CP-JKU'S SUBMISSION TO TASK 6b OF THE DCASE2023 CHALLENGE: AUDIO RETRIEVAL WITH PaSST AND GPT-AUGMENTED CAPTIONS

Technical Report

Paul Primus[1], *Khaled Koutini*[1,2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU)
[2]LIT Artificial Intelligence Lab
Johannes Kepler University, Austria

ABSTRACT

This technical report describes CP-JKU's submission to the natural-language-based audio retrieval task of the 2023 DCASE Challenge (Task 6b). Our proposed system uses pretrained audio and text embedding models to project recordings and textual descriptions into a shared audio-caption space in which related examples from different modalities are close. We pre-train our models on WaveCaps, AudioCap, and ClothoV2, three large datasets with audio-caption pairs. We further augment the captions in the ClothoV2 dataset using the provided metadata and the ChatGPT API in order to reduce overfitting. Our best single system submission outperforms the current state-of-the-art text-to-audio retrieval system on the ClothoV2 test split by 4.6 pp. R@1. Furthermore, our ensemble beats the previous year's best submission on the test split by 11.5 pp. mAP@10. Our implementation is available in GitHub[1].

Index Terms— Natural-Language-Based Audio Retrieval, PaSST, ChatGPT

1. INTRODUCTION

The natural-language-based audio retrieval task of the 2023 DCASE Challenge [1] revolves around ranking audio recordings based on their similarity to textual descriptions. Current state-of-the-art systems solve this task by converting both recordings and textual descriptions into high-level representations and aligning them within a shared audio-caption space. The ranking process is then carried out by measuring the distance between these embeddings (Fig. 1). This setup has been the standard among last year's submissions to the DCASE challenge [2–5]. CNN architectures pre-trained on Audioset [6] were the most common audio encoders. At the same time, large transformer models, such as BERT [7] and RoBERTa [8], were the most popular text encoders. Recently, Mei et al. [9] managed to set a new state-of-the-art performance on ClothoV2 [10] by introducing WaveCaps [9], a new large dataset with synthetic captions.

Our submission retrieves audio recordings based on text queries in a similar manner as most of last year's submissions, but it differs from those systems in three major aspects:

The Audio Encoder: We used an audio spectrogram transformer called PaSST [11] instead of CNN14 [6] for audio embedding because PaSST outperforms CNN14 on Audioset [12] and other relevant audio benchmarks [13]. PaSST employs Patchout during training, which increases training speed and memory efficiency while at the same time acting as a regularizer. Using PaSST in this task was previously investigated by Pellegrini [5].

Pre-training We trained our models on AudioCaps [14] and WaveCaps [9], two large audio captioning datasets with human-generated and synthetic captions, respectively. Using these large datasets to (pre-)train the models resulted in significantly better representations in the embedding space and consequently in better retrieval performance, as we will show in Section 3.

GPT Caption Augmentation We took advantage of the available metadata associated with each audio recording and augmented the captions during training by generating additional captions using the available keywords and the GPT3.5-turbo API, also known as ChatGPT.

Our best single model submission, consisting of PaSST and

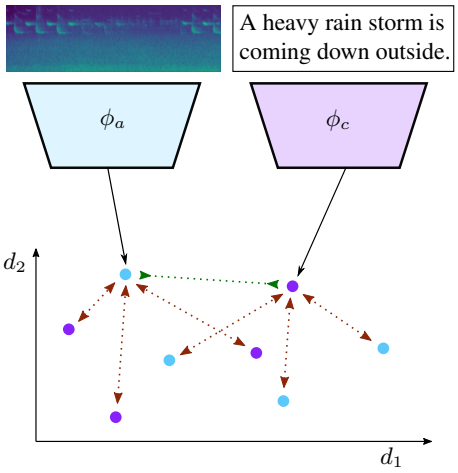

Figure 1: The proposed audio-retrieval system in a nutshell: audio and descriptions are transformed into the shared audio-caption embedding space via the audio and description embedding models ϕ_a and ϕ_c, respectively. The contrastive loss maximizes the similarities between matching pairs. Colours of the dots identify audio and text embeddings, respectively. Red and green arrows represent non-matching and matching pairs

[1]https://github.com/OptimusPrimus/dcase2023_task6b

RoBERTa, is pre-trained on WaveCaps and AudioCaps and utilizes the GPT-augmented version of Clotho2. This model outperforms the current state of the art [9] by 4.5 points R@1. Our ensemble, consisting of eight models, beats the baseline [1] and the previous year's best [3] submission on the publicly available test set by 19.2 and 11.5 pp. mAP@10, respectively.

