TED TALK TEASER GENERATION WITH PRE-TRAINED MODELS

Gianluca Vico, Jan Niehues

Maastricht University
Department of Data Science and Knowledge Engineering
Maastricht, The Netherlands

ABSTRACT

While we have seen significant advances in automatic summarization for text, research on speech summarization is still limited. In this work, we address the challenge of automatically generating teasers for TED talks. In the first step, we create a corpus for automatic summarization of TED and TEDx talks consisting of the talks' recording, their transcripts and their descriptions. The corpus is used to build a speech summarization system for the task. We adapt and combine pre-trained models for automatic speech recognition (ASR) and text summarization using the collected data. This initial work shows that is more important to adapt the summarization model to the ASR transcripts than to adapt the ASR model to the talks.

Index Terms— speech summarization, automatic speech recognition, abstractive summarization

1. INTRODUCTION

We investigated the task of speech summarization on the example of TED teaser generation.

The first contribution of this work is a corpus for automatic summarization of TED and TEDx talks [0]. It includes the recordings of the talks, their descriptions as summaries, but also their human-generated transcripts. Furthermore, we provide the tools to generate additional resources. While we are able to create a training corpus for the target task, the amount of data is limited. A promising approach in this condition is to fine-tune pre-trained models. Therefore, we propose a cascade approach to speech summarization by fine-tuning an ASR and text summarization model.

The second contribution is about fine-tuning the cascade model. We show the importance of adapting both the ASR and the summarization component and demonstrate the value of the new dataset. In particular, we observe that fine-tuning the model with documents processed to look similar to the ASR transcripts improves the model more than simply using the original documents.

2. RELATED WORK

A common approach to build a speech summarization system is to pair an ASR model with a text summarization model [1][2][3].

In 2019, Palaskar et al. [3] compared multi-modal models for summarizing How2 videos. One of the considered models consists of a text summarization model with the ASR transcripts of the videos as input. They observed that this model performs worse than a summarization model that uses the ground-truth transcripts.

The next year Vartakavi and Garg [1] propose PodSumm. PodSumm can generate speech-to-text and speech-to-speech extractive summaries of podcasts by using a pre-trained ASR model and by fine-tuning PreSumm [4] on the ASR transcripts to recognise the sentences that summarize the podcasts.

In the same year, Zheng et al. [5] compared different summarization models used for podcast summarization, by using the Spotify podcast dataset [6], a dataset for abstractive summarization.They observed that fine-tuning the model on the ASR transcripts is more effective than fine-tuning on another dataset for abstractive summarization (e.g. CNN/DailyMail [7] [8]) or not fine-tuning at all. The main differences between this last two works and this paper are the text summarization models taken into consideration, and the fact that they have no control over the ASR model used to transcribe the talks.

3. DATASET CREATION

We build a dataset of TED and TEDx talks to train the models. It has three main components: a set of talk recordings, the corresponding transcripts with time annotations and the summaries of the talks. We gathered the data from three different sources: TED's website, TED's page on amara.org [1], and the

G. Vico acknowledges the contribution of the Rotary Club of Alba for publication expenses.

[0] The textual part of the corpus is available at
https://osf.io/gb6ns/?view_only=1eac1a0ce7384ca29af8ee4871b357ff
The code can be found at
https://github.com/GianlucaVico/ted-summarization

[1] TED is no longer using Amara

	Average length		Length ratio
	Document	Summary	
Train	1417 ± 605	58 ± 20	5.07%
Test	1370 ± 655	59 ± 22	5.59%
XSum	431.07 [12]	23.26 [12]	-

Table 1. LAverage length in tokens and standard deviation of the documents and summaries in the dataset. XSUM as comparison

English portion MuST-C [9] [2].

We used the talk's descriptions from www.ted.com as reference summaries and the transcripts are human-generated.

The talks are split with a proportion of 85% for the train set and 15% for the test set. The split is done on the talk level to ensure that, for instance, the ASR model is not trained on a talk used for testing the final cascade model.

Furthermore, we considered two different sets for training the ASR model: one that includes only data from MuST-C and one that uses also talks from the other sources. The first is referred to as MuST-C, while the second as TED+MuST-C.

3.1. Preparing the talks for summarization

We use the length of the summaries and of the documents and their ratio to remove potential erroneous summaries: documents whose length is between 50 and 4096 tokens and whose summaries are between 5 and 256 tokens are included in the dataset, as well as the samples whose length ratio between the summary and the document is lower than 20 %.

