CL4AC: A CONTRASTIVE LOSS FOR AUDIO CAPTIONING

Xubo Liu[1*], *Qiushi Huang*[2,3*], *Xinhao Mei*[1], *Tom Ko*[3], *H Lilian Tang*[2], *Mark D. Plumbley*[1], *Wenwu Wang*[1],

[1] Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, UK,
{xubo.liu, x.mei, m.plumbley, w.wang}@surrey.ac.uk
[2] Department of Computer Science, University of Surrey, UK, {qiushi.huang, h.tang}@surrey.ac.uk
[3] Southern University of Science and Technology, Shenzhen, China, {tomkocse}@gmail.com

ABSTRACT

Automated Audio captioning (AAC) is a cross-modal translation task that aims to use natural language to describe the content of an audio clip. As shown in the submissions received for Task 6 of the DCASE 2021 Challenges, this problem has received increasing interest in the community. The existing AAC systems are usually based on an encoder-decoder architecture, where the audio signal is encoded into a latent representation, and aligned with its corresponding text descriptions, then a decoder is used to generate the captions. However, training of an AAC system often encounters the problem of data scarcity, which may lead to inaccurate representation and audio-text alignment. To address this problem, we propose a novel encoder-decoder framework called **C**ontrastive **L**oss for **A**udio **C**aptioning (CL4AC). In CL4AC, the self-supervision signals derived from the original audio-text paired data are used to exploit the correspondences between audio and texts by contrasting samples, which can improve the quality of latent representation and the alignment between audio and texts, while trained with limited data. Experiments are performed on the Clotho dataset to show the effectiveness of our proposed approach.

Index Terms— Audio captioning, cross-modal translation, contrastive loss, deep learning

1. INTRODUCTION

Automated Audio captioning (AAC) is a cross-modal translation task of generating a natural language description for an audio clip. It has various potential applications. For example, AAC can be used for generating subtitles for the audio content in a television program, or for generating text descriptions of audio to help the hearing impaired in accessing audio content. It can also be used by sound search engines to achieve more accurate retrieval and recommendation, or by a surveillance system to facilitate the detection of acoustic anomalies. The AAC problem has attracted increasing interest from the acoustic signal processing and machine learning communities in recent years.

Existing AAC systems are usually based on an encoder-decoder architecture [1, 2, 3, 4]. The audio data is encoded into a latent representation and aligned with its corresponding text description. Then a decoder is used to generate the captions. Training of an AAC system often encounters the problem of data scarcity, which may lead to inaccurate representation and audio-text alignment. For example, Clotho [5] is a popular AAC dataset and was used for the DCASE challenge. However, it only contains 6974 audio samples, and each audio sample has five captions. To address this problem, information from keywords has been exploited for AAC [4, 6, 7]. The keywords of the caption are tagged firstly and then used to assist the generation of captions. However, due to the diversity of keywords, the tagging results of unseen audio samples may not be accurate in the inference stage. On the other hand, transfer learning techniques [8, 9] have been widely used in task 6 of the DCASE 2021 challenge, offering substantially improved performance. However, transfer learning relies heavily on large-scale external data [10] and pre-trained models [11].

Contrastive learning [12, 13] is a self-supervised paradigm that helps the model obtain high-quality representation. Inspired by the recent success of contrastive learning in computer vision (CV) [14] and natural language processing (NLP) [15, 16], we propose a novel encoder-decoder framework called **C**ontrastive **L**oss for **A**udio **C**aptioning (CL4AC). In CL4AC, the self-supervision signals derived from the original audio-text paired data are used to exploit the correspondences between audio and texts by contrasting samples. More precisely, we construct mismatched audio-text pairs as negative samples. Then, a contrastive learning objective is designed to maximize the difference between the representation of the matched audio-caption pair derived from the negative pairs. In this way, the quality of latent representation and the alignment between audio and texts can be improved, when they are trained with limited amount of data. To the best of our knowledge, contrastive learning has not been used for AAC in the literature.

