

AUTOMATED AUDIO CAPTIONING USING TRANSFER LEARNING AND RECONSTRUCTION LATENT SPACE SIMILARITY REGULARIZATION

Andrew Koh, Xue Fuzhao, Chng Eng Siong

Nanyang Technological University

ABSTRACT

In this paper, we examine the use of Transfer Learning using Pretrained Audio Neural Networks (PANNs) [1], and propose an architecture that is able to better leverage the acoustic features provided by PANNs for the Automated Audio Captioning Task [2]. We also introduce a novel self-supervised objective, Reconstruction Latent Space Similarity Regularization (RLSSR). The RLSSR module supplements the training of the model by maximizing the similarity between the encoder and decoder embedding. The combination of both methods allows us to surpass state of the art results by a significant margin on the Clotho dataset [3] across several metrics and benchmarks.

1. INTRODUCTION

Automated Audio Captioning is an emerging research area [2] which aims to describe in textual form, the events taking place in the foreground and background of an audio clip. Automated Audio Captioning is the amalgamation of both Audio Processing and Natural Language Processing research, much like Automated Speech Recognition. However, unlike Automated Speech Recognition, Automated Audio Captioning does not require an alignment between the source audio and target text sequence. Furthermore, Automated Audio Captioning requires the model to detect both high level events in the background and the more intricate details that occur in shorter time-frames in the foreground.

Convolutional Neural Networks [1] are a popular choice of encoder for encoding information from the audio input. Convolutional Neural Networks have done well for tasks such as sound event detection, audio tagging, by being able to extract local features using convolutional filters. This has also led to previous work in Automated Audio Captioning also similarly using a single convolutional neural network as the encoder. However, we argue that this is insufficient for Automated Audio Captioning, as this task requires the model to capture not just local events (e.g a loud honk) that occur in narrower timeframes, but also overarching events (e.g. car

engine running in the background) that can span the whole duration of the audio. To combat this, we propose to append the transformer encoder [4] to the convolutional encoder. The transformer encoder consists of several layers of multi-head self attention and feedforward layers, and many variants of the transformer have broken the state of the art on Natural Language Processing benchmarks [5]. We hypothesise that the combination of both the convolutional encoder and transformer encoder, along with transfer learning, will allow the model to perform much better.

Additionally, we propose a simple self-supervised learning objective, Reconstruction Latent Space Similarity Regularization (RLSSR), to regularize the training of model. As the Automated Audio Captioning is arguably less fine-grained and more abstract in its objective compared to Automated Speech Recognition, we argue that it is insufficient to simply use a cross attention mechanism to bridge information from the audio encoder to the text decoder. RLSSR provides a straightforward way to regularize the training of the decoder by reducing the euclidean distance between the reconstruction of the decoder layer in the latent space and the latent output of the encoder.

Our contributions are as follows:

1. We examine the usefulness of the transformer encoder in the encoder, and show that including the transformer encoder is instrumental for the model performance when using transfer Learning.
2. We introduce the novel Reconstruction Latent Space Similarity Regularization (RLSSR) objective and show that simply using this objective allows a model trained from scratch to achieve competitive state of the art results.
3. We demonstrate that using both Transfer Learning and RLSSR in tandem consistently beats state of the art results on the Clotho Dataset.

2. RELATED WORK

Automated Audio Captioning has been tackled by various authors.

This research is supported by ST Engineering Mission Software Services Pte. Ltd under collaboration programme (Research Collaboration No: REQ0149132).

2.1. Audio Captioning Datasets

The Clotho Dataset [3] is by far the dataset most researchers have examined and experimented on. The Clotho Dataset consists of 4981 audio samples of duration from 15 to 30 seconds. Each audio sample is matched to 5 crowdsourced English audio annotations or captions. These captions describe the events in that audio. The Clotho Dataset was created to address the critiques of the larger Audiotocaps dataset [6], which was found to contain biases in the annotation process. There are also niche datasets like the Hospital Scene and Car Scene dataset [7] which are Mandarin annotated.