2. PROPOSED METHOD

Our model uses separate audio and caption embedding networks, denoted as $\phi_a(\cdot)$ and $\phi_c(\cdot)$, respectively, to embed pairs of spectrograms and descriptions $\{(a_i, c_i)\}_{i=1}^{N}$ into a shared D-dimensional space such that representations of matching audio-caption pairs are close. This behavior is achieved by contrastive training, which makes the embeddings of matching audio-caption pairs (a_i, c_i) more similar while pushing the representations of mismatching pairs $(a_i, c_{j;j\neq i})$ apart. The agreement between audio a_i and description c_j is estimated via the normalized dot product in the shared embedding space:

$$C_{ij} = \frac{\phi_{\mathrm{a}}(a_i)^T \cdot \phi_{\mathrm{c}}(c_j)}{\|\phi_{\mathrm{a}}(a_i)\|^2 \|\phi_{\mathrm{t}}(c_j)\|^2}$$

The similarity matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$ holds the agreement of matching pairs on the diagonal and the agreement of mismatching pairs off-diagonal. We train the system using the NT-Xent [15] loss, which is defined as the average Cross Entropy (CE) between the ground truth and the posterior probability distribution of recordings given a text query and Cross Entropy between the ground truth and the posterior probability distribution of text queries given an audio recording; the ground truth is given by the identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$:

$$\mathcal{L} = \frac{1}{2 \cdot N} \sum_{i=1}^{N} \mathrm{CE}(\mathbf{C}_{i*}, \mathbf{I}_{i*}) + \mathrm{CE}(\mathbf{C}_{*i}, \mathbf{I}_{*i})$$

2.1. Audio Embedding Models

We choose the Patchout faSt Spectrogram Transformer (PaSST) [11] to convert audio recordings into a compact, high-level vector representation because it achieves state-of-the-art results on multiple audio classification benchmarks [16] while keeping the memory and computational complexity low compared to the vanilla audio spectrogram transformers [17]. PaSST uses pre-trained parameters taken from a vision transformer [18, 19] trained on ImageNet [20], and fine-tunes them on AudioSet [12] for general-purpose audio tagging. The relatively low computational and memory footprint is achieved by dropping patches from the input sequence. This procedure, called Patchout [11], additionally regularizes the model during training. We conducted experiments with PaSST models that take audios of 10 or 15 seconds in length as input and extract overlapping or non-overlapping patches from the spectrogram:

PaSST-S was pre-trained on AudioSet and achieves a mAP of 48.6 on the AudioSet test set. The model holds a positional encoding for inputs of up to 10 seconds. The extracted patches are 16×16 pixels, and the stride for extraction is 10 pixels, resulting in an overlap of 6 pixels. This results in $12 \times 99 = 1188$ patches per 10-second segment. We apply *Structured* Patchout of 4 and 50 over frequency and time dimensions, respectively, reducing the input sequence length to $8 \times 49 = 392$.

PaSST-N extracts non-overlapping patches from the input spectrogram, which greatly reduces the input sequence length and, therefore, the computational requirements. The model's performance on AudioSet is a mAP of 46.8. This model also uses a fixed positional encoding of 10 seconds in length. The extracted patches are 16×16 pixels, and a stride of 16 pixels is used for patch extraction. This results in $8 \times 62 = 496$ patches per 10 seconds of input. We apply *Structured* Patchout of 2 and 15 over frequency and time dimensions, respectively, reducing the input sequence length to $6 \times 47 = 282$.

PaSST-S15 achieves a mAP of 47.4 on AudioSet. This model provides a positional encoding of up to 20 seconds in length; however, we only feed 15-second long audio snippets into this model. We used this model to investigate whether providing a larger audio context to PaSST can lead to better retrieval performance. Similar to PaSST-S, the extracted patches are 16×16 pixels with a stride of 10 pixels. This results in $12 \times 149 = 1788$ patches per 15-seconds. We apply *Structured* Patchout of 4 and 80 over frequency and time dimensions, respectively, reducing the input sequence length to $8 \times 69 = 552$.

