REVIEW

Automated Audio Captioning: An Overview of Recent Progress and New Challenges

Xinhao Mei*, Xubo Liu, Mark D. Plumbley and Wenwu Wang

Abstract

Automated audio captioning is a cross-modal translation task that aims to generate natural language descriptions for given audio clips. This task has received increasing attention with the release of freely available datasets in recent years. The problem has been addressed predominantly with deep learning techniques. Numerous approaches have been proposed, such as investigating different neural network architectures, exploiting auxiliary information such as keywords or sentence information to guide caption generation, and employing different training strategies, which have greatly facilitated the development of this field. In this paper, we present a comprehensive review of the published contributions in automated audio captioning, from a variety of existing approaches to evaluation metrics and datasets. We also discuss open challenges and envisage possible future research directions.

Keywords: Audio captioning; deep learning; audio processing; natural language processing; encoder-decoder framework

1 Introduction

Sound is ubiquitous in our daily lives. It carries a wealth of information about the environment, from sound scenes to individual events happening around us. For most people, the ability to perceive and understand the everyday sounds around us is taken for granted. However, mining helpful information from sounds is a challenging task for machines. With the development of machine learning, the field of machine listening has attracted increasing attention, with significant progress made in recent years, in areas such as audio tagging (AT) [1–5], sound event detection (SED) [6–8] and acoustic scene classification (ASC) [9, 10]. However, in these areas, the focus has been mostly on identifying acoustic scenes or events in an audio clip, rather than considering relationships between the audio events and acoustic scenes.

Automated audio captioning (AAC) aims at describing the content of an audio clip using natural language, which is a cross-modal translation task at the intersection of audio signal processing and natural language processing (NLP) [11]. Compared with automatic speech recognition (ASR), audio captioning focuses only on the environmental sounds and ignores the voice content that may be present in an audio clip. Compared with other popular audio-related tasks such as AT, SED, and ASC, audio captioning requires not only determining what audio events are present in the audio clip, but also describing these audio events using natural language, which allows the relationships between the audio events and the content of the audio clip to be summarized. An example caption may be “a person was walking on a sidewalk adjacent to a school where children were playing on the playground”[1] which describes the scenes and sound events given an audio clip. Generally speaking, audio captions are one-sentence descriptions of the predominant audio events and audio scenes occurring in the audio clips, where the detailed information may be included, such as the spatial-temporal relationships between the audio events and scenes, and the physical properties of sound objects and the acoustic environment.

Audio captioning has practical potential for various applications such as helping the hearing-impaired to understand environmental sounds, and analyzing sounds for video-based security surveillance systems. In addition, audio captioning can be used for multimedia retrieval [12, 13] in areas including education, film production, and web searching.

Unlike image and video captioning, which have been widely studied for almost a decade, audio captioning is a relatively new task that has been studied since

*Correspondence: x.mei@surrey.ac.uk
Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK
Full list of author information is available at the end of the article

[1] This caption is from the Clotho dataset.

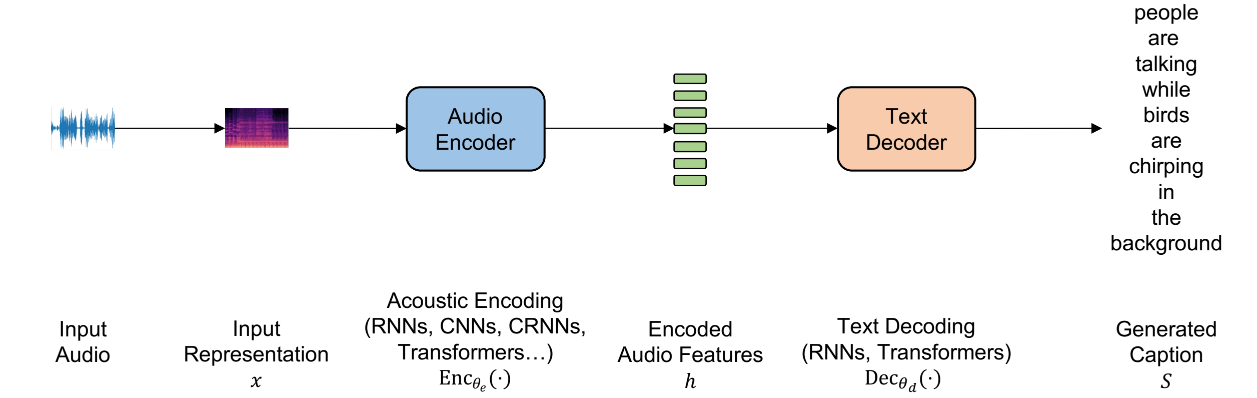
Audio Encoder
Text Decoder
people are talking while birds are chirping in the background
Input Audio
Input Representation x
Acoustic Encoding (RNNs, CNNs, CRNNs, Transformers…) $\mathrm{Enc}_{\theta_e}(\cdot)$
Encoded Audio Features h
Text Decoding (RNNs, Transformers) $\mathrm{Dec}_{\theta_d}(\cdot)$
Generated Caption S

Figure 1 Overview of an encoder-decoder-based AAC system, where the input is the waveform of an audio clip and the output is a natural language sentence describing the content of the input audio clip.

2017 [11]. In the past three years, this field has received increasing attention due to the release of freely available datasets and the organisation of a task in DCASE[2] Challenges from 2020 to 2022. A number of papers about audio captioning have been published, with deep learning being a popular method. Specifically, the encoder-decoder framework [14] has been adopted as a standard recipe for solving this cross-modal translation task. In this method, the encoder extracts audio features from the input audio clips, and the decoder generates captions based on the extracted audio features. Analyzing audio largely depends on obtaining robust audio features. Different kinds of neural networks, such as Recurrent Neural Networks (RNNs) [15], Convolutional Neural Networks (CNNs) [16], and Transformers [17], have been used as the encoders to learn feature representations. For the decoder, RNNs and Transformers are usually employed, inspired by works in NLP. In addition to the encoder-decoder framework, auxiliary information such as keywords or sentence information [18, 19], attention-based approaches [11, 20] and different training strategies [21–23] have been proposed to improve the performance of captioning systems. However, there is still a large gap between achieved results and human level performance [20].

To the best of our knowledge, no survey papers on audio captioning have been published so far. In this paper, we aim to provide a comprehensive overview of audio captioning with the hope of stimulating novel research ideas. Articles published up to April 2022 in the literature are considered in our survey. The encoder-decoder framework has been a standard recipe for AAC systems, therefore, we develop a taxonomy of acoustic encoding and text decoding approaches.

This paper is organized as follows. Section 2 introduces the preliminaries of audio captioning. In Section 3 and Section 4, we discuss acoustic encoding and text decoding approaches respectively. Auxiliary information is discussed in Section 5. We discuss training strategies adopted in the literature in Section 6. Furthermore, we review popular evaluation metrics and main datasets in Section 7 and Section 8, respectively. Finally, we discuss some open challenges and future research directions in Section 9 and briefly conclude this paper in Section 10.

2 Preliminaries of audio captioning

Existing methods for audio captioning are built predominantly on an encoder-decoder architecture where the captions are generated in an auto-regressive manner using deep learning techniques. We, therefore, take the popular encoder-decoder architecture as an example to introduce the preliminaries of an audio captioning system. Figure 1 shows the pipeline of an AAC system based on the encoder-decoder architecture.

Suppose we have a raw waveform of an input audio clip. Human-engineered features are usually extracted from the waveform as input representations for the audio encoder. Assume here that mel-spectrogram is used as the input representation, denoted by x, with a shape of $\mathbb{R}^{T\times F}$, where T is the number of time frames and F is the number of mel bins. The audio encoder takes the mel-spectrogram x as input and produces the encoded audio features h, which could be a single vector of shape $\mathbb{R}^{C}$, or a vector sequence of shape $\mathbb{R}^{T'\times C}$ where C is the dimension of the audio feature and T' is

[2] http://dcase.community/

the number of feature vectors, depending on the type of the encoder and the pooling method used for learning the encoded audio features. This process can be formulated as follows:

$$h = \mathrm{Enc}_{\theta_e}(x) \tag{1}$$

where θ_e are the model parameters of the encoder (Enc). More discussions about how the features are learned are given in Section 3.