In this work, we focus only on the Clotho Dataset, as it is the most popular dataset.

2.2. Transfer Learning

Several works has applied transfer learning to Automated Audio Captioning. Fine-tune PreCNN Transformer[8] used a 3 stage pipeline of first pretraining an encoder, then applying transfer learning to the pretrained model by freezing parts of the model for training, and finally finetuning the entire model.

AT-CNN [9], the current best performing model on Clotho, follows a similar approach with a 2 pipeline which first pretrains the CNN10 [1] / CRNN5 [10] encoder on an audio tagging task, and then performs transfer learning on the pretrained model on the audio tagging task. Similarly, the Audio Captioning Transformer [11] also pretrains the encoder on audio tagging before training on audio tagging.

Other approaches uses specialized embeddings for transfer learning. [12] applied transfer learning to a pretrained GPT2 model, which was pretrained in the text modality. This approach also requires several other prerequisites, such as a BERTScore [13] model, and several preprocessing steps for caption retrieval. In another vein, [14] used a combination of pretrained audio embeddings and subject-verb embeddings to generate captions.

In this work, we show that applying transfer learning using publicly available pretrained models in conjunction with the transformer encoder allows for significant performance improvements.

3. PROPOSED METHOD

We use a typical encoder-decoder model in all our experiments. In addition to the encoder-decoder structure, there is an external Reconstruction Latent Space Similarity Regularization (RLSSR) module. A detailed overview of the model is shown in Figure 1.

3.1. Encoder

The encoder consists of a 10 layer CNN (CNN10) and the encoder of the transformer, and takes in log-mel spectrograms as input. CNN10 is adapted from different CNN architectures

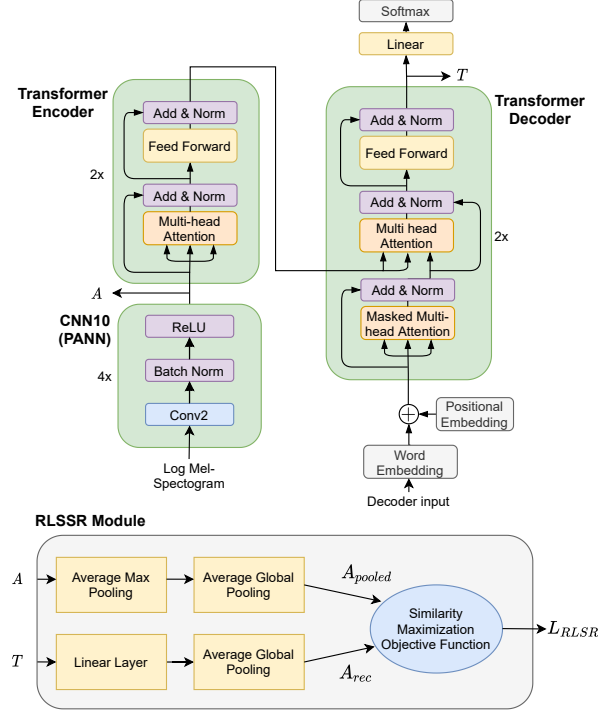


Fig. 1. Detailed Overview of the model. The encoder consists of a CNN10 and a Transformer Encoder, and the decoder consists of a single Transformer decoder. The output of the CNN10, A , and the output embedding of the transformer decoder, T , is passed to the RLSSR module, where the L1/L2 loss is used as objective function in the RLSSR Module.

available in the Pretrained Audio Neural Networks collection [1]. The output of CNN10 is passed to a transformer encoder. The transformer encoder [4] consists of two layers with a hidden size of 192. We hypothesize that the multihead attention mechanism in the transformer encoder is instrumental in learning higher-level representations from the features provided by the CNN. In our experiments, we show that using the transformer encoder in transfer learning provides a huge boost in performance. Where transfer learning is applied, only the CNN10 encoder is preloaded with pretrained weights, while the transformer encoder is always trained from scratch.

