IS MY AUTOMATIC AUDIO CAPTIONING SYSTEM SO BAD? SPIDEr-max: A METRIC TO CONSIDER SEVERAL CAPTION CANDIDATES

Etienne Labbé, Thomas Pellegrini, Julien Pinquier

IRIT, Université Paul Sabatier, CNRS, Toulouse, France
{etienne.labbe, thomas.pellegrini, julien.pinquier}@irit.fr

ABSTRACT

Automatic Audio Captioning (AAC) is the task that aims to describe an audio signal using natural language. AAC systems take as input an audio signal and output a free-form text sentence, called a caption. Evaluating such systems is not trivial, since there are many ways to express the same idea. For this reason, several complementary metrics, such as BLEU, CIDEr, SPICE and SPIDEr, are used to compare a single automatic caption to one or several captions of reference, produced by a human annotator. Nevertheless, an automatic system can produce several caption candidates, either using some randomness in the sentence generation process, or by considering the various competing hypothesized captions during decoding with beam-search, for instance. If we consider an end-user of an AAC system, presenting several captions instead of a single one seems relevant to provide some diversity, similarly to information retrieval systems. In this work, we explore the possibility to consider several predicted captions in the evaluation process instead of one. For this purpose, we propose SPIDEr-max, a metric that takes the maximum SPIDEr value among the scores of several caption candidates. To advocate for our metric, we report experiments on Clotho v2.1 and AudioCaps, with a transformed-based system. On AudioCaps for example, this system reached a SPIDEr-max value (with 5 candidates) close to the SPIDEr human score of reference.

***Index Terms*—** audio captioning, evaluation metric, beam search, multiple candidates

1. INTRODUCTION

Automated Audio Captioning (AAC) is the task, in which a system takes an audio signal as input and provides a short description of its content using natural language. AAC could be useful for hearing-impaired people, in machine-to-machine interaction, surveillance and information retrieval in general. In the last few years, the research community has developed a keen interest in AAC, in particular thanks to the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenges and Workshops [1], which have provided datasets and benchmarks for this task.

Most AAC systems use deep neural networks with a sequence-to-sequence encoder-decoder architecture, to build a semantic audio representation and generate a valid sentence as output [1]. They rely on models pretrained on large-scale datasets, to solve the data scarcity issue in AAC [2, 3, 4].

In this work, we are interested in the evaluation of AAC systems. AAC evaluation borrows metrics from machine translation and image captioning, and consists of comparing a candidate caption to one or several manually produced captions of reference. Since evaluating text generated automatically is a difficult problem, several metrics are used in combination. We investigate in particular the SPIDEr metric [5], a short name used to designate the average of two metrics called Consensus-based Image Description Evaluation (CIDEr) [6] and Semantic Propositional Image Caption Evaluation (SPICE) [7]. SPIDEr is used, for instance, in the DCASE yearly challenges to rank the participant AAC systems [2].

In this paper, we report experiments using the AAC system we developed to participate in the DCASE 2022 AAC task. Like most AAC systems, we use a beam search decoder that allows to generate several candidate captions. The most likely one is used to compute the SPIDEr score of our system. A strong limitation of SPIDEr, in our opinion, is that only one caption candidate is considered for evaluation. As we shall illustrate in this paper, two correct captions that differ by a single word may have very different SPIDEr scores, if one of the words happens to be in the caption(s) of reference. To overcome this issue, we propose a metric that we call SPIDEr-max, which takes into account multiple candidates for a single audio recording.

2. METRICS

In the literature, most AAC systems are evaluated using the CIDEr, SPICE or SPIDEr metrics. These metrics come from the field of image captioning and evaluate a single candidate caption against a reference set.

2.1. CIDEr

CIDEr [6] is a metric based on the TF-IDF (term frequency-inverse document frequency) scores of each n-gram of the candidate and reference sentences. TF-IDF is used to give a higher weight to infrequent n-grams and lower weight to frequent n-grams.