All the PaSST models consist of approximately 86.2 million trainable weights. Pre-trained PaSST models are available on GitHub.[2]

2.2. Sentence Embedding Models

We conducted experiments with four different sentence embedding models: bert-base, bert-large, roberta-base, and roberta-large. All models are bi-directional self-attention-based sentence encoders that underwent self-supervised pretraining on the BookCorpus [21], and WikiText datasets [22]. BERT and RoBERTa differ in the masking strategy used during training: the former was trained using 10 fixed masks for each sentence, while the latter used new, dynamically generated masks in each forward pass. For both models, we selected the output vector that corresponds to the class token as sentence embedding. The parameter counts for bert-base, bert-large, roberta-base, and roberta-large are around 110, 345, 123, and 354 million, respectively. Pretrained models were taken from HuggingFace[3].

2.3. Shared Audio-Caption Space

The audio and text embeddings generated by the encoders are integrated into a shared audio-caption space by using a simple linear projection of size 1024. Initial experiments suggested that using a non-linear projection (e.g., a multilayer neural network) does not significantly improve performance.

2.4. Datasets

Our models were trained in two steps on multiple datasets. First, we performed pretraining on ClothoV2, AudioCaps, and WaveCaps. The resulting models were then further finetuned on a custom, augmented version of the ClothoV2 (call ClothoV2_GPT below), which also takes into account the available meta-data associated with each audio file.

2.4.1. ClothoV2

ClothoV2 [10] contains 10-30 second-long audio recordings and captions that are between 8 and 20 words long. The development

[2] https://github.com/kkoutini/passt_hear21
[3] https://huggingface.co/

Instruction	GPT3.5-turbo's Response
I will give a description of an audio recording and tags associated with the audio recording. Generate 5 audio captions describing the sound event. Each audio caption should be one sentence with less than 20 words. Use grammatical subject-verb-object sentences. Do not include names of cities, countries, and persons. Do not include the time. Do not include recording device names. Do not write introductions or explanations. Do not use "heard" or "recorded". Start each output sentence with its index. 'A person is making noise by tapping their fingernails on a solid surface.' [dull, metal]	1. The sound of nails tapping on a metal surface is heard. 2. Fingernails are producing a dull sound when tapped. 3. A person is tapping their nails on a solid, metal surface. 4. The noise of someone tapping their fingernails is audible. 5. Nails are rhythmically tapping on a hard metal object.

Table 1: Example query fed to GPT3.5-turbo to augment a ClothoV2 caption (in blue) with the available keywords (in red; query inspired by WaveCaps [9]). The response is a list of rephrased captions, some of which take the keywords into account (highlighted in red).

set's training, validation, and test split suggested by the organizers contains 3840, 1045, and 1045 recordings, respectively, and each recording is associated with five human-generated captions. The leaderboard evaluation split used for the final system ranking contains 1000 audio recordings and 1000 captions. We used the validation split to monitor the generalization performance and report the performance on the test spit in this report.

2.4.2. *AudioCaps*

AudioCaps [14] contains 51.308 audio recordings taken from AudioSet and one human-written caption for each of them. Each audio recording has a duration of 10 seconds, and the captions are, on average, 9.8 words long. We concatenated the training, validation, and testing split of AudioCaps into one large dataset and utilized it for pretraining.

2.4.3. *WaveCaps*

WaveCaps [9] is a weakly-labeled audio-caption dataset that contains 403.050 audio recordings of varying length collected from FreeSound, BBC Sound Effects, SoundBible, and the strongly supervised AudioSet subset. Each audio file is associated with a synthetic audio caption that was created by instructing the GPT3.5-turbo model to extract relevant sound events from metadata and output a single-sentence description. The generated captions are, on average, 7.8 words long. The authors demonstrated the usefulness of these synthetic captions by successfully utilizing this dataset for audio retrieval, audio captioning, and text-based sound generation. We used the complete dataset available on HuggingFace[4] for pretraining.

2.4.4. *ClothoV2_GPT*

Each audio recording in the ClothoV2 dataset is associated with additional metadata consisting of the file name, a list of keywords, a username, and a web URL. We took advantage of the additional information and used GPT3.5-turbo to augment the human-generated captions with the available keywords. To this end, we instructed the model to take the keywords into account and rephrase the available captions. We generated five new captions for each caption in the training set, resulting in $96,000$ additional captions. Table 1 gives an example query and the resulting augmented captions. We will refer to this augmented version of ClothoV2 as ClothoV2_GPT.