The remainder of this paper are organised as follows. We introduce our proposed CL4AC in Section 2. Experiments are described in Section 3. Results are shown in Section 4. Finally, we conclude our work and discuss the future work in Section 5. The code of this work is made available on GitHub[1].

2. CONTRASTIVE LOSS FOR AUDIO CAPTIONING

In this section, we present our proposed contrastive learning framework for audio captioning (CL4AC). We first introduce the encoder-decoder architecture of CL4AC in Section 2.1. Then, we present the contrastive learning framework in Section 2.2.

2.1. Encoder-Decoder architecture

We first define the notations used in this section. The training data for AAC consists of paired audio and texts data. We denote a training set of N audio-text pairs by $D = \{(a_n, C_n)\}_{n=1}^{N}$, where $a \in \mathbb{R}^{H \times W}$ is the log mel-spectrogram of an audio clip with H and W being

[*] The first two authors contributed equally to this work.

[1] `https://github.com/liuxubo717/contrastive_loss_for_audio_captioning`

its height and width, respectively, $C = \{w_m\}_{m=1}^{M}$ is the token sequence of a caption where w_m is the m-th token in the caption C having M tokens, a_n is the log mel-spectrogram of the n-th audio clip in the dataset, and C_n is the token sequence of the n-th caption in the dataset.

The sequence-to-sequence architecture with Convolutional Neural Network (CNN) encoder and Transformer decoder are used as the basis of our proposed framework, as shown in Figure 1. This architecture was shown to offer the state-of-the-art performance [8, 9] in Task 6 of the DCASE 2021 challenge.

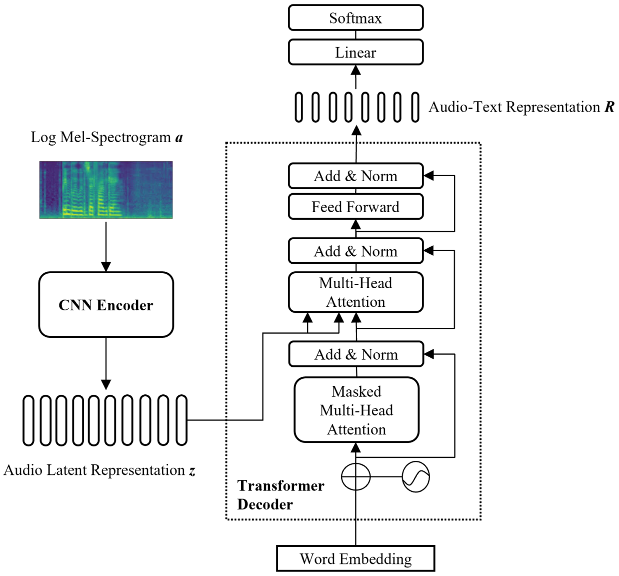

Figure 1: Sequence-to-sequence architecture with CNN encoder and Transformer decoder for audio captioning. The components in the dashed box indicate the Transformer decoder.

2.1.1. CNN encoder

Pre-trained audio neural networks (PANNs) [11] have demonstrated a powerful ability in extracting latent representation of audio signals for different downstream audio recognition tasks. To benefit from its high-quality audio representation, we choose PANNs as the encoder, which will be described in Section 3.3 in details. The PANNs encoder takes the log mel-spectrogram a of an audio clip as the input and extracts its latent representation $z \in \mathbb{R}^{H' \times W'}$. Formally:

$$z = \text{Encoder}(a). \tag{1}$$

2.1.2. Transformer decoder

The Transformer model has shown the state-of-the-art performance on language-related cross-modal task [17, 18], and is used as the decoder in our work. There are two main components in the decoder. Firstly, each token w_m in the input token sequence C is converted into a word embedding $e_m \in \mathbb{R}^{1 \times E}$, where E is the dimension of the word embedding, by the word2vec algorithm using Continuous Bag of Words Model (CBOW) [19] and Skip-Gram [20] model trained purely on the caption corpus. Then the word embedding of tokens are fed into the first self-attention layer to obtain their hidden states. The latent representation z of an audio clip extracted by the encoder is aligned and calculated with the hidden states of tokens, then the audio-text representation is obtained by the transformer decoder, denoted as $R \in \mathbb{R}^{M \times T}$, which consists of M vectors $\{r_m\}_{m=1}^{M}$, where the number of vectors is equal to the length of the input token sequence C and the dimension of each vector is T. The vector r_m of the audio-text representation R is calculated based on the word embeddings $\{e_1, ..., e_{m-1}\}$ and the audio latent representation. Hence, each r_m corresponds to the token w_m in the input token sequence C one-to-one, which can be used to predict the probability of the word over the vocabulary after it is passed through the final linear layer with softmax function. The transformer decoder predicts the m-th word w_m based on the previous tokens $\{w_1, ..., w_{m-1}\}$ and the audio latent representation z, as follows,

$$p(w_m|z, w_1, ..., w_{m-1}) = \text{Decoder}(z, w_1, ..., w_{m-1}). \tag{2}$$

The training objective is to optimize the cross entropy (CE) loss defined in terms of the predicted words as:

$$\text{Loss}_{\text{CE}} = -\mathbb{E}_{(a,C)\sim D} \log p(w_m|z, w_1, ..., w_{m-1}). \tag{3}$$

2.2. Contrastive learning framework

To obtain accurate audio-text representation R while the model is trained with limited data, we use the self-supervised signal derived from the audio-text training data by contrasting samples. First, we construct mismatched audio-text pairs as negative samples. Then, a contrasting auxiliary task is designed to maximize the difference between the representation R of the matched audio-text pair derived from negative pairs. The representations of the audio-text paired data are pulled together in the latent space while simultaneously pushing apart clusters of unpaired negative data by contrastive learning, as shown in Figure 2. In this way, the quality of audio-text representation and the alignment between audio and texts can be improved.

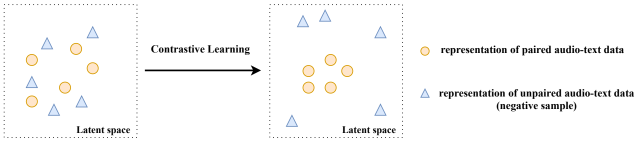
Contrastive Learning
Latent space
Latent space
representation of paired audio-text data
representation of unpaired audio-text data (negative sample)

Figure 2: The representations of the audio-text paired data are pulled together in the latent space while simultaneously pushing apart clusters of unpaired negative data by Contrastive Learning (CL).

More specifically, for each anchor audio-text paired training data $x = (a, C)$, we replace the caption C by $C_{negative}$ which is a randomly selected caption unpaired with a in the training set D. Then, the mismatched audio-text pair as the negative training sample is constructed, denoted as $x_{negative} = (a, C_{negative})$. Table 1 shows the examples of x and $x_{negative}$ in the Clotho dataset. Since the last vector in the audio-text representation R is able to attend the context of all input tokens and the audio feature, the value of last vector of R is fed into a binary classifier $f(.)$ to predict whether the input audio and text data are paired $(y = 0)$ or not $(y = 1)$. The contrastive learning (CL) loss for this auxiliary task is defined as follows:

$$\text{Loss}_{\text{CL}} = -\mathbb{E}_{x'\sim D'} \log p(y|f(x')), \tag{4}$$

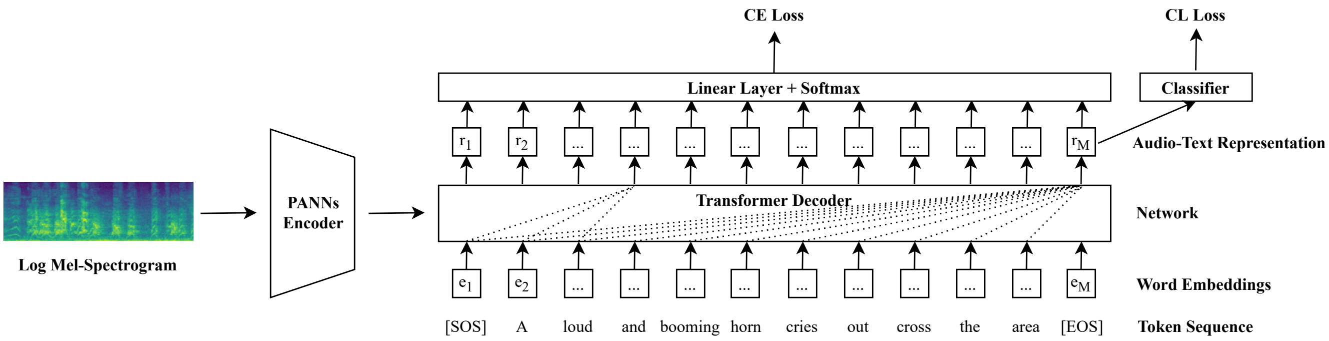
CE Loss
CL Loss
Linear Layer + Softmax
Classifier
r_1 r_2 ... ... r_M
Audio-Text Representation
PANNs Encoder
Transformer Decoder
Network
Log Mel-Spectrogram
e_1 e_2 ... ... e_M
Word Embeddings
[SOS] A loud and booming horn cries out cross the area [EOS]
Token Sequence

Figure 3: Contrastive loss for audio captioning (CL4AC) framework. The dashed lines indicate that the vector r_m of the audio-text representation R is calculated based on the word embeddings $\{e_1, ..., e_{m-1}\}$ and the audio latent representation obtained from PANNs. The last audio-text representation vector r_M is fed to the classifier $f(.)$ whose output is used to calculate the Contrastive Learning (CL) loss.

Example	paired caption C	unpaired caption $C_{negative}$
audio a	Something goes round that is playing its song At the fair, music is playing near a carousel through the speaker Chiming of bells, whistles and horns at a performance Fair kind music is being played at the circus grounds Polka or fair kind of music is being played	The Air is blowing some what fast outside A hand held sander was used as various speeds A hard gravel ground is walked on by someone A person using a hard object to tap and scrape glasses The wind is blowing and the waves are flowing

Table 1: Examples of paired audio-text training data $x = (a, C)$ and negative training sample $x_{negative} = (a, C_{negative})$. Examples are selected from the Clotho dataset, where each audio data has five corresponding captions.

.

where D' is the extended training set by merging the negative samples into the original training set D and x' is the audio-text pair drawn from D'. The full training objective of CL4AC is:

$$\mathrm{Loss}_{\mathrm{Training}} = (1 - y)\,\mathrm{Loss}_{\mathrm{CE}} + \mathrm{Loss}_{\mathrm{CL}}\,. \qquad (5)$$

When the input is a negative audio-text pair, the gradient provided by the CE loss is meaningless, for this case, only CL loss is used for updating the model. The framework of CL4AC is shown in Figure 3.

3. EXPERIMENTS

3.1. Dataset

Clotho [5] is an AAC dataset whose sound clips are from the Freesound platform and annotated by Amazon Mechanical Turk. Clotho v2 was released for Task 6 of the DCASE 2021 Challenge, which contains 3839, 1045 and 1045 audio clips for the development, validation and evaluation split respectively. The sampling rate of all audio clips in Clotho dataset is 44 100 kHz. Each audio clip has five captions. Audio clips are of 15 to 30s duration and captions are eight to 20 words long. We merge the development and validation split, forming a new training set with 4884 audio clips. The performance of AAC system is evaluated on the evaluation split.

3.2. Data pre-processing

We use the original sampling rate to load audio data, and an 64-dimensional log mel-spectrogram is calculated using the short-time Fourier transform (STFT) with a frame size of 1024 samples, a hop size of 512 samples, and a Hanning window. SpecAugment [21] is used for data augmentation.

We transform all captions in the Clotho dataset to lower case with punctuation removed. Two special tokens “<sos>” and “<eos>” are added on the start and end of each caption. The vocabulary of the Clotho dataset contains 4367 words.