Next, we consider the extractive fragments density [10] and the extractive fragment coverage [10], together to the proportion summary's 2-grams that are also in the documents to estimate if the information contained in the summary is also in the document. The extractive fragments are a sequence of tokens shared between the summary and the document. The coverage measures the proportion of the summary's tokens that are in an extractive fragment, while the density measures "the average length of the extractive fragments to which each word in the summary belongs"[10]. Zhang et al. used these metrics to describe the dataset they used when developing Pegasus [11]. We keep the documents with at least 50% of extractive fragments coverage and at least 10 % of tokens in common between the summary and the document. In this way, the documents with the lowest extractive fragments density are also removed.

Table 1 reports the length of the transcribed talks and their summaries. If we compare this dataset to XSUM [12], used to pre-train Pegasus by its authors [11], it is possible to note that TED talks are generally longer than the XSUM documents. Similarly for the summaries.

[2]We used the English-to-Czech set of MuST-C v1.2 which is available at https://ict.fbk.eu/must-c/

Set	Avg. Len. (Train)	Avg. Len. (Test)
MuST-	$6.47\,\mathrm{s}$	$6.15\,\mathrm{s}$
TED+MuST-C	$3.53\,\mathrm{s}$	$2.92\,\mathrm{s}$

Table 2. Average length of the audio fragments in the train and test set.

Moreover, the size of this dataset is extremely small: 4168 talks while XSUM contains 226711 documents. However, Passali et al. [13] show that a corpus of about 2000 documents is sufficient to adapt Pegasus pre-trained on XSUM.

3.2. Preparing the recordings for ASR

In total, 739 hours of talks are retrieved, of which 234 hours from MuST-C. When segmenting the talks according to the transcripts' annotations, the talks from MuST-C tend to have longer fragments than the other talk as shown in Table 2. Note that MuST-C aligns the transcripts and the audio fragments with automatic systems, while the additional talks rely on the time annotations in the ground-truth transcripts created by human annotators.

3.3. Summarization from the audio recordings

In the end, the talks with both the summary and the audio recording are 2413 for training and 474 for testing. This is due to talks with missing recordings or missing transcripts. These talks could be used to train an end-to-end speech summarization model. Though, in this paper, we use them only to test the cascade model. The recording is considered silent when the signal is -27 dBFS or lower for more than $1\,\mathrm{s}$. These are arbitrary values, therefore they need to be fine-tune and different talks may require different thresholds.

4. SPEECH SUMMARIZATION MODEL

We propose a speech summarization model that uses a loosely coupled cascade model of a speech recognition and summarization model. The models use subtask training, where each component is trained independently. The talks are first segmented into audio fragments, which are transcribed using an ASR system. Finally, the transcripts are summarized using an abstractive summarization model. This way of combining an ASR model and a summarization model has been used by similar works in speech summarization [1][3].

In order to generate good quality summaries, we address several challenges: First of all, although we create a new data set for this task, the provided data is still limited. Therefore, we investigated the use of pre-trained models to achieve a good quality. Secondly, cascaded models suffer from error propagation. We address this issue by increasing the robustness of the models using noisy training data.

4.1. Individual models

As a baseline for the ASR component, we used a pre-trained wav2vec 2.0. In a second step, these models are adapted to the task by fine-tuning the baseline model on the task. We produced two models: one is trained only on MuST-C, while the other is trained on TED+MuST-C.

Similar, we used the pre-trained Pegasus for summarization. Again, we adapt the model to the task by fine-tuning the model on the newly created data set.

4.2. Integration

Secondly, we investigated how we can improve the model to better handle the error propagation. We address this challenge by training the model on noisy data that is similar to the ASR output.

First, we create a model by fine-tuning Pegasus on the clean ground-truth transcripts. We remove HTML tags, annotations, and non-English characters. This is indicated as **clean text** in this paper.

Next, we remove the punctuation and capitalization from the text and convert the numbers to words. This kind of text mimics the transcript of an ASR model, but it does not include any ASR error. We indicate these documents as **ASR-like text** and we train a second Pegasus model on them.

Then, we use the actual transcripts from the fine-tuned wav2vec 2.0 to fine-tune Pegasus. These are indicated as **ASR transcripts**.

4.3. Audio segmentation

During the training of the ASR model, we segmented the talk recordings according to the time annotations from the ground-truth transcripts. This method makes it possible to train the model on the pair of recordings and transcripts.

However, when testing the cascade model, we used silence as the criterion to segment the talks. The advantage of this method it is closer to the real system where this information is not available.

5. EXPERIMENTS

5.1. Integration strategies

In a first experiment, we analyse the influence of the different integration strategies by comparing the performance of the Pegasus model trained on different types of training data.