After getting the encoded audio feature h, the decoder generates a sentence $S = \{w_1, ..., w_N\}$, where w_n is a word and N is the number of words in the sentence. The decoding process can be formulated as follows:

$$S = \mathrm{Dec}_{\theta_d}(h) \tag{2}$$

where θ_d are the model parameters of the decoder (Dec). Typically, the sentence is generated from the left (i.e. the first word) to the right (i.e. the final word) in an auto-regressive manner. That is, at time step t, the decoder predicts a posterior probability over the vocabulary, given the encoded audio feature h, a start token w_0, and previously generated words w_1 to w_{t-1}. Mathematically,

$$p(w_t|h, w_0, ..., w_{t-1}) = \mathrm{Dec}_{\theta_d}(h, w_0, ..., w_{t-1}), \tag{3}$$

where w_0 is a starting word of the sentence. After obtaining the word probability $p(w_t|h, w_0, ..., w_{t-1})$, the word w_t can be sampled by different decoding methods, such as greedy decoding or beam decoding [24]. The generation process is terminated when a stop token is generated or a maximum number of generation steps is reached.

3 Acoustic encoding

Analyzing the content of an audio clip largely depends on obtaining an effective feature representation for it, which is the aim of the encoder in an AAC system. The time domain waveforms are lengthy 1-D signals and it is challenging for machines to directly identify sound events or sound scenes from raw waveforms [25]. Current popular approaches for acoustic encoding consist of two steps, first extracting input representations, which are often hand-crafted features, such as spectrograms from the audio clip, and then passing them into a neural network to learn encoded compact audio features. In this section, we first discuss popular hand-crafted features used in literature, then audio encoding approaches, focusing on those based on deep neural networks.

3.1 Hand-crafted features

It is challenging for machines to directly understand an audio clip from its time domain representation. Hand-crafted features, inspired by the human auditory system have been widely used as sound representations for years [25]. In deep learning methods, these hand-crafted features are used as input representations to the neural networks to obtain encoded audio features. Time-frequency representations, such as spectrograms, are probably the most popular ones. To obtain a spectrogram, an audio signal is first split into short frames of length at around 20-60 ms, as these short time segments can be regarded as quasi-stationary [25]. Each time frame is shifted with a fixed time step. Then a window function is applied at each frame to enforce continuity and avoid spectral leakage at the frame boundaries [26]. The short time Fourier transform (STFT) is calculated for each time frame to get the spectrogram, a 2-D representation whose horizontal axis is time and vertical axis is frequency, the value at each point of the spectrogram represents the magnitude at a specific time and frequency. Inspired by the selectivity of human auditory system to different frequencies, the frequency axis of a spectrogram may be converted to different scales, resulting in representations such as mel-spectrogram and log mel-spectrogram [25]. The log mel-spectrogram generally leads to better performance when compared with other input representations in deep learning based methods for audio-related tasks [27–29], therefore, it is mainly used as the input representation in the literature. In addition, mel-frequency cepstral coefficients (MFCCs) were used in some early works [20, 30]. MFCCs are calculated by applying a discrete cosine transform (DCT) on log mel-spectrograms. Compared with time-frequency representations, MFCCs contain less information and are only able to estimate the global spectral shape of an audio clip [25], thus MFCCs are rarely used in recent works.

3.2 Neural networks

3.2.1 RNNs

RNNs are designed to process sequential data with variable lengths [15]. Audio is time series signal, therefore RNNs are naturally adopted as encoders in initial works [11, 30]. In a simple recipe, an RNN is used to model temporal relationships between the inputs, and the hidden states of the last layer of the RNN are regarded as the audio feature sequences, which are then fed into the text decoder for caption generation. Figure 2 shows the diagram of an RNN audio encoder. Drossos et al. [11] utilized a three-layered bi-directional gated recurrent unit (GRU) network [31] as the encoder. Further, unlike using multi-layer RNNs, Xu et

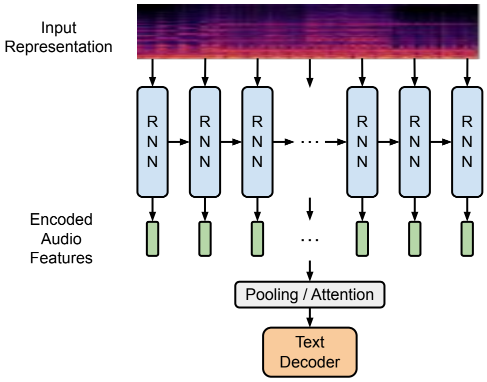

Figure 2 Diagram of an RNN audio encoder for acoustic encoding. The RNN encoder aims at modelling temporal relationships within the input representation. The encoded audio features usually have the same number of time frames as the input representation and interact with the decoder through a pooling or attention mechanism.

al. [19] and Wu et al. [29] used a single-layered uni-directional GRU network while Ikawa et al. [30] used a single-layered bi-directional long-short term memory (LSTM) network [32]. The encoded audio features output by RNNs usually have thousands of time steps, Nguyen et al. [33] argued that the length of the captions is significantly less than the length of the encoded audio features, making the captioning models difficult to learn the correspondence between words and audio features. They proposed a temporal sub-sampling method to sub-sample the learned features between the RNN layers, and showed that the temporal sub-sampling of audio features could be beneficial for audio captioning methods.

The main advantages of employing RNNs as encoders are their simplicity and their ability to process sequential data. However, using RNNs alone as the encoder is not found to give strong performance [20]. The reason might be that inputs are usually long sequences, RNNs may not be able to effectively model long-range time dependencies. In addition, getting an global audio feature from long hidden states also leads to excessive compression of information, making it difficult for the language decoder to generate fine-grained descriptions.

3.2.2 CNNs

CNNs have been applied with great success to the field of computer vision (CV) [16]. In recent years, CNNs have been adapted to audio-related tasks and show powerful ability in extracting robust audio patterns [27, 28]. Figure 3 shows the diagram of a 10-layer CNN audio encoder that is popularly used in the literature [34, 35].

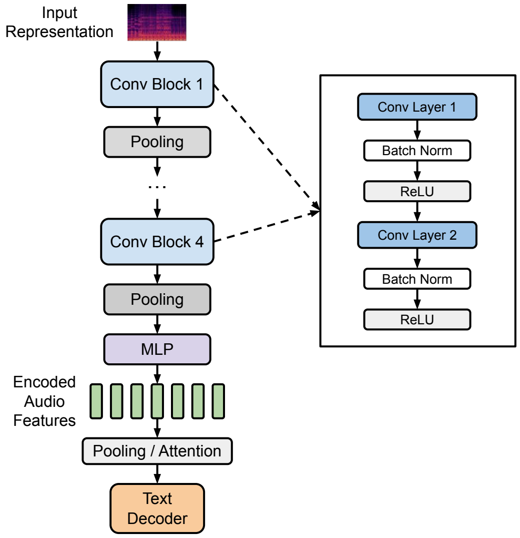

Figure 3 Diagram of a 10-layer CNN audio encoder. The input representation is first processed via four convolutional blocks and pooling layers, where each block consists of two convolutional layers. The feature maps output by the last convolutional block are then averaged along the frequency axis and fed into a two-layer multi-layer perceptron (MLP) to obtain encoded audio features.

Many CNN models pre-trained on large audio datasets have been published. Most works directly employ pre-trained CNN models as the audio encoder. VGG-like CNNs [34, 35] and ResNets [36–38] are popular choices as these networks perform well on audio-related tasks such as audio tagging and sound event detection [27]. In these works, CNNs treat the input spectrograms as 1-channel images, and model local dependencies within the spectrograms. Moreover, 1-D CNN is also incorporated to exploit temporal patterns. For example, Eren et al. [39] and Han et al. [37] used Wavegram-Logmel-CNN adapted from pre-trained audio neural networks (PANNs) [27]. The Wavegram-Logmel-CNN takes both raw waveform and spectrogram as inputs, which are processed using 1-D convolution and 2-D convolution, respectively. The outputs of 1-D convolutional layers and those of 2-D convolutional layers are combined in deep layers. Tran et al. [40] also proposed to utilise 1-D and 2-D convolutions for extracting temporal and time-frequency information. However, they only used spectrogram as input and reshape it for 1-D convolution. In summary, the use of 1-D convolution requires increased computation overhead, but offers only small performance improvement. The output feature maps of the convolutional blocks are generally in three dimensions, time, frequency and channels. To obtain encoded audio features, some methods use a global pooling along the time and frequency axis to obtain fixed-sized features [39], while others keep the time axis and apply pooling operation along the frequency axis to get a feature sequence [35, 36].

