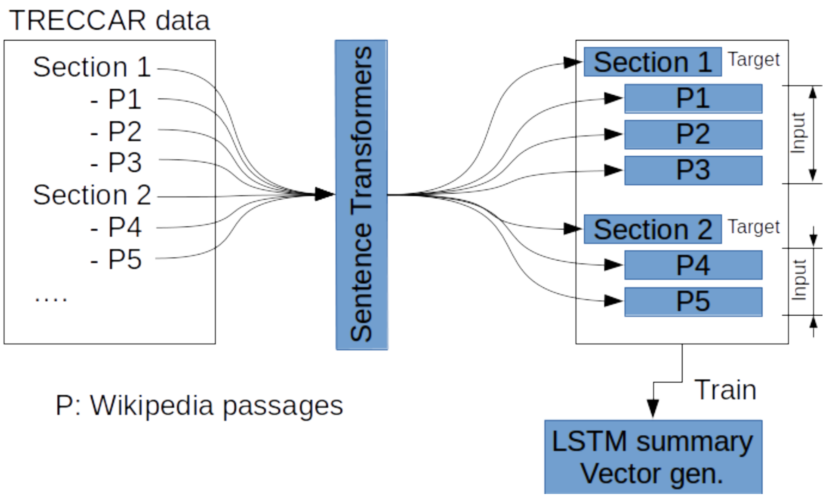

Figure 2: Generating training data from a hierarchical outline of TREC CAR ground truth

Table 1: Submitted methods and their description, n = no. of segments, k = no. of top ranked segments chosen, s = maximum output summary words for each segment

Method	Description
unhtrema1	$k = 3$, $s = 15$, $n = 10$, segment size of 1000 words
unhtrema2	$k = 10$, $s = 15$, $n = 10$, segment size of 1000 words
unhtrema3	$k = 10$, $s = 20$, $n = 100$, segment size of 100 words
unhtrema4	$k = 20$, $s = 20$, $n = 100$, segment size of 100 words

embedding vector of the section heading. The model consists of an LSTM layer with input and output vector modified by linear transformation (https://github.com/nihilistsumo/TREC_podcast).

3 Results

For our experiments, we consider the first 10000 words of each episode to generate our summary. We split these episode texts of 10000 words into n segments of equal lengths ($\frac{10000}{n}$ words each). Each of these segments are ranked using our summary vector generation model described earlier. From this ranking, we pick the top k segments and generate summary text using a GAN model of maximum s words. These generated summary lines are concateneated to obtain our final summary. We experiment with four variations of our model based on different choice of n, k, s. These variations are described in Table 1.

The summaries are evaluated qualitatively by the NIST assessors which

Table 2: Average ROUGE-L, precision and F1 scores

Method	Average ROUGE-L	Precision	F1
unhtrema1	0.061	0.156	0.076
unhtrema2	0.089	0.131	0.087
unhtrema3	0.134	0.089	0.091
unhtrema4	0.179	0.069	0.085

is the official evaluation for the task. According to the qualitative assessment, *unhtrema4* generates the best summaries. This suggest that, smaller segment size leads to improved summarization quality. Also, it is evaluated in terms of ROUGE-L metric based on the similarity between the generated summaries and a episode description text. The average ROUGE-L scores along with precision and F1 scores are reported in Table 2.

4 Conclusion

We present a supervised model that leverages the generative power of a recent implementation of GAN model to produce summaries of large podcast audio transcripts. Instead of using the large transcript text as the input to the GAN model, we generate summaries of only the salient parts of the transcript. We train a supervised extractive model to select salient text segments. Our results show that smaller segment size improves the summarization quality.

References

[1] A. Clifton, S. Reddy, Y. Yu, A. Pappu, R. Rezapour, H. Bonab, M. Eskevich, G. J. F. Jones, J. Karlgren, B. Carterette, and R. Jones, “100,000 Podcasts: A Spoken English Document Corpus,” in *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

[2] R. Jones, B. Carterette, A. Clifton, M. Eskevich, G. Jones, J. Karlgren, A. Pappu, S. Reddy, and Y. Yu, “Overview of the TREC 2020 Podcasts Track,”

[3] Y.-S. Wang and H.-Y. Lee, “Learning to encode text as human-readable summaries using generative adversarial networks,” *arXiv preprint arXiv:1810.02851*, 2018.

[4] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.

[5] L. Dietz, M. Verma, F. Radlinski, and N. Craswell, “Trec complex answer retrieval overview.,” in *TREC*, 2017.