[4] https://huggingface.co/datasets/cvssp/WavCaps

2.5. Preprocessing

To allow batched processing of recordings of varying lengths, we extracted random 30-second snippets from those audio recordings that are longer than 30 seconds and zero-padded shorter recordings to the maximum duration in the current batch. The resulting waveforms were converted to 128-bin log-MEL spectrograms using a 1024-point FFT (32ms) and hop size of 320 (10ms). The spectrograms were centered and whitened with the approximate global mean and standard deviation before feeding them into the audio embedding model. PaSST uses a positional embedding with a fixed size. We, therefore, cut the input spectrograms into 10 or 15-second long snippets, embedded them separately, and aggreged the individual embeddings by averaging them. The input sentences were pre-processed by transforming all characters to lowercase and removing punctuation. The resulting strings were tokenized with the WordPiece tokenizer [23], padded to the maximum sequence length in the current batch, and truncated if they were longer than 32 tokens.

2.6. Training

We pre-trained all models on AudioCaps, WaveCaps, and the training set of ClothoV2. Both embedding models were jointly optimized using gradient descent with a batch size of 64. We used the Adam update rule [24] for 15 epochs, with one warmup epoch. Thereafter, the learning rate was decayed from 2×10^{-5} to 10^{-7} using a cosine schedule. The hyperparameters of the optimizer were set to PyTorch's [25] defaults. We further used structured patchout as a regularizer during training with hyperparameters depending on the audio length and patch extraction (see Section 2.1). Finetuning on ClothoV2_GPT was done in a similar manner as pretraining but only for five epochs with a maximum learning rate of 8×10^{-6}. During the finetuning procedure, a caption was swapped with one of its five GPT-augmented versions with a probability of 0.3. Our main evaluation criterion for model selection was the mean Average Precision among the top-10 results (mAP@10) on the validation set, which takes the rank of the correct recording into account. In the results section, we additionally report the recall among the top-1, top-5, and top-10 retrieved results.

audio embedding	segment length (s)	overlap	text embedding	finetune	GPT-augment	mAP@10	R@1	R@5	R@10	SID \ reference
PaSST-N[1]	10	✗	bert-base	✓	✓	37.15	24.63	53.53	68.38	
PaSST-N[2]	10	✗	roberta-base	✓	✓	37.29	24.56	54.78	68.25	
PaSST-N[3]	10	✗	bert-large	✓	✓	37.85	25.11	55.69	68.42	
PaSST-N[4]	10	✗	roberta-large	✓	✓	38.56	26.07	55.27	69.30	3
PaSST-S	10	✓	bert-base	✓	✓	36.29	24.15	53.22	67.06	
PaSST-S	10	✓	roberta-base	✓	✓	36.76	24.15	54.16	68.33	
PaSST-S[5]	10	✓	bert-large	✓	✓	37.06	24.80	53.89	67.69	
PaSST-S[6]	10	✓	roberta-large	✓	✓	37.79	25.53	54.62	68.48	
PaSST-S15[7]	15	✗	bert-large	✓	✓	38.16	25.36	54.85	68.42	
PaSST-S15[8]	15	✗	roberta-large	✓	✓	38.42	26.00	55.54	69.13	
PaSST-N	10	✗	roberta-large	✓	✗	38.00	25.51	55.06	68.56	2
PaSST-N	10	✗	roberta-large	✗	✗	36.27	24.38	52.54	66.22	4
ensemble of [1,2,3,4,5,6,7,8]						41.42	28.89	58.73	71.08	1
CNN14	30	-	all-mpnet-base-v2	✗	✗	22.20	13.00	34.30	48.00	[1]
CNN14	30	-	bert-base	✓	✗	-	21.40	47.90	61.90	[9]
best submission 2022 (ensemble)						29.90	18.80	44.70	58.70	[3]

Table 2: Text-to-audio retrieval performance on the ClothoV2 test set for different audio and language model combinations. Submitted systems are indicated in the SID column.

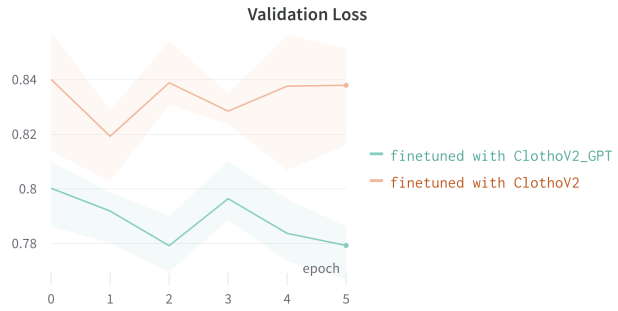

Figure 2: Validation loss during fine-tuning. Results were averaged over three runs, and the error bars indicate the min-max range.