In summary, CNNs outperform RNNs, and are now the dominant approach for audio encoding. The main advantages of CNNs are that they are invariant to time shift and good at modeling local dependencies within the spectrograms. However, CNNs have limited receptive fields and modeling long-range time dependencies for long audio signals needs a deep CNN.

3.2.3 CRNNs

Motivated by the demand for modelling the local and long-range dependencies simultaneously, convolutional recurrent neural networks (CRNNs) [41], a combination of CNNs and RNNs, have also been applied as audio encoders. In a CRNN, RNN layers are introduced after the CNN layers to model the temporal relationship between extracted CNN features. Kim et al. [20] proposed a top-down multi-scale encoder where the features are extracted from two layers of the VGGish network [28], that is, a fully connected layer for extracting the high-level semantic features and a convolutional layer for extracting the mid-level features. Those features are then encoded by a two-layer bidirectional LSTM network where the semantic features are injected in the second layer. Takeuchi et al. [42] and Xu et al. [43] both adopted a similar CRNN encoder without using multi-level features. Xu et al. [44] compared CNN and CRNN encoders, and showed that a CRNN encoder outperformed a CNN encoder when the encoders are trained from scratch but the CRNN encoder brought little improvement when pre-training was applied. In summary, CRNNs need more computation than CNNs but offer limited improvement [44].

3.2.4 Other approaches

Transformers and their variants that are built on self-attention mechanism have been probably the most popular models in the fields of NLP and CV since 2017 [17, 45, 46]. Self-attention based encoders are also employed in recent works in audio captioning. Koizumi et al. [18] introduce a self-attention block after CNN layers in the encoder to learn the temporal relationship between CNN features. Mei et al. [47] proposed Audio Captioning Transformer (ACT), where the encoder is a convolution-free Transformer that directly models the relationships between the patches of the spectrogram. Figure 4 shows the diagram of the Transformer-based audio encoder in ACT. More details about Transformer and self-attention will be introduced in Section 4.2.2. ACT shows comparable performance with CNN-based methods while it may need more data for pre-training to obtain good performance. In addition to simply adding self-attention layers after convolutional layers, convolution and self-attention can be combined as in [48] by leveraging a convolution-augmented Transformer (Conformer) [49] to take advantage of their respective strengths. However, the Conformer encoder did not outperform the CNN encoders. The reason might be that they did not pre-train the Conformer encoder on a large-scale audio dataset.

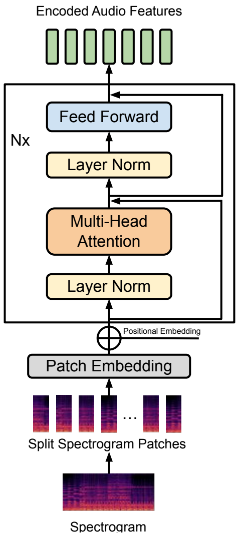

Figure 4 Diagram of the Transformer-based audio encoder. The input spectrogram is first split into small patches. These patches are then projected into 1-D embeddings through a linear layer, where a positional embedding is further added to each patch embedding to capture position information. The resulting embeddings are then fed into the Transformer blocks to obtain the encoded audio features.

In summary, various neural network architectures have been investigated as the audio encoder in order to obtain robust audio representations. CNNs are probably the most popular audio encoders and have achieved state-of-the-art performance. Early works adopted RNNs as encoders, but the trend has shifted from RNNs to CNNs. Recently, novel Transformer-based architectures have received increasing attention and have shown competitive performance in learning robust audio features as compared with the CNN encoders, however, they typically require more data for training to achieve similar performance, in comparison to the CNN networks [47].

4 Text decoding

The aim of the text decoder is to generate a caption given the audio features from the encoder. Existing works adopt an auto-regressive method for text generation, where each word in the caption is predicted based on the condition of the audio features extracted by the encoder and previously predicted words by the decoder. In addition to the main decoder block, a word embedding layer is used before the main decoder block to embed each input word into a fixed-dimension vector, so that discrete words can be processed by the network. In this section, we first introduce popular methods for obtaining word embeddings and then discuss main approaches for text decoding.

4.1 Word embeddings

A simple method to obtain word vectors is to represent each word as a one-hot vector, in which the element whose position corresponds to the index of the word in the vocabulary is set to one, while the remaining elements are set to zeros. This is called one-hot encoding. If the vocabulary is large, the dimension of the one-hot vector can be high. Hence, this method may suffer from the curse of dimensionality and the loss of semantic information [50]. Word embedding methods have become popular in recent years. Word embeddings are vectors of fixed-dimension, obtained by neural networks trained on large-scale text corpora. Semantically similar words are close to each other in the embedding space, while dissimilar words are far away from each other [51]. Examples of pre-trained word embeddings include Word2Vec [51], GloVe [52] and fastText [53], which are widely used in existing audio captioning works [18, 20, 34, 39, 43].

Recently, large-scale pre-trained Transform-based language models [54, 55] have shown powerful ability in language modelling thanks to the use of the self-attention mechanism. Weck et al. [56] employed BERT [54] to obtain word embeddings. They compared the effect of different pre-trained models on obtaining the word embeddings, and found that the BERT model leads to the best performance in obtaining word embeddings, while other models, such as Word2Vec and GloVe, provide slight improvement as compared to randomly initialized word embeddings.

In summary, word embeddings are semantic vector representations of words. They are generally stored in a matrix with the shape of $\mathbb{R}^{V \times d}$, where V is the size of the vocabulary and d is the dimension of the word vector. They can be retrieved using the indices of the words in the vocabulary.

4.2 Neural networks

4.2.1 RNNs

Sentences are also sequential data composed by discrete words, thus RNNs are popularly employed as the language decoder. Figure 5 shows a diagram of RNN-based language decoder. At each time step, the hidden state of the RNN is projected into a probability distribution along the vocabulary through a linear layer with a softmax activation function, and a word can be predicted accordingly.

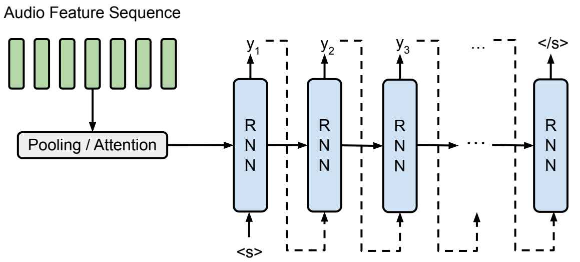

Figure 5 Diagram of an RNN-based language model. The RNN decoder generates the sentence from the left (i.e. the first word) to the right (i.e. the final word) in an auto-regressive manner, given the audio feature sequence generated from the encoder and previously generated words by the decoder. A start token '<s>' is fed into the RNN at the first time step to start the generation, while the generation process is terminated when a stop token '</s>' is generated.

Drossos et al. [11] proposed a 2-layer GRU network as the decoder in their initial work. Many subsequent works have adopted single-layer RNNs, either GRU or LSTM networks [20, 30, 33, 36, 57]. The main differences among these works are on how the audio features generated by the encoder are fused with the decoder. In a simple recipe, a global audio feature representation is obtained by applying mean pooling on the audio feature sequence extracted by the encoder, which is then used as the initial hidden state of the RNN decoder [19, 29] or is injected to the RNN decoder at each time step [33, 39]. This simple mean pooling method for getting a global audio representation is widely used in audio tagging task to detect what audio events are present in the whole audio clip [27]. However, this method does not consider the relationships between audio features, and thus, it is unable to capture the fine-grained information about audio events. These fine-grained information could be important for caption generation. Attention mechanism has been employed to overcome this problem [11]. When generating a word at each time step, the RNN decoder can attend to the whole audio feature sequence and place more weights on the informative audio features. Thus, the global audio representation at each time step is a different combination of the whole audio feature sequence. In addition, to exploit previously generated words, Ye et al. [36] introduced another attention module to attend to previously generated words at each time step.