We compare the ROUGE F1 scores that the pre-trained Pegasus and the three fine-tuned Pegasus models achieve on the test set of talks transcribed by the fine-tune ASR model. The talks are transcribed using the manual split. The model used is *google/pegasus-xsum* from the Hugging Face Model Hub.We used the default settings that use 512 tokens as input and greedy decoder to generate the summaries and we truncate them at 256 tokens. All the models use Adafactor [14] as optimizer and 5×10^{-5} as learning rate.

Fine-tuning	ROUGE-1	ROUGE-2	ROUGE-l
No	16.59 %	2.33 %	13.55 %
Clean text	20.64 %	5.56 %	17.57 %
ASR-like text	23.23 %	7.24 %	19.95 %
ASR transcripts	**25.10 %**	**8.34 %**	**21.52 %**

Table 3. The pre-trained Pegasus and the fine-tuned Pegasus models are compared on the talks transcribed by wav2vec 2.0 trained on MuST-C. The models are tested on the same test talk, however, they use different train sets.

5.2. Individual components

First, the quality of the ASR is assess with the word error rate (WER) and the character error rate (CER). Next, we fine-tune wav2vec 2.0 on the manually split audio fragments and the corresponding transcripts to see if this dataset is suitable for ASR and to reduce the errors being propagated to the summarization model. Two models are trained: one using only on MuST-C, while the other is trained on TED+MuST-C. The quality of the data is demonstrated by comparing the performance of the two models and by showing that both models improve the original pre-trained model. The optimizer is AdamW [15] with a learning rate of 5×10^{-5}. For simplicity, we used the default greedy decoder when generating the transcripts. In contrast to the experiment illustrated in Section 5.1, here the audio recordings are split automatically when silence is detected.

Then we investigate the influence of the different ASR models and the different summarization models on the final performance. We take into consideration models that uses wav2vec 2.0 fine-tuned on TED+MuST-C and without fine-tuning, and that uses Pegasus without fine-tuned on the ASR transcripts and fine-tuning. These models are tested on the same set of talks recordings and reference summaries.

Next, we compare the performance of the best model to the performance where talks are segmented manually. In a second experiment, we replace the segmentation and ASR component with the manual transcript of the talk.

Finally, we inspect the summaries of a sample talk to have insight into the issues and strengths of the cascades models.

6. RESULTS

6.1. Integration strategies

In table 3, we report the ROUGE F1 scores of the fine-tuned summarization models. It is clear that fine-tuning on the actual transcripts gives the best performance when the model is tested on the same kind of text. Also, the scores obtained by

Fine-tuning	WER	CER
No	24.45%	13.67%
MuST-C	**15.28%**	**7.65%**
TED+MuST-C	15.30%	8.02 %

Table 4. WER and CER of wav2vec 2.0 on the test talks when splitting the recordings manually. The model has been fine-tuned on different sets.

wav2vec 2.0	Pegasus	ROUGE-1/2/l
No fine-tune	No fine-tune	16.57%/2.22%/13.19%
No fine-tune	ASR Tr.	24.42%/6.43%/20.22%
TED+MuST-C	No fine-tune	16.52%/1.99%/12.83%
TED+MuST-C	ASR Tr.	**24.85%/6.65%/20.46%**

Table 5. Comparison of the cascade models. The baseline model has not been fine-tuned. The splitting is automatic.

the models are related to how similar to the ASR transcripts the training documents are. The difference in the results between Pegasus fine-tuned on the ASR-like text and Pegasus fine-tuned on the ASR transcripts is due to the fact that the second model is trained on the ASR errors, while the first model encountered this kind of errors only during the testing.

6.2. Individual components

Table 4 reports the WER and CER of wav2vec 2.0 on the MuST-C test set. The fine-tuned models generate more accurate transcripts than the original pre-trained wav2vec 2.0, showing that wav2vec 2.0 can be adapted to this task. Moreover, the model trained on TED+MuST-C achieves similar results to the one trained only on MuST-C. Therefore, MuST-C and TED+MuST-C have a similar quality, but that the additional data does not improve the model.

Table 5 reports the ROUGE F1 scores that the cascade models obtained in the test set. It appears that fine-tuning both the ASR component and the summarization component gives the highest ROUGE F1 scores. However, fine-tuning only the summarization part does not cause a drastic drop in the performance. Hence, it is crucial to adapt the summarization part. When wav2vec 2.0 is fine-tuned, the decrease in the WER does not cause a similar increase in the ROUGE F1 scores.

From table 6, it is possible to see that using automatic transcript and automatic splitting (how the model is used in a real application), is slightly worse than the model with manual splitting. However, the automatic split does not take into account the fragment's length, which may not be suitable for the model. This can degrade the model's performance.