The CIDEr metric calculation starts by stemming all the words and compute all the n-grams of size 1 to N across all candidates and references. The frequency of each n-gram in references are used to compute TF-IDF of all captions. This means that the score of each candidate does not only depend on its corresponding references, but also on all the other references of the corpus being evaluated. Then, the TF-IDF scores are vectorized and used to compute cosine similarity between the candidate and each reference. The similarities are rescaled by a factor of 10 and averaged across the references to get the final score of the candidate. All the scores are averaged again to get the global score on a dataset.

The CIDEr-D metric is a more robust version of CIDEr supposed to be closer to human judgement. It removes the stemming operation to take into account the tense and plural of words, adds

[1] http://dcase.community/

[2] http://dcase.community/challenge2022/

a penalty factor, and limits the maximum occurrence of candidate n-grams to penalize longer repetitive sentences. The penalty is multiplied by a similarity measure based on the length of the candidate c and the reference r:

$$\text{Penalty}(c, r) = \exp\Big(-\frac{(|c| - |r|)^2}{2\sigma^2}\Big) \tag{1}$$

Some AAC papers do not specify which version of CIDEr they use, but in this paper we report CIDEr-D scores as used in the DCASE challenge. We use the default settings of CIDEr-D with the maximum n-gram size N set to 4, and the hyperparameter σ used for the penalty set to 6.

2.2. SPICE

SPICE [7] attempts to extract the semantic content of a sentence. Sentences are used as input to a Probabilistic Context-Free Grammar dependency parser[8], with several additional rules to build a dependency tree where each node is a word and each edge is a syntactic dependency. Custom rules are used to compute another graph, a "semantic scene graph", comprised of three types of nodes: objects, attributes and relations. Attributes are linked to a single object, and relations connect objects between them. The reference graphs are merged into one to be compared with a candidate graph. Then, the scene graphs are converted into lists of word tuples. An object is a tuple with the object name, an attribute is a tuple of two words with the object and attribute names, and a relation is a tuple of three words containing the two objects connected and the relation names. Finally, the list is binarized for the candidate and the references, and used to compute an F-Score.

The M-SPICE metric [9] is a variant of SPICE, which takes multiple candidates for a single audio. This metric was introduced to evaluate the diversity of the words used in multiple candidates generated by stochastic decoding methods. The only difference is that the semantic graph of each candidate is merged into one, exactly as for the reference list. The other steps remain the same.

2.3. SPIDEr

SPIDEr [5] is a metric originally used as a cost function to optimize a model on SPICE and CIDEr-D at the same time. SPIDEr is the average of CIDEr-D and SPICE, and is supposed to have the benefits of both previous metrics. Since CIDEr-D gives a score between 0 and 10 and SPICE between 0 and 1, the SPIDEr score is between 0 and 5.5, which is quite uncommon for a metric. SPIDEr is usually the metric used in AAC papers to compare models, even if other machine-translation metrics like BLEU [10], ROUGE-L [11], and METEOR [12] scores are also reported.

3. SYSTEM DESCRIPTION

3.1. Datasets

The Clotho v2.1 dataset [13] is an audio captioning dataset containing 6974 audio files of approximately 43.6 hours from Freesound between 15 and 30 seconds. Each audio is described by 5 captions annotated by humans. The dataset is divided in 3 different splits: development, validation and evaluation, which corresponds to development-training, development-validation and development-testing, the conventional names used in the DCASE Challenge. In this paper, we use these names. The training subset contains 217362 words with a caption length between 8 and 20 words.

AudioCaps [14] is another audio captioning dataset containing 49838 training files of approximately 136.6 hours from AudioSet [15], a large audio tagging dataset with audio extracted from YouTube. AudioCaps contains only 1 caption per audio in the training subset and 5 captions for the validation and testing subsets. Since YouTube removes videos uploaded by users for various reasons, our version of AudioCaps contains only 46230 over 49838 files in training subset, 464 over 495 in validation subset and 912 over 975 files in testing subset. Our training subset contains 402482 words with a caption length between 1 and 52 words.