In summary, RNNs with attention show reasonable performance in audio captioning and are widely used

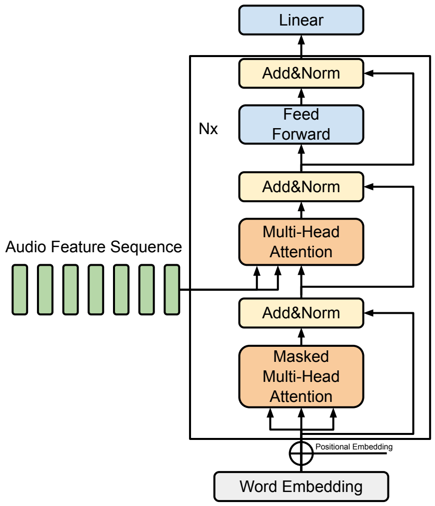

Figure 6 Diagram of a Transformer-based language model. When generating a word at each time step, the masked multi-head attention module attends to the previously generated words to exploit contextual information. The output of the masked self-attention module is then fused with the audio feature sequence from the encoder in the cross-attention module.

[36, 43]. The main disadvantage of RNNs is that they may be struggling to capture long-range dependencies between the generated words. Fortunately, audio captions are usually short in length, thus RNN decoders do not need to model very long-range dependencies.

4.2.2 Transformers

Since Vaswani et al. [17] proposed the Transformer network in 2017, the self-attention mechanism, which is the core of Transformer, has quickly become the basic building block in large language models. Transformer-based models such as BERT [54], GPT [55] and BART [58] outperform RNNs in language modelling and dominate most of the tasks in the field of NLP. Transformer-based models have also been employed in audio captioning works recently and achieved state-of-the-art performance.

Transformers are built on the self-attention mechanism. The self-attention module takes a sequence of N inputs and return N outputs, allowing each input directly interacts with others within the input sequence and finds out which they should pay more attention to. This makes it easier to model long range and global dependencies within the sequence, as compared with RNNs. Concretely, given an input sequence, the self-attention module first transforms the inputs into three representations, query vectors Q, key vectors K, and value vectors V by three learnable matrices. For each input, a scaled dot-product is first calculated between its query with respect to all keys to obtain the attention weights. After that, the attention weights are first converted to probabilities by a softmax function and then multiplied with each value and summed together to get the output. This can be formulated as:

$$\mathrm{Attn}(Q, K, V) = \mathrm{Softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{4}$$

where d_k is a scaling factor. In addition, this computation can be parallelized for all inputs through matrix multiplication, the training of Transformer could be more efficient than that of RNNs. The Transformer decoder is generally employed as the language decoder and has a stack of blocks, each of which consists of a masked self-attention module, a cross-attention module and a feed-forward layer module. Figure 6 shows a diagram of Transformer-based language decoder. The audio feature sequence obtained from the encoder interacts with the decoder through the cross-attention module, where K and V are derived from the audio features, while Q is obtained from the output of the masked self-attention module. Through these two attention modules, each word in the sequence can attend to the previously generated words and all the audio features, which may facilitate the model to capture the temporal relationships between audio events.

Due to the limited data available in audio captioning, many works use shallow Transformer decoders [34, 35, 37, 47], usually only two blocks, unlike in NLP tasks where very deep Transformers are often used [54, 58]. Some modifications to the standard Transformer architecture have also been investigated. For example, Xiao et al. [59] introduced an attention-free Transformer decoder to reduce computation overhead, which they claimed could better capture local information within audio features.

In summary, Transformer-based decoders show state-of-the-art performance in audio captioning, and are computationally more efficient from RNN-based decoders during training.

5 Auxiliary information

In addition to the standard encoder-decoder architecture, researchers have investigated the use of auxiliary information such as keywords or sentence information to guide caption generation. In this section, we discuss the auxiliary information used in the literature.

Audio signals have variable lengths and sound events can occur over arbitrary time frames, making direct mapping audio signals to sentences challenging. Furthermore, each sound event can be described with different words, which may lead to the word-selection indeterminacy problem [18]. To improve caption generation, keywords are widely employed. To obtain keywords, Kim et al. [20] retrieve the most similar training audio clip from AudioSet, the largest dataset for audio tagging available so far, and convert audio tagging labels of the retrieved audio clip into keywords. They then align these keywords with the audio features via an attention module and feed the output into the decoder. Some datasets may not have corresponding label information for each audio clip, and in this case, researchers first extract keywords or tags from human-annotated captions according to some rules such as frequency of the words and part-of-speech of the words [18, 34, 37, 57, 60]. Different methods were investigated to make use of the keywords. Cakir et al. [57] introduce a keyword decoder to estimate keywords of an audio clip, and jointly train the keyword decoder with the audio captioning model. Chen et al. [34] extract keywords from captions, and pre-train the audio encoder with an audio tagging task to enhance the ability of the encoder to learn robust audio patterns. Koizumi et al. [18] employ a keyword estimation branch after the encoder, combining the keywords with audio features before passing them to the language decoder. Ye et al. [36] utilize multi-scale features extracted by a CNN encoder for keyword prediction. However, some researchers found that keywords might not really improve the system performance in some situations. Takeuchi et al. [42] found that keywords may not work well when the model was trained from scratch. Ye et al. [36] claimed their model did not converge when only using keywords information. The accuracy of the keywords could be a bottleneck as wrong keywords might adversely impact on the captioning performance.

Sentence information has also been investigated. Ikawa et al. [30] introduce a 'specificity' term to measure the output text based on the amount of information it carries. The model is trained to generate captions whose 'specificity' is close to ground truth captions. Similarly, Xu et al. [19] introduce a sentence loss to generate captions closer to their ground truths in the sentence embedding space, employing a pre-trained language model BERT [54] to get the sentence embeddings.

Although different auxiliary information has been used to improve the caption generation process, these methods have not brought significant improvements and they may not work well for all datasets. In the DCASE challenge 2021, most teams still used the standard encoder-decoder model without using auxiliary information, and still achieved promising results [21, 35]. How to improve the AAC system with the auxiliary information still needs more investigation.

6 Training strategies

Supervised training with a cross-entropy (CE) loss is a standard recipe for training an audio captioning model. The main drawback of this setting is that it may cause 'exposure bias' due to the discrepancy between training and testing [61]. Reinforcement learning has been introduced to solve this problem and directly optimize evaluation metrics. In addition, transfer learning has been widely used to overcome the data scarcity problem. In this section, we discuss the popular training strategies used in the literature.

6.1 Cross-entropy training

The cross-entropy loss with maximum likelihood estimation (MLE) is widely used for training audio captioning models. During training, this approach adopts a 'teacher-forcing' strategy [61]. That is, the objective of training is to minimize the negative log-likelihood (equivalent to maximizing the log-likelihood) of current ground truth word given previous ground truth words at each time step. The cross-entropy loss can be formulated as follows:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t|y_{1:t-1}, x, \theta) \tag{5}$$

where y_t is the ground truth word at time step t, T is the length of the ground truth caption, x is the input audio clip, and θ are the parameters of the audio captioning model.

Models trained via the cross-entropy loss can generate syntactically correct sentences and achieve high scores in terms of the evaluation metrics [35]. However, there are also some disadvantages. First, the 'teacher forcing' strategy brings the problem known as 'exposure bias' [61], that is, each word to be predicted is conditioned on previous ground-truth words in the training stage, while it is conditioned on previous output words in the test stage. This discrepancy leads to error accumulation during text generation in the test stage. Second, models tend to generate generic and simple captions even though each audio clip has multiple diverse human-annotated captions in the training set [62]. This is because the MLE training tends to encourage the use of highly frequent words appearing in the ground truth captions.