3.3. Model implementation

CNN-10 of PANNs [11] is used as the encoder to prevent over-fitting while trained with limited data. Specifically, the CNN-10 consists of four convolutional blocks where each has two convolutional layers with a kernel size of 3 × 3. Batch normalization and ReLU are used after each convolutional layer. The channels number of each block are 64, 128, 256 and 512, respectively. An average pooling layer with kernel size 2 × 2 is applied between them for down-sampling. Global average pooling is applied along the frequency axis after the last convolutional block followed by two fully connected layers to align the dimension of the output with the decoder input. Two transformer blocks with four heads and 128 hidden units are used as the decoder. The implementation for the encoder and decoder is the same as that in our DCASE 2021 Challenge system[2], which is the highest-scoring system without using model ensembles.

We trained the proposed model using Adam [22] optimizer with a batch size of 16. Warm-up is used in the first 5 epochs to increase the learning rate to the initial learning rate linearly. The learning rate is then decreased to 1/10 of itself every 10 epochs. Dropout with a rate of 0.2 is applied in the proposed model to mitigate the over-fitting problem. We train the model for 30 epochs with an initial learning rate of 5×10^{-4} on the training set of the Clotho dataset.

[2] `https://github.com/XinhaoMei/DCASE2021_task6_v2`

Model	BLEU_1	BLEU_2	BLEU_3	BLEU_4	ROUGE_L	METERO	CIDEr	SPICE	SPIDEr
Baseline	0.550	0.345	0.222	0.139	0.372	0.169	0.356	0.115	0.235
CL4AC	0.553	0.349	0.226	0.143	0.374	0.168	0.368	0.115	0.242

Table 2: Performance of models on the Clotho v2 evaluation set. Baseline: baseline system described in Section 3.4, which is similar to our DCASE submitted system but without transfer learning and reinforcement learning techniques. CL4AC: Proposed framework Contrastive Loss for Audio Captioning (CL4AC). During the inference stage, captions are generated using greedy search.

3.4. Baseline system

The baseline system is similar to our DCASE 2021 system which uses transfer learning (TL) from external dataset and reinforcement learning (RL) [8]. Our motivation is to address the data scarcity problem for AAC, so we train the baseline without using TL from external dataset. Previous studies [23] proved that although RL techniques can optimize neural networks towards non-differentiable metrics, they may generate syntactically incorrect and incomplete captions. Thus, RL is also removed in the baseline system. The hyper-parameters used for training the baseline system are similar to the proposed model (mentioned in Section 3.3), except that the training batch size is 32 and the initial learning rate is 1×10^{-3}.

3.5. Evaluation

During the inference stage, the mel-spectrogram of an audio clip along with the special token “<sos>” are fed into the encoder and decoder separately to generate the first token. Afterwards, the following tokens are predicted in terms of the previously generated tokens until the token “<eos>” or the maximum length (35 words in our experiments) is reached. The greedy search strategy is used to generate captions.

We evaluate the performance of the proposed framework using the same metrics adopted in Task 6 of the DCASE 2021 Challenge, including machine translation metrics: BLEU_n [24], METEOR [25], ROUGE_l [26] and captioning metrics: CIDEr [27], SPICE [28], SPIDEr [29]. BLEU_n measures the quality of the generated text by calculating the precision of n-gram inside the text, which is an inexpensive metric to measure the correspondence between generated text and the ground truth. Generally, the higher BLEU_n usually implies better precision and fluent text. The SPIDEr, a combination of SPICE and CIDEr, is designed for image captioning task measurement, which considers scene graph inside the generated caption and the term frequency-inverse document frequency (TF-IDF) of the n-gram. By considering the scene graph and the TF-IDF of n-gram, the metric will focus on the relationships among objects and the text's property, which ensures the semantic fidelity to the audio and the syntactical fluency of the language.