To extract audio features, we resample audio signals to 32 kHz and compute log-Mel spectrograms with a window size of 32 ms, a hop size of 10 ms and 64 Mel bands. All captions are put in lowercase and punctuation characters are removed. We used the spaCy tokenizer [16] to split sentences into words, resulting in a vocabulary of 4370 tokens for Clotho and 4724 words for AudioCaps.

3.2. Model architecture

We adopt a standard encoder-decoder structure used in most AAC systems, with a pre-trained encoder to extract audio features and a transformer decoder to generate our captions. The encoder is the CNN10 model, a convolutional network from the Pretrained Audio Neural Networks study (PANN) [2]. We used the weights available on Zenodo[3] to initialize the model at the beginning of the training. An affine layer was added to project 512-dimensional to 256-dimensional embeddings. We kept the time axis of the audio embedding used as input for the decoder.

The decoder is a standard transformer decoder [17]. It takes the audio embeddings as inputs and all the previous words predicted. The word embeddings are randomly initialized and learned during training. We use teacher forcing with cross-entropy to train the model. During the testing phase, captions are generated using beam search, and we select the best candidate using the probability of the sentence P given by the model. The combination of our encoder and decoder is simply named "CNN10-Transformer".

3.3. Experimental setup

We trained models for 50 epochs, on both datasets separately. To optimize our networks, we used Adam [18], with a learning rate set to 5.10^{-4} at the first epoch, a 10^{-6} weight decay, a 0.9 β_1 and 0.999 β_2, and ϵ set to 10^{-8}. We used a cosine learning rate scheduler with the following rule:

$$\text{lr}_k = \frac{1}{2}\bigg(1 + \cos\Big(\frac{k\pi}{K}\Big)\bigg)\text{lr}_0 \tag{2}$$

with k being the current epoch index, and K the total number of epochs.

The transformer decoder uses an embedding dimension d_{model} of 256, four attention heads h, six stacked standard decoder layers, and a global dropout P_{drop} set to 0.2. The last affine layer projects the 256-dimensional embeddings to an output of the vocabulary size of the dataset. We used label smoothing to reduce overfitting, set to 0.1 for AudioCaps and 0.2 for Clotho. In order to avoid gradient explosion, we clip gradients by a maximal L2-norm value set to 10 and 1 for AudioCaps and Clotho, respectively. During testing, beam size is set to 8 for Clotho and 2 for AudioCaps. The final encoder-decoder model results in 16M trainable

[3] `https://zenodo.org/record/3987831`

parameters. We also used SpecAugment [19] as audio data augmentation with two bands dropped on the time axis with a maximal size of 64 bins and one band dropped on the frequency axis with a maximum size of two bins. Our implementation uses PyTorch [20], PyTorch-Lightning [21] and our aac-datasets [4] package to download and manage audio captioning datasets.

4. SPIDEr RESULTS

Results on Clotho and AudioCaps of our model CNN10-Transformer are shown in Table 1. Standard deviations of our model are very small (0.001 and 0.004 for Clotho and AudioCaps, respectively). Cross-reference scores are computed by using one of the reference as a candidate and the four others as references five times.

Table 1: SPIDEr scores on Clotho v2.1 and AudioCaps with state-of-the-art results and cross-reference scores.

System	Clotho	AudioCaps
Best	0.320 [22]	0.465 [4]
Human	N/A	0.565 [14]
Cross-Referencing	0.573	0.564
CNN10-Transformer (ours)	0.247	0.401

Our model performs much better on AudioCaps than Clotho, with a SPIDEr score of 0.401 and 0.247, respectively. It is also closer to the cross-reference and human scores in AudioCaps. This is probably due to the fact that the CNN10 encoder has been pretrained on AudioSet, which is a superset of AudioCaps. In addition, the captions in AudioCaps are simpler than those in Clotho, with shorter sentences and a relatively smaller vocabulary. The current best score on AudioCaps is also much closer to the cross-reference top score (0.100 difference) than the one on Clotho (0.253 difference).