6.2 Reinforcement learning

Xu et al. [43] employ a reinforcement learning approach to solve the 'exposure bias' problem and directly optimize the non-differentiable evaluation metrics. In a reinforcement learning setting, the captioning model is regarded as an agent and a policy is determined by the model's parameters. The agent executes an action at each time step to sample a word according to the policy. Once a sentence is generated, the agent receives a reward of the generated sentence. The goal of training is to optimize the model to maximize the expected reward that could be any evaluation metrics. This can be formalized as minimizing negative expected reward:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -E_{w^s \sim p_\theta}[r(w^s)], \tag{6}$$

where w^s is a sampled caption from the model, r is the reward of the sampled caption and θ are the model parameters. The caption can be sampled via Monte-Carlo sampling [63], however, it is computationally expensive. Another computationally efficient method, self-critical sequence training (SCST) [61] is generally employed. SCST employs the reward of a sentence sampled by greedy search as baseline thus avoids learning an estimate of expected future rewards. The expected gradient with respect to a single sample caption $w^s \sim p_\theta$ can be approximated as:

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta) \approx -(r(w^s) - r(\hat{w}))\nabla_\theta \log p_\theta(w^s), \tag{7}$$

where $r(\hat{w})$ is the reward of a caption generated by the current model using a greedy search.

Reinforcement learning could substantially improve the scores of the evaluation metrics, although it is used to optimize only one metric. However, Mei et al. [35] found that reinforcement learning may impact adversely on the quality of generated captions by introducing repetitive words and incorrect syntax. This also implies that existing evaluation metrics may not correlate well with human judgements.

6.3 Transfer learning

Availability of audio captioning datasets is limited due to the challenging and time consuming process in data collection and annotation [64, 65]. To overcome the data scarcity problem, transfer learning is widely adopted. In the encoder of the captioning system, pre-trained audio neural networks such as VGGish [28] and PANNs [27], are widely used to initialize the parameters of encoders [35–37, 66, 67]. Xu et al. [44] investigated the impact of pre-training on audio captioning performance. They show that audio encoders pre-trained with an audio tagging task give the best performance. Weck et al. [56] compare four off-the-shelf audio networks. In all the cases, pre-trained audio encoders substantially improve the performance of the audio captioning system. In the language decoder, although a lot of pre-trained Transformer-based language models have been released in recent years, most of those models cannot be directly used as the language decoder, since the decoder needs to interact with audio features from the encoder via a cross-attention module. Koizumi et al. [68] utilize GPT-2 [55] to get word embeddings. Gontier et al. [69] fine-tune BART [58] conditioned on the pre-trained audio embeddings and tags to generate captions and achieve state-of-the-art performance. To leverage pre-trained BERT [54], Liu et al. [70] investigate the addition of cross-attention layers with randomly initialized weights in the pre-trained BERT models as the decoder and demonstrate the efficacy of the pre-trained BERT models for audio captioning.

In summary, pre-trained audio encoders have proved to be effective to get robust audio features and overcome the data scarcity problem, while how to incorporate existing large pre-trained language models into an audio captioning system still needs further investigation.

6.4 Other approaches

Contrastive learning has been a popular training method in CV tasks [75, 76], Liu et al. [23] and Chen et al. [73] both investigated using contrastive training to learn better alignment between audio and text. Specifically, an audio clip and its paired caption are regarded as a positive pair while audio clips with other non-paired captions are regarded as negative pairs. A contrastive loss is combined with the original cross-entropy loss to encourage the model to maximize the similarity of the embeddings from the positive pairs while minimizing the similarity of the embeddings from the negative pairs. Koh et al. [66] also proposed an auxiliary objective aiming at maximizing the similarity between latent space formed by audio features obtained by the encoder and the latent space formed by text features obtained by the decoder. Berg et al. [22] presented a continual learning approach for continuously adapting an audio captioning method to new unseen general audio signals without forgetting learned information. Mei et al. [62] argued that an audio captioning system should have the ability to generate diverse captions for a given audio clip or across similar audio clips like human beings. They proposed an adversarial training framework based on generative adversarial network (GAN) [77] to encourage the diversity of audio captioning systems. In addition, because ASR and AAC share similarities in translating sound into natural language, Narisetty et al. [74] proposed approaches for end-to-end joint modeling of speech recognition and audio captioning tasks.

A brief overview of the published audio captioning methods is shown in Table 1, which contains the type of deep neural networks used to encode audio information, the language models used to generate captions, and the key aspects of these methods in the final column.

Table 1 An overview of published methods for audio captioning.

Reference	Year	Audio Encoder	Text Decoder	Key aspects
Drossos et al. [11]	2017	RNN	RNN	Attention
Wu et al. [29]	2019	RNN	RNN	N\A
Xu et al. [19]	2019	RNN	RNN	Sentence similarity loss
Ikawa et al. [30]	2019	RNN	RNN	'Specificity' term
Kim et al. [20]	2019	CNN(VGGish)+RNN	RNN	Multi-scale features, Semantic attention
Nguyen et al. [33]	2020	RNN	RNN	Temporal subsampling
Cakir et al. [57]	2020	RNN	RNN	Multi-task learning (keywords)
Perez-Castanos et al. [71]	2020	CNN	RNN	Attention
Chen et al. [34]	2020	CNN	Transformer	Pre-trained encoder
Xu et al. [43]	2020	CRNN	RNN	Reinforcement learning
Takeuchi et al. [42]	2020	CNN+RNN	RNN	Keywords, sentence length estimation
Tran et al. [40]	2020	CNN	Transformer	1-D and 2-D CNN
Eren et al. [39]	2020	CNN(PANNs)+RNN	RNN	Keywords
Koizumi et al. [18]	2020	CNN(VGGish)+Transformer	Transformer	Keywords
Koizumi et al. [68]	2020	CNN(VGGish)	GPT-2+Transformer	GPT-2, similar captions retrieval
Xu et a. [44]	2021	CNN\CRNN	RNN	Attention, transfer learning
Mei et al. [35]	2021	CNN(PANNs)	Transformer	Transfer learning, reinforcement learning
Mei et al. [47]	2021	Transformer	Transformer	Full transformer network
Han et al. [37]	2021	CNN(PANNs)	Transformer	Weakly-supervised pre-training, keywords
Ye et al. [36]	2021	CNN(PANNs)	RNN	Keywords, attention
Gontier et al. [69]	2021	CNN(VGGish)	BART	YAMNet tags, BART
Narisetty et al. [48]	2021	CNN(PANNs)+Conformer	Transformer+RNN	ASR techniques
Liu et al. [23]	2021	CNN(PANNs)	Transformer	Contrastive learning
Won et al. [72]	2021	CNN(PANNs)	Transformer	Transfer learning
Berg et al. [22]	2021	CNN	Transformer	Continual learning
Weck et al. [56]	2021	CNN(VGGish,YAMNet,OpenL3,COALA)	Transformer	Transfer learning
Mei et al. [62]	2021	CNN(PANNs)	Transformer	GAN, diversity
Xiao et al. [59]	2022	CNN	Transformer	Attention-free Transformer
Liu et al. [70]	2022	CNN(PANNs)	BERT	Transfer learning, BERT
Chen et al. [73]	2022	CNN	Transformer	Transfer learning, contrastive learning
Koh et al. [66]	2022	CNN(PANNs)+Transformer	Transformer	Transfer learning, regularization
Narisetty et al. [74]	2022	Transformer	Transformer	Joint modelling of ASR and AAC

7 Evaluation metrics

Evaluating audio captions is a challenging and subjective task, because an audio clip can correspond to several correct captions that may use different words, grammar, and/or describe different parts of the audio clip. Existing works adopt the same evaluation metrics used in image captioning, where most of these metrics are borrowed from NLP tasks such as machine translation and summarization, and the remaining are designed specifically for image captioning. The automatic evaluation metrics compare the machine-generated captions with human-annotated references where the number of references for each audio clip may vary across different datasets. The number of parallel references will influence the evaluation results. Generally, more reference captions are favored to reduce the evaluation bias [78]. In this section, we first introduce the conventional rule-based metrics, and then discuss some novel model-based metrics.

7.1 Conventional evaluation metrics

BLEU. BLEU (BiLingual Evaluation Understudy) [79] is originally designed to measure the quality of machine-generated sentences for machine translation systems. BLEU calculates modified n-gram precision for n up to four: the counts for n-grams in candidate sentence is first collected and clipped by their corresponding maximum count in references, the clipped counts are then summed and divided by the total number of candidate n-grams, where the n-gram is a window consisting of n consecutive words. The modified n-gram precisions are averaged with uniform weights to account for both adequacy and fluency, where 1-gram matches account for adequacy while longer n-gram matches account for fluency. As precision tends to give short sentences higher scores, a brevity penalty is introduced to penalize short sentences.