3.2. Decoder

The decoder is a typical Transformer decoder with multihead-attention. It takes a sequence of word tokens, along with the output from the encoder, as decoder input and outputs the next most probable word token. In our experiments, we used a 2 layer Transformer decoder with a hidden size of 192. While we did experiment on decoders of different layer sizes, we found that using 2 layers worked the best.

3.3. Reconstruction Latent Space Similarity Regularization Module

The aim of the Reconstruction Latent Space Similarity Regularization (RLSSR) module is to maximize the similarity between the latent space of the decoder and the latent space of the encoder. Intuitively, this module has to reconstruct the latent space of the encoder output in the audio domain, A , from the latent space of the decoder output in the text domain, T . This is done by introducing a very small module of parameters that bridges the decoder output to the encoder.

$$\begin{aligned} A_{rec} &= \text{FFN}(\text{pool}(T)) \\ A_{pooled} &= \text{pool}(A) \\ L_{RLSSR} &= \text{sim}(A_{rec}, A_{pooled}) \end{aligned} \quad (1)$$

The RLSSR module takes the output text embedding, T from the last layer of the decoder and the output audio embedding, A , from the CNN from the encoder as input. A average max pooling is applied to A to extract the most prominent features to be reconstructed. After which, an average global pooling is applied to standardize the length of the audio features to obtain A_{pooled} . The text embeddings T are first passed through a linear layer and then an average global pooling layer to obtain the reconstructed representation from the text, A_{rec} . Finally, a similarity metric is used to minimize the distance between A_{pooled} and A_{rec} . These operations are shown in the Equation 1. While we did try other similarity metrics such as cosine similarity, Huber loss [15], we found that the L1 distance, L2 distance worked the best as the similarity metric.

$$L = \alpha L_{ce} + \beta L_{RLSSR} \quad (2)$$

The model minimizes the sum of the cross entropy loss from the generated caption, along with the loss from the RLSSR module. We tried various combinations of α and β and found that weighing both loss values equally worked the best.

4. EXPERIMENTAL DETAILS

4.1. Data

As mentioned in Section 2.1, the experimental dataset in use is the Clotho dataset. We try to follow as closely as possible the experimental settings of previous work. These are the specific details. The raw audio files are first preprocessed into log mel-spectrograms. We use 64 Mel-bands, sampling rate of 44100, FFT window length of 1024, and a hop size of 512.

4.2. Training and Evaluation

These are our training hyperparameters and settings. We use a batch size of 64 with a gradient accumulation steps of 4 for 200 epochs with early stopping. Learning rate is set to

3×10^{-4} and SpecAugmentation [16] is applied to all log mel-spectrogram inputs as an data augmentation tactic. We do not apply label smoothing. The weights of the cross entropy loss and the similarity loss from the RLSSR module is weighted and optimized equally. Where transfer learning is applied, we used the weights of the CNN10 model¹ which scored a mAP of 0.380 on the Audioset dataset.

For inference, we perform beam search with beam size of 4 for decoding. As the Clotho dataset has 5 reference captions for each audio, we follow the COCO image captioning evaluation process [17] to evaluate the generated caption. The BLEU_n scores [18] calculates the degree of n-gram overlap between the generated and reference caption. ROUGE_L [19] measures the longest common sequence and gives a score base on the similarity. METEOR [20] is said to better correlate with human annotators and uses a harmonic mean of unigram precision and recall to score the generated sequence. CIDEr [21] also measures similarity using the average cosine similarity between the candidate sentence and the reference sentences. Like METEOR, SPICE [22] is also said to correlate with human judgement more by using semantic propositional content and is less reliant on n-gram overlaps. Finally, SPIDEr is a linear combination of both SPICE and CIDEr. We use all the aforementioned metrics for evaluation.