4. RESULTS

Table 2 shows the performance of our proposed method on the Clotho v2 evaluation set. By adopting the contrastive loss technique during the training process, all the metrics except METERO increased on the evaluation set. For BLEU_1, BLEU_2, BLEU_3, BLEU_4, the relative improvement percentages for contrastive loss are 0.55%, 1.16%, 1.80%, and 2.88%, respectively. The n in BLEU_n means the n-grams matching between the predicted results and ground truths. The ascending increases of the relative improvement from BLEU_1 to BLEU_4 show that our proposed method generates more matching n-grams, demonstrating a more fluent and better quality captioning result. Besides, CIDEr and SPIDEr, the captioning metrics, obtained 3.37% and 2.98% relative improvement correspondingly. The better CIDEr and SPIDEr ensure the captions are better semantically faithful to the audio clip with the better language fluency. Numerical improvement of the machine translation and captioning metrics shows the effectiveness of CL4AC while trained with limited data.

5. CONCLUSIONS

This paper demonstrated the problem of data scarcity for AAC, which may lead to the inaccurate representation and audio-text alignment. To alleviate this issue, a novel encoder-decoder framework called **C**ontrastive **L**oss for **A**udio **C**aptioning (CL4AC) was proposed to learn a better cross-modal representation. In CL4AC, the self-supervision signals derived from the original audio-text data are used to exploit the correspondences between audio and text by contrasting samples in a limited dataset setting. Experiment results on BELU_n, CIDEr, and SPIDEr showed the effectiveness of the proposed approach with a relative improvement of up to 3.37%, compared to the baseline. In future work, we will explore more contrastive representation on audio-text data with different architectures, such as Momentum Contrast (MoCo) [30] and SimCLR [14].

6. ACKNOWLEDGMENT

This work is partly supported by grant EP/T019751/1 from the Engineering and Physical Sciences Research Council (EPSRC), a Newton Institutional Links Award from the British Council, titled “Automated Captioning of Image and Audio for Visually and Hearing Impaired” (Grant number 623805725) and a Research Scholarship from the China Scholarship Council (CSC) No. 202006470010.

References

[1] K. Drossos, S. Adavanne, and T. Virtanen, “Automated audio captioning with recurrent neural networks,” in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 374–378.

[2] A. Tran, K. Drossos, and T. Virtanen, “Wavetransformer: A novel architecture for audio captioning based on learning temporal and time-frequency information,” *arXiv preprint arXiv:2010.11098*, 2020.

[3] 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*, 2020, pp. 21–25.

[4] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, “A transformer-based audio captioning model with keyword estimation,” *arXiv preprint arXiv:2007.00222*, 2020.

[5] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[6] D. Takeuchi, Y. Koizumi, Y. Ohishi, N. Harada, and K. Kashino, "Effects of word-frequency based pre-and post-processings for audio captioning," *arXiv preprint arXiv:2009.11436*, 2020.

[7] A. Ö. Eren and M. Sert, "Audio captioning based on combined audio and semantic embeddings," in *2020 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2020, pp. 41–48.

[8] 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.

[9] W. Yuan, Q. Han, D. Liu, X. Li, and Z. Yang, "The DCASE 2021 challenge task 6 system: Automated audio captioning with weakly supervised pre-traing and word selection methods," DCASE2021 Challenge, Tech. Rep., July 2021.

[10] 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)*, 2019, pp. 119–132.

[11] 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 Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[12] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[13] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 661–18 673.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.

[15] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *International Conference on Learning Representations*, 2021.

[16] Q. Huang, T. Ko, H. L. Tang, X. Liu, and B. Wu, "Token-level supervised contrastive learning for punctuation restoration," *arXiv preprint arXiv:2107.09099*, 2021.

[17] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 578–10 587.

[18] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *European Conference on Computer Vision*. Springer, 2020, pp. 121–137.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 3111–3119.

[21] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "DialoGPT: Large-scale generative pre-training for conversational response generation," *arXiv preprint arXiv:1911.00536*, 2019.

[24] 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 of the Association for Computational Linguistics*, 2002, pp. 311–318.

[25] A. Lavie and A. Agarwal, "METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 228–231.

[26] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74–81.

[27] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575.

[28] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *European conference on Computer Vision*. Springer, 2016, pp. 382–398.

[29] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *Proceedings of the IEEE international conference on Computer Vision*, 2017, pp. 873–881.

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.