5. SPIDEr LIMITATIONS

5.1. The SPIDEr score varies greatly between beam search candidates

Tables 2 and 3 show examples of candidates and captions of reference, one from Clotho, one from AudioCaps. The probability P given by the model is also indicated. It used to select the best candidate among the beam search hypotheses. We also reported the SPIDEr score associated to each candidate.

In the Clotho example, the most likely caption candidate is also the one with the highest SPIDEr score, based on the fact that the rather rare word “tin” was found by the automatic system. Thus, in this example, the differences observed between the various hypotheses seem justified. On the contrary, in the second example, from AudioCaps, the most likely automatic caption is different from the one with the highest SPIDEr score.

The agreement accuracy between the best candidate according either to the likelihood and to the SPIDEr score is only of 26.5% on Clotho, and 22.6% on AudioCaps. The correlation coefficient on all the likelihoods and the SPIDEr scores is 0.224 on Clotho and 0.259 on AudioCaps. This shows that the maximum candidate likelihood P does not select the best caption according to the SPIDEr score.

Table 2: Captions for the Clotho development-testing file named “rain.wav”.

Candidates	P	SPIDEr
heavy rain is falling on a roof	0.361	0.562
heavy rain is falling on a **tin** roof	**0.408**	**0.930**
a heavy rain is falling on a roof	0.369	0.594
a heavy rain is falling on the ground	0.351	0.335
a heavy rain is falling on the roof	0.340	0.594
References		
heavy rain falls loudly onto a structure with a thin roof		
heavy rainfall falling onto a thin structure with a thin roof		
it is raining hard and the rain hits a tin roof		
rain that is pouring down very hard outside		
the hard rain is noisy as it hits a tin roof		

Table 3: Captions for an AudioCaps testing file (id: ‘jid4t-FzUn0’).

Candidates	P	SPIDEr
a woman speaks and a sheep bleats	0.475	0.190
a woman speaks and a **goat** bleats	0.464	**1.259**
a man speaks and a sheep bleats	0.464	0.344
an adult male speaks and a sheep bleats	0.450	0.231
an adult male is speaking and a sheep bleats	**0.491**	0.189
References		
a man speaking and laughing followed by a goat bleat		
a man is speaking in high tone while a goat is bleating one time		
a man speaks followed by a goat bleat		
a person speaks and a goat bleats		
a man is talking and snickering followed by a goat bleating		

5.2. Can we choose a better candidate automatically?

Selecting automatically the best candidate among the beam search hypotheses is a difficult problem: most candidates are very similar and usually describe the same events with different words. Figure 1 shows the histogram of the beam hypothesis indices that give the maximum SPIDEr score possible, for each candidate list when using a beam size of five. It reveals that no beam index seems better than another *a priori*. The same conclusion can be drawn with Clotho. We tried to automatically select the best candidate using several features: vocabulary size, sentence length, and even with a shallow neural network trained to rank the sentences, but all these approaches failed to significantly improve the global SPIDEr score.

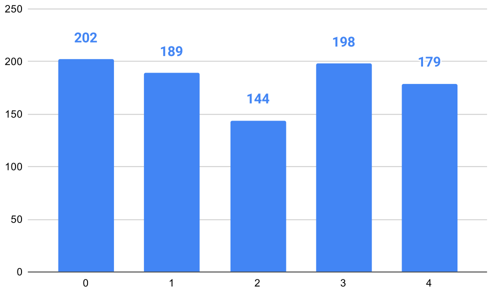

Figure 1: SPIDEr best beam indexes on AudioCaps testing subset

[4] https://pypi.org/project/aac-datasets

To overcome these limitations, we propose to consider all the candidates produced by the model and select the best SPIDEr score between them with a new metric.

6. SPIDEr-max

6.1. Definition

We propose SPIDEr-max, defined by the following equation:

$$\text{SPIDEr-max}(C, R) = \max_{i} \ \text{SPIDEr}(C_i, R) \tag{3}$$

where C is a list of N caption candidates and R a list of references.