5. EXPERIMENTAL RESULTS

5.1. Usefulness of the Transformer Encoder

We compare the effectiveness of including the transformer encoder in the encoder. The results are shown in Table 1. Experiments show that using pretrained weights along with the transformer encoder outperformed all the other models by a significant margin. On the other hand, not including the transformer encoder along with pretrained weights resulted in only marginal improvements compared to not using pretrained weights. It is also interesting to note that, without pretrained weights, not using the transformer encoder boosted the performance. We believe this is because the size of the Clotho dataset is insufficient to meaningfully train a transformer encoder from scratch. However, using pretrained weights in the convolutional encoder provided the initialization required for the transformer encoder to learn much better.

5.2. Effect of the Reconstruction Latent Space Similarity Regularization Module

We examine the impact of the RLSSR Module. Results are shown in Table 2. From our experiments, using the L1 loss outperforms the L2 loss consistently. However, using either the L1 loss or L2 loss in RLSSR module outperforms the

¹This pretrained model is available publicly for download at https://github.com/qiuqiangkongaudioset_tagging_cnn

Model	BLEU ₁	BLEU ₂	BLEU ₃	BLEU ₄	ROUGE _L	METEOR	CIDEr	SPICE	SPIDEr
Base (with transformer encoder)	0.516	0.330	0.219	0.142	0.348	0.152	0.319	0.102	0.210
Base (with transformer encoder) + PANN	0.542	0.363	0.248	0.163	0.362	0.161	0.369	0.108	0.242
Base - transformer enc	0.523	0.337	0.227	0.151	0.353	0.153	0.332	0.102	0.211
Base - transformer enc + PANN	0.538	0.350	0.235	0.153	0.362	0.158	0.348	0.106	0.227

Table 1. Ablation experiments to determine usefulness of the transformer encoder. Base refers to our default model, consisting of the convolutional encoder, transformer encoder and transformer decoder. PANN refers to loading the pretrained weights for transfer learning.

Model	BLEU ₁	BLEU ₂	BLEU ₃	BLEU ₄	ROUGE _L	METEOR	CIDEr	SPICE	SPIDEr
Base	0.516	0.330	0.219	0.142	0.348	0.152	0.319	0.102	0.210
Base + L2 loss	0.518	0.335	0.228	0.152	0.352	0.151	0.326	0.102	0.214
Base + L1 loss	0.515	0.338	0.234	0.159	0.351	0.151	0.325	0.098	0.212
Base + PANN	0.542	0.363	0.248	0.163	0.362	0.161	0.369	0.108	0.242
Base + PANN + L2 loss	0.552	0.370	0.251	0.166	0.369	0.163	0.375	0.112	0.240
Base + PANN + L1 loss	0.551	0.369	0.252	0.168	0.373	0.165	0.38	0.111	0.246

Table 2. Results of using the RLSSR module. L1 and L2 loss refers to the distance metric used to optimize the RLSSR module. PANN refers to applying transfer learning using the pretrained weights from the pretrained audio neural network.

Model	BLEU ₁	BLEU ₂	BLEU ₃	BLEU ₄	ROUGE _L	METEOR	CIDEr	SPICE	SPIDEr
Baseline [3]	0.389	0.136	0.055	0.015	0.262	0.084	0.074	0.033	0.054
Fine-tune PreCNN Transformer [8]	0.534	0.343	0.230	0.151	0.356	0.160	0.346	0.108	0.227
AT-CNN10 [9]	0.556	0.363	0.242	0.159	0.368	0.169	0.377	0.115	-
Base + PANN + L1 loss	0.551	0.369	0.252	0.168	0.373	0.165	0.380	0.111	0.246

Table 3. Comparison with other state of the art. Our model consistently beats the previous state of the art on the BLEU_n, ROUGE_L, and CIDEr scores.

Reference	base-enc	base+PANN+L1
water flowing at a constant rate then slows down to a drip	water is running into a sink or tub	water is running at a constant rate in a sink
a person is arranging objects on a table and pouring water into a glass	someone is washing dishes and dishes in a sink	water is being poured into a container while a person speaks
a man slides the file across the wood tapping it a few times at the end	a door is opened and closed several times	a hard object is being dragged across a hard surface

Table 4. Table containing the generated examples of our models, along with the reference captions.

models which did not use the RLSSR module. Models which did not use transfer learning but used the RLSSR module outperformed the base model by a significant margin and even performed competitively compared to state of the art models using transfer learning. When using transfer learning, the performance of our models increased even further. This attests to the RLSSR module being an useful objective to optimize in order to regularize the model.

