IRIT-UPS DCASE 2020 AUDIO CAPTIONING SYSTEM

Technical Report

Thomas Pellegrini

IRIT (UMR 5505), Université Paul Sabatier, CNRS, Toulouse, France

ABSTRACT

This technical report is a short description of the sequence-to-sequence model used in the DCASE 2020 task 6 dedicated to audio captioning. Four submissions were made: i) a baseline one using greedy search, ii) beam search, iii) beam search integrating a 2g language model, iv) with a model trained with a vocabulary limited to the most frequent word types (1k words instead of about 5k words).

Index Terms— Audio captioning, Sequence-to-sequence models, listen-attend-caption

1. INTRODUCTION

The IRIT-UPS approach to the audio captioning challenge task of DCASE 2020 (Task 6) is a Listen-Attend-Spell architecture [1], that could be better renamed "Listen-Attend-Tell", since we do audio captioning and not speech recognition.

My model implementation in PyTorch is open source and publicly available to facilitate reproducibility[1]. The pretrained weights of the two models are also available[2,3].

2. MODEL DESIGN

In this section, the model architecture and design choices are described.

2.1. Audio data and pre-processing

The challenge organizers provided the participants with Clotho, an audio dataset built for audio captioning purposes [2]. Clotho is comprised of a development split of 2893 audio clips with 14465 captions, an evaluation split of 1045 audio clips with 5225 captions, and a testing split of 1043 audio clips with 5215 captions. We had full access to the development and evaluation subsets but not to the test split, for which participants could submit up to four predictions to be scored.

The pre-processing scripts given by the clotho providers were used: 64 log mel-band energies are extracted fro the audio recordings with a Hamming window of 46 ms length with 50% overlap.

2.2. Model architecture

Two models with the same architecture were used to generate the four submissions to the challenge: one using an output layer with 4367 units, corresponding to the total number of word types of the dataset, one with only 1384 units, corresponding to the word types occurring at least 10 times in the dev subset. We will refer to the two models as M1 and M2, respectively.

The architecture is the following:

- Encoder: a dropout layer on the input layer ($p = 0.1$), a Bi-directional LSTM (BLSTM) with 128 cells, followed by two pyramidal BLSTM (pBLSTM) layers of 128 cells each and a time resolution reduction by a factor two, with dropout ($p = 0.1$) after each layer. The output of the two stacked pBLSTM layers is fed into two different linear layers of 64 units each to output key and value tensors of size T, B, h (time, batch size, hidden size).
- Decoder: two stacked LSTM layers of 128 and 64 units, respectively. At each time step, their input is the value outputted by the encoder, multiplied by attention weights, concatenated with the embedding of either the prediction at the preceding time step, or the ground-truth word if teacher forcing is used. The predictions are made with a 4367-sized (or 1384-sized) linear layer.
- The attention module is a simple dot-product operation between the key outputted by the encoder and the query generated at each time step by the decoder. The query is generated by a linear layer (64 units) on the output of the second decoder LSTM layer.

Masking was used both in the attention module and the computation of the cross-entropy loss.

A lot of variants were compared for the encoder architecture, varying the time resolution factor (2, 4 or 8) and the number of stacked pBLSTM layers (1,2,3). The best configuration was found to be two stacked pBLSTM layers and a TR factor two.

The models with the best performance on the evaluation subset were obtained after training for 40 epochs for M1 and 50 epochs for M2. The Adam update rule was used with an initial learning rate of 0.0005, on minibatches of size 64. Teacher forcing with 0.98 chance was used during training.

2.3. Length normalized beam search and language modeling

At inference time, greedy and beam search decoders were compared. For beam search, normalizing the log-scores by the number of words of the predicted captions was found to help:

$$\log p(w_1, \ldots, w_L|x) = \frac{1}{L^\alpha} \sum_{i=1}^{L} \log p(w_i|x, w_1, \ldots, w_{i-1})$$

α has a great impact on the system output, favoring shorter or

Thanks to ANR agency for funding.

[1] https://github.com/topel/listen-attend-tell

[2] https://zenodo.org/record/3893974

[3] https://zenodo.org/record/3895774

longer sentences. The best value on the evaluation subset and the metrics of interest was found to be $\alpha = 1.2$.

Adding the contribution of a bi-gram language model (2g LM) was also tested. The LM with Kneser-Ney smoothing was trained using the SRILM toolkit, on the 5225 sentences of the dev subset. The perplexity of the evaluation subset was 44.2 with this LM and 44.5 with a pruned version of this LM, comprised of 42790 bi-grams. We used the pruned LM since no difference in performance was observed. We tested a 3g LM but the 2g LM performed best.

The LM contribution is introduced with a λ weight as follows. A default $\lambda = 0.5$ has been used.

$$\log p(w_1, \ldots, w_L|x) = \frac{1}{L^\alpha} \sum_{i=1}^{L} \log p(w_i|x, w_1, \ldots, w_{i-1}) + \lambda \log p(w_i|w_{i-2}, w_{i-1})$$

When using an LM, length normalization was found to degrade performance, so $\alpha = 0$ was used in this case.

3. RESULTS

Table 1 shows the results obtained with four configurations, corresponding to the four submissions to the challenge.

Configuration 1,2,3 were obtained with M1, config. 4 with M2. The best results were obtained with M1, and beam search with a beam size of 25 and length normalization. Using a 2g LM lead to worse performance, although more experiments are needed to fine-tune the LM weight.

Table 1: Results on the evaluation subset of the DCASE development dataset. Larger scores are better.

System	1 (M1)	2 (M1)	3 (M1)	4 (M2)
Vocab	4367	4367	4367	1384
Search algo	greedy	beam (25)	beam (10)	beam (25)
α (1.2)	✗	✓	✗	✓
2g LM (0.5)	✗	✗	✓	✗
BLEU1	0.436	0.430	0.426	0.415
BLEU2	0.234	0.248	0.247	0.230
BLEU3	0.138	0.160	0.157	0.143
BLEU4	0.076	0.096	0.094	0.085
ROUGEL	0.124	0.133	0.112	0.125
METEOR	0.301	0.305	0.283	0.298
CIDEr	0.140	0.169	0.165	0.162
SPICE	0.072	0.079	0.063	0.071
SPIDEr	0.106	**0.124**	0.114	0.116

4. ACKNOWLEDGMENT

This work was granted access to the HPC resources of the CALMIP supercomputing center under the allocation 2020-P20022. It was partially supported by the Agence Nationale de la Recherche LUDAU (Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning) project (ANR-18-CE23-0005-01), and has benefitted from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement ANR-19-PI3A-0004.

5. REFERENCES

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[2] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020. [Online]. Available: https://arxiv.org/abs/1910.09387