ROUGE. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [80] includes a set of metrics proposed to measure the quality of a machine-generated summary [80]. ROUGE-L which is widely used in image and audio captioning is based on the matching of the longest common subsequence between a candidate and a reference caption. ROUGE-L first counts the length of the longest common subsequence between a candidate and a reference, which is then divided by the total lengths of the candidate and reference to get precision and recall respectively. An F-measure combining precision and recall is then calculated as the score of ROUGE-L, which favors recall more.

METEOR. METEOR (Metric for Evaluation of Translation with Explicit ORdering) [81] is also a metric to evaluate machine translation systems. METEOR calculates unigram precision and unigram recall based on an explicit word-to-word matching in terms of their surface forms, stemmed forms, and meanings between a candidate and one or more references. An F-mean placing most of the weight on recall is then computed. To take into account longer matches, unigrams that are in adjacent positions in candidate and references are grouped into chunks, a penalty based on chunks is introduced and combined with F-mean to give the final METEOR score.

CIDEr. CIDEr (Consensus-based Image Description Evaluation) [82] is an automatic consensus metric for evaluating image description quality. CIDEr also represents sentences using n-grams presented in them, where each n-gram is weighted by the term frequency inverse document frequency (TF-IDF) weights because n-grams that commonly occur in a dataset are likely to be non-informative. CIDEr computes the cosine similarity of weighted n-grams between candidate and references, which accounts for both precision and recall. Similar to BLEU, CIDEr considers higher order n-grams (up to four) to capture grammatical properties and richer semantics.

SPICE. SPICE (Semantic Propositional Image Caption Evaluation) [83] is an image captioning evaluation metric based on semantic content matching. SPICE parses both candidate and references into scene graphs in which the objects, attributes and relations are encoded. An F-score is then calculated based on the matching of tuples extracted from the candidate and reference scene graphs. SPICE ignores the properties of grammar and fluency of sentences but only focuses on semantic matching.

SPIDEr. SPIDEr [84] is proposed for evaluating image captions and used as the official ranking metric in the automatic audio captioning task in DCASE Challenge. SPIDEr is the average of SPICE and CIDEr: the SPICE score ensures captions are semantically faithful to the content, while CIDEr score ensures captions are syntactically fluent.

7.2 Model-based metrics

BERTScore. BERTScore [85] is an evaluation metric for text generation tasks. Unlike conventional metrics which almost all rely on surface-form similarity, BERTScore utilizes pre-trained BERT [54] contextual embeddings that can capture semantic similarity, distant dependencies and ordering. After getting contextual embedding of each token through BERT, BERTScore measures the similarity of each token between candidate and references through cosine similarity where each token is matched to the most similar token in the other sentences. The matched token pairs are used to calculate a precision, recall and an F1 measure. Importance weighting with inverse document frequency is also introduced to weight rare words more.

SentenceBERT. SentenceBERT [86] is not essentially an evaluation metric but a modification of the pre-trained BERT model [54]. The SentenceBERT model can be used to obtain fixed-sized sentence embeddings for input captions. The sentence embeddings are then used to calculate a similarity score, such as cosine similarity, Euclidean distance or other similarities, between candidate and reference captions. Compared with BERTScore that measures the similarity in token level, SentenceBERT can be used for audio captioning for similarity comparison in sentence level.

FENSE. FENSE (Fluency ENhanced Sentence-bert Evaluation) [87] is a model-based evaluation metric specifically proposed for audio captioning. FENSE utilizes the Sentence-BERT to derive sentence embeddings for candidate and reference captions, and calculates its average cosine similarity score. To capture grammar issues like repeated words or phrases and incomplete sentences, FENSE uses a separate pretrained error detector to penalize the Sentence-BERT scores when fluency issues are detected.

In summary, conventional rule-based metrics are widely used to evaluate the performance of audio captioning systems. Most of these metrics focus on n-gram or sub-sequence based matching between the generated and reference captions. CIDEr and SPICE, proposed for image captioning, have shown better correlation with human judgements in captioning tasks than those borrowed from NLP tasks [82, 84]. However, some authors have shown that these metrics still cannot resemble human judgment well [87, 88]. Model-based metrics have received increasing attention and shown better correlation with human judgements in NLP tasks, but they have not been widely used in captioning tasks so far. We introduce them here to encourage research effort for developing novel metrics for audio captioning.

8 Datasets

The release of high quality audio captioning datasets has greatly promoted the development of this area. Almost all existing datasets (except one) are collections of single-sentence English captions, however, these datasets differ in many aspects such as the number of audio clips, the number of captions per audio clip, and the length of each audio clip. These different characteristics will affect the design and the performance of the audio captioning model. We describe the details of existing datasets in this section. To better understand the datasets, we then use consensus score of previously introduced metrics to evaluate these datasets.

8.1 Datasets description

AudioCaps. AudioCaps [20] is the largest audio captioning dataset so far. All the audio clips are 10-

Table 2 Performances of some surveyed audio captioning methods on two main datasets. Scores are taken from the respective papers. Only single model performance is considered. Compared to Clotho v1, Clotho v2 introduces new audio clips into the training set and a new validation set, while retaining the same evaluation set. Some methods merge the new validation set into the training set, these methods are still evaluated using the same evaluation set. We report these results separately.

Dataset	Method	Year	BLEU$_1$	BLEU$_2$	METEOR	CIDEr	SPICE	SPIDEr
AudioCaps	Kim et al. [20]	2019	0.614	0.446	0.203	0.593	0.144	0.369
	Koizumi et al. [68]	2020	0.638	0.458	0.199	0.603	0.139	0.371
	Eren et al. [39]	2020	**0.710**	0.490	**0.290**	0.750	-	-
	Xu et al. [44]	2021	0.655	0.476	0.229	0.660	0.168	0.414
	Mei et al. [47]	2021	0.647	0.488	0.222	0.679	0.160	0.420
	Gontier et al. [69]	2021	0.699	**0.523**	0.241	**0.753**	**0.176**	**0.465**
	Liu et al. [70]	2022	0.671	0.498	0.232	0.667	0.172	0.420
Clotho v1	Drossos et al. [64]	2019	0.420	0.140	0.090	0.100	-	-
	Cakir et al. [57]	2020	0.409	0.156	0.088	0.107	0.040	0.074
	Nguyen et al. [33]	2020	0.417	0.154	0.089	0.093	0.040	0.067
	Perez-Castanos [38]	2020	0.469	0.265	0.136	0.214	0.086	0.150
	Tran et al. [40]	2020	0.489	0.303	0.143	0.268	0.095	0.182
	Takeuchi et al. [42]	2020	0.512	0.325	0.145	0.290	0.089	0.190
	Koizumi et al. [18]	2020	0.521	0.309	0.149	0.258	0.097	0.178
	Chen et al. [34]	2020	0.534	0.343	0.160	0.346	0.108	0.227
	Xu et al. [43]	2020	0.561	0.341	0.162	0.338	0.108	0.223
	Eren et al. [39]	2020	**0.590**	0.350	**0.220**	0.280	-	-
	Xu et al. [44]	2021	0.556	0.363	0.169	0.377	**0.115**	**0.246**
	Koh et al. [66]	2022	0.551	**0.369**	0.165	**0.380**	0.111	**0.246**
Clotho v2	Narisetty et al. [48]	2021	0.536	0.341	0.160	0.346	0.108	0.227
	Won et al. [72]	2021	0.564	0.376	**0.177**	0.441	**0.128**	0.285
	Ye et al. [36]	2021	0.577	-	0.174	0.419	0.119	0.269
	Han et al. [37]	2021	**0.585**	**0.392**	**0.177**	**0.474**	**0.130**	**0.302**
Clotho v2 + val set	Narisetty et al.[48]	2021	0.541	0.346	0.161	0.362	0.110	0.236
	Liu et al. [23]	2021	0.553	0.349	0.168	0.368	0.115	0.242
	Mei et al. [35]	2021	0.561	0.374	0.171	0.426	**0.124**	0.275
	Chen et al. [73]	2022	0.572	0.379	0.171	0.407	0.119	0.263
	Xiao et al. [59]	2022	**0.578**	**0.387**	**0.177**	**0.434**	0.122	**0.278**

Table 3 An overview of English-annotated datasets.