It consists of retaining the largest SPIDEr score among the scores calculated for a set of caption candidates, to avoid having to choose a single hypothesis. The SPIDEr-max values are between 0 and 5.5, like the SPIDEr score. The source code in PyTorch will be made available on GitHub[5] upon paper acceptance.

6.2. Results

The score of SPIDEr-max highly depends on how many candidates we use, so we report results with various beam sizes in figures 2 and 3 for AudioCaps and Clotho, respectively. We varied the beam size from 1 to 10. If we imagine a human end-user, proposing at most 5 candidates captions would be reasonable, in our opinion.

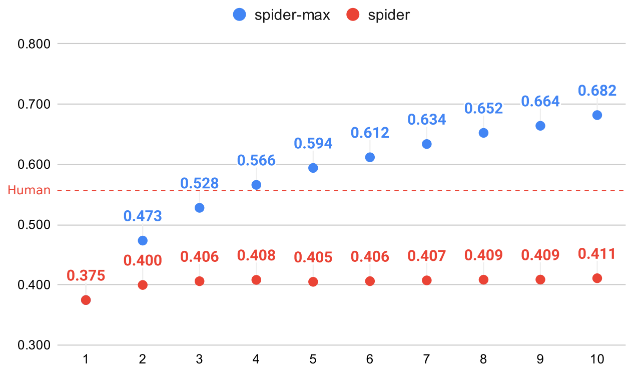

Figure 2: SPIDEr and SPIDEr-max scores with different beam sizes, calculated on the AudioCaps testing subset with CNN10-Transformer.

On AudioCaps, the SPIDEr-max score increases rapidly above the score of our model from 0.401 to 0.473 with only a beam size of two. The scores continue to rise above the human SPIDEr score (0.565), meaning that our model is already producing human-like captions, but fail to select them, if we take the maximum likelihood criterion.

On Clotho, the scores also increase with a higher beam size, but they do not reach the cross-reference score on the first beam sizes. This is probably due to the references of Clotho, which show more diversity in terms of vocabulary and n-grams than AudioCaps.

We also tried to compute the SPIDEr-max score for a beam size equal to 100, which gave 0.953 on AudioCaps and 0.535 on Clotho, but we decided to focus on a few candidates, as it would be more realistic in a real scenario, where, for instance, automatic captions are proposed to an end-user.

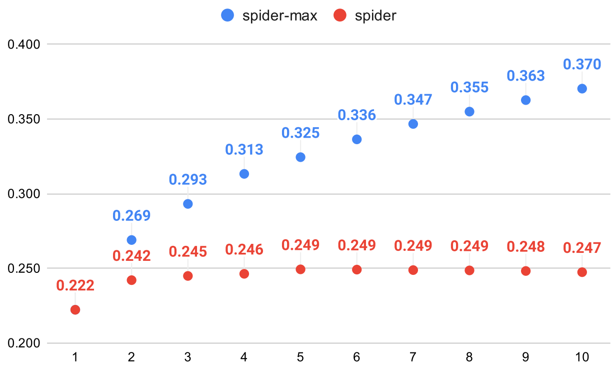

Figure 3: SPIDEr and SPIDEr-max scores with different beam sizes, calculated on the Clotho development-testing subset with CNN10-Transformer.

6.3. Why such a boost in SPIDEr-max?

As we saw in the previous section, SPIDEr-max increases rapidly and even outreaches the human SPIDEr score on AudioCaps. We also noticed that predicting a correct infrequent n-gram seems to drastically improves the score of a candidate, probably due to the CIDEr-D metric based on the TF-IDF of the n-grams. To see if there is a relation between TF-IDF and the SPIDEr and SPIDEr-max scores, we computed the difference between them with the best candidate given by the model and the best one given by the SPIDEr score for various beam sizes.

The correlation value between this variation of TF-IDF and SPIDEr scores is almost one for AudioCaps and Clotho. It suggests that the candidates selected by SPIDEr-max have a much higher TF-IDF than those selected by the model probabilities, which appears to significantly increase the CIDEr-D score and, thus, also the SPIDEr-max score.