Table 7 shows summaries obtained from the original pre-trained models, used as a baseline, and from a fine-tuned ones. The talk by Nicholas Negroponte at TED2006 [16] is about an educational project and the name of the speaker is mentioned

Transcripts	ROUGE-1	ROUGE-2	ROUGE-l
Clean text	19.12 %	4.25 %	15.95 %
ASR, manual split	**25.10 %**	**8.34 %**	**21.52 %**
ASR, automatic split	24.53 %	6.67 %	20.34 %

Table 6. Average ROUGE F1 scores of Pegasus on the test talks with the recordings. We used the ground-truth transcripts and the transcripts generated by wav2vec 2.0 fine-tuned on MuST-C

Reference
Nicholas Negroponte, founder of the **MIT Media Laboratory**, describes how the **One Laptop Per Child** project will build and distribute the **"$100 laptop."**
Baseline model
It's my last day as a *professor of education* at ted ona and i'm going to tell you about one of the things i've been doing for a year and a halfe any so i'm going to tell you why doing it and then I'm going to pass around
Cascade model
MIT Media Lab founder Barry Schwartz talks about his plan to give **one laptop per child** in the US – and how it's going to change the way kids *learn*.

Table 7. Sample summaries of a talk by Nicholas Negroponte at TED2006 [16]. In *italic* the reference to education; in **bold** the correctly reported named entities, while underlined the incorrect ones. The baseline model uses the pre-trained components. The cascade model is trained on TED+MuST-C, the ASR transcripts and uses the automatic split.

only in the reference summary. As we can see, the baseline model can grasp the main topic of the talk (i.e. education), but it is not able to report additional information. The fine-tuned cascade model, is able to report the name of the project and the institute. It fails to identify the name of speaker since it is not mentioned in the talk.

7. CONCLUSION

In this paper, we present a speech summarization model made up of state-of-the-art ASR and summarization components for generating teasers for TED talks and the dataset used to train it. We show that the data can be used to train such a model. Furthermore, fine-tuning both the ASR system and the text summarization system gives higher ROUGE F1 scores than fine-tuning only the summarization part. However, the decrease of the WER of the ASR model does not correspond to a similar increase in ROUGE F1 scores.

Future works include using a language model and beam search decoding for both components, and a summarization model for longer sequences, like BigBirdPegasus [17].

8. REFERENCES

[1] Aneesh Vartakavi and Amanmeet Garg, "Podsumm - podcast audio summarization," *CoRR*, vol. abs/2009.10315, 2020.

[2] András Beke and György Szaszák, "Automatic summarization of highly spontaneous speech," in *Speech and Computer*, Andrey Ronzhin, Rodmonga Potapova, and Géza Németh, Eds., Cham, 2016, pp. 140–147, Springer International Publishing.

[3] Shruti Palaskar, Jindřich Libovický, Spandana Gella, and Florian Metze, "Multimodal abstractive summarization for how2 videos," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019, pp. 6587–6596, Association for Computational Linguistics.

[4] Yang Liu and Mirella Lapata, "Text summarization with pretrained encoders," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3730–3740, Association for Computational Linguistics.

[5] Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan, "A baseline analysis for podcast abstractive summarization," *CoRR*, vol. abs/2008.10648, 2020.

[6] 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*, Barcelona, Spain (Online), Dec. 2020, pp. 5903–5917, International Committee on Computational Linguistics.

[7] Abigail See, Peter J. Liu, and Christopher D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017, pp. 1073–1083, Association for Computational Linguistics.

[8] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching machines to read and comprehend," in *NIPS*, 2015, pp. 1693–1701.

[9] Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi, "Must-c: A multilingual corpus for end-to-end speech translation," *Computer Speech & Language*, vol. 66, pp. 101155, 2021.

[10] Max Grusky, Mor Naaman, and Yoav Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, June 2018, pp. 708–719, Association for Computational Linguistics.

[11] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh, Eds. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 11328–11339, PMLR.

[12] Shashi Narayan, Shay B. Cohen, and Mirella Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, Oct.-Nov. 2018, pp. 1797–1807, Association for Computational Linguistics.

[13] Tatiana Passali, Alexios Gidiotis, Efstathios Chatzikyriakidis, and Grigorios Tsoumakas, "Towards human-centered summarization: A case study on financial news," in *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, Online, Apr. 2021, pp. 21–27, Association for Computational Linguistics.

[14] Noam Shazeer and Mitchell Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604, PMLR.

[15] Ilya Loshchilov and Frank Hutter, "Fixing weight decay regularization in adam," 2018.

[16] Nicholas Negroponte, "One laptop per child, [Online Video], Available: `https://www.ted.com/talks/nicholas_negroponte_one_laptop_per_child`," 2006, Accessed: Dec-2021.

[17] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed, "Big bird: Transformers for longer sequences," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 17283–17297, Curran Associates, Inc.