Dataset	# of audios	# of captions per audio	Audio duration	Vocab size	Avg caption lengths
AudioCaps	51 308	1, 5	10 s	5066	8.79
Clotho	5929	5	15-30 s	4365	11.33
MACS	3930	2, 3 ,4, 5	10 s	2776	9.24

seconds long and are sourced from AudioSet, a large-scale audio event dataset [1]. The audio clips are selected by following some selection qualifications that ensure the chosen audio clips are balanced with respect to the ground truth annotations (tags) in the original dataset and diverse in terms of content. The audio clips are annotated by crowdworkers through Amazon Mechanical Turk (AMT), annotators are provided with an audio clip with corresponding word hints and video hints, and are required to write a natural language description with provided information.

The official release of AudioCaps contains 51k audio clips and is divided into a training set, a validation set and a test set. Each audio clip in the training set contains one corresponding human-annotated caption while those in validation set and test set contain five corresponding captions. Audio clips in AudioSet are not freely available but can be extracted from YouTube videos. It is worth noting that some audio clips might be no longer downloadable, thus the number of downloadable audio clips might be different from the official release of AudioCaps. The statistics in Table 3 are reported based on the official release version of AudioCaps.

Clotho. Clotho [64] is the dataset used for official ranking of the submitted systems in the task 6 (Automated Audio Captioning) of DCASE challenges in 2020 and 2021. All the audio clips are sourced from the online platform Freesound [89] and are ranging almost uniformly from 15 to 30 seconds. Annotators are employed through AMT for crowdsourcing the captions. During the annotation process, only the audio signal was provided to the annotators, with no additional information such as word or video hints (different from AudioCaps), to avoid introducing biases.

The latest Clotho v2 published a development set containing three subsets. There are 3839 audio clips in the training set and 1045 audio clips in the validation and evaluation set, respectively. Each audio clip contains five human-annotated captions, ranging from 8 to 20 words long. In the DCASE challenges, all three of these published sets can be used to train the models, while the final performance is evaluated using a preserved testing split by the organisers. For reporting results for conference or journal papers, performances are assessed using the published evaluation set and some authors may include the validation set into training since the validation set is added in Clotho v2. As a result, the model performance reported on Clotho may not all be on the same ground.

MACS. MACS (Multi-Annotator Captioned Soundscapes) [65] consists of audio clips from the development set of TAU Urban Acoustic Scenes 2019 dataset [90]. The audio clips are all 10-second long recorded from three acoustic scenes (airport, public square and park) and are annotated by students. The annotation process contains two stages. Given a list of ten classes and an audio clip, the annotators are first required to select the audio events presented in an audio clip from the given class list. Afterwards, the annotators are required to write a description of the audio clip.

MACS contains 3930 audio clips without being split into subsets. The number of captions per audio clip varies in the dataset. Most audio clips have five corresponding human-annotated captions, while some of them may only have two, three or four.

AudioCaption AudioCaption is a domain-specific Mandarin-annotated audio captioning dataset. Two scene-specific sets have been published: one for hospital scene [29] and another for car scene [19]. The hospital-scene set contains 3707 audio clips with three captions per clip while the car-scene set contains 3602 audio clips with five captions per clip. All the audio clips are annotated by native Mandarin speakers.

8.2 Datasets evaluation

Since all the datasets mentioned above are annotated under different protocols, they show different characteristics such as the number of captions per audio clip, caption lengths and sample variance in multi-reference captions. We believe these characteristics will influence the design and performance of audio captioning models. To better understand the datasets, we evaluate three English-annotated datasets from different aspects. Table 2 reports the performance of some surveyed methods on two main datasets, AudioCaps and Clotho, for which some methods listed in Table 1, such as [11, 19, 30], are not considered as they were not evaluated on these two datasets. Table 3 summarizes the datasets with some basic statistics. In addition, we use a consensus score [78] to represent the agreement among the parallel reference captions for the same audio clip, and the results are shown in Table 4. The consensus score c among n parallel reference captions $\mathcal{R} = \{r_i\}_{i=1}^{n}$ for an audio clip is defined as:

$$c = \frac{1}{n}\sum_{i=1}^{n} \text{metric}(r_i, \mathcal{R}\backslash r_i) \tag{8}$$

where r_i is the i-th caption and the metric can be anyone mentioned above. Since the number of references are varied among different datasets, we report the consensus score of AudioCaps using the validation and test set, Clotho using the training set and MACS using all the audio clips having five reference captions.

As the consensus scores are computed among the human-annotated captions, they can be also regarded as upper bound human-level performance on each dataset. As can be seen from Table 4, the consensus scores on AudioCaps and Clotho are close to each other except that the SentenceBERT score on AudioCaps is clearly higher than that of Clotho. Surprisingly, the consensus scores on MACS are lower than the other two datasets while only the SentenceBERT is close to them. This may reveal that the human-annotated captions in MACS are more diverse than the other two, and SentenceBERT can better capture semantic relevance between diverse captions. The consensus scores can be regarded as a measure of the dataset quality to some extent.

9 Challenges and future directions

Many deep learning-based methods have been proposed to improve automated audio captioning systems, and this task has seen rapid progress in recent years. However, there is still a large gap between the performance of the resulting systems and human level performance. In this section, we discuss challenges remaining in this area and envisage possible future research directions.

9.1 Data

There are two main challenges about data for audio captioning. First, the data scarcity problem is still a main challenge. Existing datasets are limited in size. The collection of an audio captioning dataset is time consuming, and it is hard to control the quality of human-annotated captions. Han et al. [37] collect weakly labeled dataset from online available sources to pre-train the AAC model and show that more training data (even weakly-labelled) can greatly improve the system performance. This reveals that we can make use of audio clips available online with their weakly-labelled text description to learn more robust audio-text representation, such as CLIP [91] in computer vision.

Second, existing datasets usually do not cover all possible real-life scenarios, and thus, audio captioning models cannot generalize well to different contexts. Martin et al. [65] investigate dataset bias of existing datasets from a lexical perspective. The bias problem still needs more investigation, e.g. how it will influence the model performance.

Table 4 Consensus scores of English-annotated datasets.

	BLEU$_1$	BLEU$_2$	BLEU$_3$	BLEU$_4$	ROUGE-L	METEOR	CIDEr	SPICE	SPIDEr	BERTScore	SentenceBERT
AudioCaps	0.65	0.48	0.37	0.29	0.49	0.28	0.90	0.21	0.56	0.52	0.64
Clotho	0.65	0.49	0.38	0.31	0.50	0.30	0.86	0.23	0.54	0.54	0.53
MACS	0.49	0.28	0.16	0.08	0.32	0.18	0.21	0.13	0.17	0.24	0.52

9.2 Model and training strategies

Existing AAC methods all follow the encoder-decoder paradigm and generate sentences in an auto-regressive manner. These two techniques have been the standard recipe for audio captioning models. Nonetheless, novel methods should be investigated in future research. For example, BERT-like architectures which fuse acoustic and textual modalities in early stage can be a replacement for the encoder-decoder paradigm, and work well in image captioning [92, 93]. Non-auto-regressive language models could reduce the inference time by generating all words in parallel [94], which might be a worthwhile research direction as it offers computational advantages, despite the fact that it under-performs the auto-regressive models in terms of captioning accuracy.

For the training strategies, the standard cross-entropy loss brings the problem of 'exposure bias' and tends to generate simple and generic captions. Although reinforcement learning is introduced to solve this problem, it may adversely affect the quality of generated captions. A promising line of research is to design new objective functions or add human feedback in a reinforcement learning setting to solve these problems. In addition, it requires more investigation on how to make use of learned knowledge in large pre-trained language models to improve caption generation.

9.3 Evaluation

The performance of audio captioning systems is generally assessed by objective evaluation metrics, since the human evaluation can be time-consuming and expensive. As discussed in the end of Section 7, existing objective metrics may not correlate well with human judgements [35, 87, 88], and none of them are designed specifically for audio captioning. Future work is expected to figure out to what extent the existing objective metrics correlate with human judgements, and to develop more reliable evaluation metrics.