7. CONCLUSION

In this paper, we showed that the SPIDEr score is very sensitive to the words used in the caption candidates, so we proposed a new metric, SPIDEr-max, that takes into account multiple candidates for each audio recording. The scores of SPIDEr-max compared to human scores of SPIDEr show that our model already produces human-like caption candidates, but selecting the caption with the highest SPIDEr score is not trivial. As future work, we are interested to study other metrics that do not use TF-IDF, such as model-based metrics like BERTScore [23] or FENSE [24]. We also look forward to testing SPIDEr-max with new models to see if our findings are repeated across architectures and training methods.

8. ACKNOWLEDGMENT

This work 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). Experiments presented in this paper were carried out using the OSIRIM platform that is administered by IRIT and supported by CNRS, the Region Midi-Pyrénées, the French Government, ERDF (see http://osirim.irit.fr/).

[5] `https://github.com/Labbeti/spider-max`

9. REFERENCES

[1] X. Xu, M. Wu, and K. Yu, "A comprehensive survey of automated audio captioning," *arXiv preprint arXiv:2205.05357*, 2022.

[2] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *arXiv:1912.10211 [cs, eess]*, Aug. 2020, arXiv: 1912.10211.

[3] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. L. Tang, X. Shao, M. D. Plumbley, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning for DCASE challenge 2021 task 6," DCASE2021 Challenge, Tech. Rep., July 2021.

[4] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *DCASE 2021 - 6th Workshop on Detection and Classification of Acoustic Scenes and Events*, Virtual, Spain, Nov. 2021.

[5] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved Image Captioning via Policy Gradient optimization of SPIDEr," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 873–881, Oct. 2017, arXiv: 1612.00370.

[6] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," *arXiv:1411.5726 [cs]*, June 2015, arXiv: 1411.5726.

[7] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," *arXiv:1607.08822 [cs]*, July 2016, arXiv: 1607.08822.

[8] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*, vol. 1. Sapporo, Japan: Association for Computational Linguistics, 2003, pp. 423–430.

[9] W.-N. Hsu, D. Harwath, C. Song, and J. Glass, "Text-Free Image-to-Speech Synthesis Using Learned Segmental Units," Dec. 2020, arXiv:2012.15454 [cs, eess].

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001, p. 311.

[11] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[12] M. Denkowski and A. Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014, pp. 376–380.

[13] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An Audio Captioning Dataset," *arXiv:1910.09387 [cs, eess]*, Oct. 2019, arXiv: 1910.09387.

[14] C. D. Kim, B. Kim, H. Lee, and G. 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: Association for Computational Linguistics, June 2019, pp. 119–132.

[15] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA: IEEE, Mar. 2017, pp. 776–780.

[16] I. Montani, M. Honnibal, M. Honnibal, S. V. Landeghem, A. Boyd, H. Peters, P. O. McCann, M. Samsonov, J. Geovedi, J. O'Regan, G. Orosz, D. Altinok, S. L. Kristiansen, Roman, E. Bot, L. Fiedler, G. Howard, W. Phatthiyaphaibun, Y. Tamura, S. Bozek, murat, M. Amery, B. Böing, P. K. Tippa, L. U. Vogelsang, B. Vanroy, R. Balakrishnan, V. Mazaev, and GregDubbin, "explosion/spaCy: v3.2.1: doc_cleaner component, new Matcher attributes, bug fixes and more," Dec. 2021.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017, arXiv: 1412.6980.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *proc. NeurIPS*, 2019, pp. 8026–8037.

[21] W. Falcon and .al, "Pytorch lightning," *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, vol. 3, 2019.

[22] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2022 challenge task 6: Audio captioning with audio-text retrieval pre-training," DCASE2022 Challenge, Tech. Rep., July 2022.

[23] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *arXiv:1904.09675 [cs]*, Feb. 2020, arXiv: 1904.09675.

[24] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, "Can Audio Captions Be Evaluated with Image Caption Metrics?" Jan. 2022, number: arXiv:2110.04684 arXiv:2110.04684 [cs, eess].