3. RESULTS

We summarize the results of our experiments in Table 2 and highlight some trends in this section. Firstly, we can observe that the models with the larger sentence embedding models tend to perform better than the ones with smaller ones (e.g., compare mAP@10 of row one with row three in Table 2). Secondly, we observe that PaSST-N (without overlap) tends to give better retrieval performance compared to PaSST-S (with overlap) (compare mAP@10 in the first and second sections of Table 2). A small ablation study suggests that both fine-tuning on ClothoV2 and fine-tuning on ClothoV2_GPT are effective (see section four in Table 2 and compare this to row four). We further observe slightly better test results and less overfitting on the validation set when finetuning with ClothoV2_GPT (compare row four and the first row of the fourth section in Table 2; see also Fig. 2). An ensemble of eight models yields the best results on the test set; adding additional models or removing models from the ensemble reduces the performance on the test set. Our ensemble improves the mAP@10 compared to the baseline and the previous year's best submission by 19.2 and 11.5 pp., respectively (compare the last row and section five in Table 2). Our best single model further improves the R@1 by 4.5 pp. (compare row four and the second row in the last section in Table 2).

4. SUBMISSION

Since the training procedure was fairly stable and in order to remain competitive, we retrained all previously discussed models and utilized the whole ClothoV2 development set (i.e., train, validation, and test splits) instead of the ClothoV2 training split only. We submitted retrieval results on the leaderboard evaluation set for four models to the challenge; the item numbers in the following list correspond to the numbers in the SID column in Table 2:

1. an ensemble of eight models (ensembled models are indicated in Table 2)
2. PaSST-N without overlap and a segment length of 10 seconds, roberta-large, finetuned on ClothoV2
3. PaSST-N without overlap and a segment length of 10 seconds, roberta-large, finetuned on ClothoV2_GPT
4. PaSST-N without overlap and a segment length of 10 seconds, roberta-large, not finetuned

5. ACKNOWLEDGMENT

The LIT AI Lab is financed by the Federal State of Upper Austria. The computational results presented in this work have been partially achieved using the Vienna Scientific Cluster (VSC).

6. REFERENCES

[1] "Language-Based Audio Retrieval, Task description," https://dcase.community/challenge2023/task-language-based-audio-retrieval#description, accessed: 2023-05-21.

[2] H. Xie, S. Lipping, and T. Virtanen, "Language-based audio retrieval task in dcase 2022 challenge," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[3] 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., 2022.

[4] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Language-based audio retrieval with pre-trained models," DCASE2022 Challenge, Tech. Rep., July 2022.

[5] T. Pellegrini, "IRIT-UPS DCASE 2022 language-based audio retrieval system," DCASE2022 Challenge, Tech. Rep., July 2022.

[6] 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," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2020.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. of the North American Ch. of the Ass. for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[9] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *CoRR*, vol. abs/2303.17395, 2023.

[10] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an Audio Captioning Dataset," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2020.

[11] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *23rd Annual Conf. of the Int. Speech Communication Association, Interspeech*, 2022.

[12] 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 *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2017.

[13] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. H. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, "HEAR: holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track, NeurIPS*, 2021.

[14] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. of the North American Ch. of the Ass. for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. of the 37nd Int. Conf. on Machine Learning, ICML*, 2020.

[16] K. Koutini, S. Masoudian, F. Schmid, H. Eghbal-zadeh, J. Schlüter, and G. Widmer, "Learning General Audio Representations With Large-Scale Training of Patchout Audio Transformers," *HEAR: Holistic Evaluation of Audio Representations*, vol. 166, 2022.

[17] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *22nd Annual Conf. of the Int. Speech Communication Association, Interspeech*, 2021.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020.

[19] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. of the 38th Int. Conf. on Machine Learning, ICML*, 2021.

[20] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition CVPR*, 2009.

[21] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *IEEE Int. Conf. on Computer Vision, ICCV*, 2015.

[22] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *5th Int. Conf. on Learning Representations, ICLR*, 2017.

[23] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations, ICLR*, 2015.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. 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 *Annual Conf. on Neural Information Processing Systems, NEURIPS*, 2019.