5.3. Comparison with previous work

The comparison with other work is shown in Table 3. Our best model beats the best performing model, AT-CNN10 [9], on all metrics except the METEOR and SPICE metric. In addition to being self-supervised and simple to train, our approach

makes use of publicly available pretrained models, while AT-CNN10 involves a two stage transfer learning process trained from scratch. We have also included some examples of our generated sequences at Table 4.

6. CONCLUSION

This work introduces the benefit of the transformer encoder when applying transfer learning. We have also proposed the self-supervised Reconstruction Latent Space Similarity Regularization objective, which acts as regularization during training. We have shown that combining both RLSSR and transfer learning allows us to beat state of the art on the Clotho dataset by a significant margin across many metrics.

7. REFERENCES

- [1] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” 2020.
- [2] Konstantinos Drossos, Sharath Adavanne, and Tuomas Virtanen, “Automated Audio Captioning with Recurrent Neural Networks,” *arXiv:1706.10006 [cs]*, Oct. 2017, arXiv: 1706.10006.
- [3] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen, “Clotho: An Audio Captioning Dataset,” *arXiv:1910.09387 [cs, eess]*, Oct. 2019, arXiv: 1910.09387.
- [4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019.
- [6] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim, “AudioCaps: Generating Captions for Audios in The Wild,” 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)*, Minneapolis, Minnesota, June 2019, pp. 119–132, Association for Computational Linguistics.
- [7] Xuenan Xu, Heinrich Dinkel, Mengyue Wu, and Kai Yu, “Audio Caption in a Car Setting with a Sentence-Level Loss,” *arXiv:1905.13448 [cs, eess]*, Oct. 2020, arXiv: 1905.13448.
- [8] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, “Audio captioning based on transformer and pre-trained CNN,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, Nov. 2020, pp. 21–25.
- [9] Xuenan Xu, Heinrich Dinkel, Mengyue Wu, Zeyu Xie, and Kai Yu, “Investigating local and global information for automated audio captioning with transfer learning,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 905–909.
- [10] Heinrich Dinkel and Kai Yu, “Duration robust weakly supervised sound event detection,” *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020.
- [11] Xinhao Mei, Xubo Liu, Qiushi Huang, Mark D. Plumbley, and Wenwu Wang, “Audio captioning transformer,” 2021.
- [12] Yuma Koizumi, Yasunori Ohishi, Daisuke Niizumi, Daiki Takeuchi, and Masahiro Yasuda, “Audio Captioning using Pre-Trained Large-Scale Language Model Guided by Audio-based Similar Caption Retrieval,” *arXiv:2012.07331 [cs, eess]*, Dec. 2020, arXiv: 2012.07331.
- [13] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, “Bertscore: Evaluating text generation with bert,” 2020.
- [14] Ayşegül Özkaya Eren and Mustafa Sert, “Audio Captioning using Gated Recurrent Units,” *arXiv:2006.03391 [cs, eess]*, Jan. 2021, arXiv: 2006.03391.
- [15] Ross Girshick, “Fast r-cnn,” 2015.
- [16] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” *Interspeech 2019*, Sep 2019.
- [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, “Microsoft coco: Common objects in context,” 2015.
- [18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, USA, 2002, ACL ’02*, p. 311–318, Association for Computational Linguistics.
- [19] Chin-Yew Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, Barcelona, Spain, July 2004, pp. 74–81, Association for Computational Linguistics.
- [20] Michael Denkowski and Alon Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.
- [21] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh, “Cider: Consensus-based image description evaluation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.
- [22] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould, “Spice: Semantic propositional image caption evaluation,” 2016.