9.4 Diversity and stylized captions

As argued in [95], a good captioning model should generate sentences that possess three properties: fidelity, i.e. the generated captions should reflect the audio content faithfully; naturalness, i.e. the captions should not be identified as machine-generated; diversity, i.e. the sentences should have rich and varied expressions, reflecting how different people would describe an audio clip in different ways. However, many existing approaches only consider semantic fidelity. Further research should be conducted to improve the other two properties. In addition, stylized captioning systems, which can generate outputs suitable for different audiences such as kids, could be a worthwhile research direction.

9.5 Other potential directions

There are also other potential directions for audio captioning. For example, temporal information of the sound events is not well used in existing works. Future work could investigate the use of information related to activities and timing information of sound events to generate more accurate captions. Information from other modalities could be also employed to train the audio captioning models, such as using audio-visual captioning methods [96, 97]. In addition, audio captioning can be potentially linked with other audio-language multi-modal tasks, such as audio-text retrieval [12, 98], audio question answering [99], text-based audio generation [100] and text-based audio source separation [101].

10 Conclusion

Audio captioning is a fast developing task involving both audio signal processing and natural language processing. In this paper, we have reviewed published audio captioning methods from the perspective of audio encoding and text decoding. We discussed auxiliary information employed to guide the caption generation, and training strategies adopted in the literature. In addition, main evaluation metrics and datasets are reviewed. We briefly outlined challenges and potential research directions in this area. We hope this survey can serve as a comprehensive introduction to audio captioning and encourage novel ideas for future research.

Acknowledgements

The authors acknowledge the insightful comments provided by the Associate Editor and the anonymous reviewers, which have added much to the clarity of the paper. For the purpose of open access, the authors have applied a creative commons attribution (CC BY) licence to any author accepted manuscript version arising.

Funding

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

Abbreviations

AT: Audio tagging, SED: Sound event detection, ASC: Acoustic scene classification, AAC: Automated audio captioning, NLP: Natural language processing, ASR: Automatic speech recognition, RNN: Recurrent neural network, CNN: Convolutional neural network, STFT: Short time Fourier transformer, MFCCs: Mel-frequency cepstral coefficients, DCT: Discrete cosine transform, GRU: Gated recurrent unit, LSTM: Long-short term memory, CV: Computer vision, CRNN: Convolutional recurrent neural network, MLP: Multi-layer perceptron, CE: Cross-entropy, MLE: Maximum likelihood estimation, GAN: Generative adversarial network.

Availability of data and materials

The datasets analysed during this article are available on the internet.

Authors' contributions

XM was a major contributor in writing the manuscript. XL summarized challenges and future work. MDP and WW substantially revised the manuscript. All authors read and approved the final manuscript.

Competing interests

WW is an editorial board member of EURASIP Journal on Audio Speech and Music Processing and also a guest editor of the special issue "Recent Advances in Computational Sound Scene Analysis", other authors declare that they have no competing interests.

Authors' information

The authors are with Centre for Vision Speech and Signal Processing (CVSSP), Department of Electrical and Electronic Engineering, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, GU2 7XH, UK. E-mail: [x.mei, xubo.liu, m.plumbley, w.wang]@surrey.ac.uk.

Author details

Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK.

References

1. 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017.
2. Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, 2017.
3. Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019.
4. H. Wang, Y. Zou, D. Chong, and W. Wang, "Modeling label dependencies for audio tagging with graph convolutional network," *IEEE Signal Processing Letters*, vol. 27, pp. 1560–1564, 2020.
5. Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*, May 2017.
6. Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time–frequency segmentation from weakly labelled data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 777–787, 2019.
7. Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with CNN-Transformer and automatic threshold optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2450–2460, 2020.
8. A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.
9. D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.
10. H. Wang, Y. Zou, and W. Wang, "SpecAugment++: A hidden space data augmentation method for acoustic scene classification," in *Interspeech*. ISCA, 2021, pp. 551–555.
11. 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.
12. A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, 2022.
13. X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "On metric learning for audio-text cross-modal retrieval," *arXiv preprint arXiv:2203.15537*, 2022.
14. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 3104–3112.
15. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
16. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
17. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
18. Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A Transformer-based audio captioning model with keyword estimation," in *INTERSPEECH*. ISCA, 2020, pp. 1977–1981.
19. X. Xu, H. Dinkel, M. Wu, and K. Yu, "Audio caption in a car setting with a sentence-level loss," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.
20. 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*, 2019, pp. 119–132.
21. X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2021 Challenge Task 6: audio captioning based on encoder pre-training and reinforcement learning," DCASE2021 Challenge, Tech. Rep., July 2021.
22. J. Berg and K. Drossos, "Continual learning for automated audio captioning using the learning without forgetting approach," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 140–144.
23. X. Liu, Q. Huang, X. Mei, T. Ko, H. Tang, M. D. Plumbley, and W. Wang, "Cl4ac: A contrastive loss for audio captioning," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 196–200.
24. A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
25. T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.
26. F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.
27. 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.

28. S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.
29. M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 830–834.
30. S. Ikawa and K. Kashino, "Neural audio captioning based on conditional sequence-to-sequence model," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 99–103.
31. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
32. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
33. K. Nguyen, K. Drossos, and T. Virtanen, "Temporal sub-sampling of audio feature sequences for automated audio captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 110–114.
34. 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.
35. X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. ZHAO, S. Li, T. Ko, H. Tang, X. Shao, M. D. Plumbley, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 206–210.
36. Z. Ye, H. Wang, D. Yang, and Y. Zou, "Improving the performance of automated audio captioning via integrating the acoustic and semantic information," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 40–44.
37. Q. Han, W. Yuan, D. Liu, X. Li, and Z. Yang, "Automated audio captioning with weakly supervised pre-training and word selection methods," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 6–10.
38. S. Perez-Castanos, J. Naranjo-Alcazar, P. Zuccarello, and M. Cobos, "Listen carefully and tell: An audio captioning system based on residual learning and gammatone audio representation," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 150–154.
39. 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.
40. 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.
41. B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.
42. D. Takeuchi, Y. Koizumi, Y. Ohishi, N. Harada, and K. Kashino, "Effects of word-frequency based pre- and post- processings for audio captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 190–194.
43. X. Xu, H. Dinkel, M. Wu, and K. Yu, "A CRNN-GRU based reinforcement learning approach to audio captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 225–229.
44. X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. 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)*. IEEE, 2021, pp. 905–909.
45. 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," in *International Conference on Learning Representations*, 2021.
46. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image Transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.
47. X. Mei, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Audio Captioning Transformer," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 211–215.
48. C. P. Narisetty, T. Hayashi, R. Ishizaki, S. Watanabe, and K. Takeda, "Leveraging state-of-the-art ASR techniques to audio captioning," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 160–164.
49. A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," 2020.
50. D. Jurafsky and J. H. Martin, *Speech and Language Processing (2nd Edition)*. USA: Prentice-Hall, Inc., 2009.
51. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.
52. J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
53. T. Mikolov, É. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
54. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," 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. 4171–4186.
55. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
56. B. Weck, X. Favory, K. Drossos, and X. Serra, "Evaluating off-the-shelf machine listening and natural language models for automated audio captioning," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 60–64.
57. E. Çakır, K. Drossos, and T. Virtanen, "Multi-task regularization based on infrequent classes for audio captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 6–10.
58. M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.
59. F. Xiao, J. Guan, H. Lan, Q. Zhu, and W. Wang, "Local information assisted attention-free decoder for audio captioning," *IEEE Signal Processing Letters*, vol. 29, pp. 1604–1608, 2022.
60. X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Automated audio captioning with keywords guidance," DCASE2022 Challenge, Tech. Rep., July 2022.
61. S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,

2017, pp. 7008–7024.

62. X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, “Diverse audio captioning via adversarial training,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8882–8886.
63. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
64. K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.
65. I. Martin and A. Mesaros, “Diversity and bias in audio captioning datasets,” in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 90–94.
66. A. Koh, X. Fuzhao, and C. E. Siong, “Automated audio captioning using transfer learning and reconstruction latent space similarity regularization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7722–7726.
67. H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, “Wav2CLIP: Learning robust audio representations from clip,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4563–4567.
68. Y. Koizumi, Y. Ohishi, D. Niizumi, D. Takeuchi, and M. Yasuda, “Audio captioning using pre-trained large-scale language model guided by audio-based similar caption retrieval,” *arXiv preprint arXiv:2012.07331*, 2020.
69. F. Gontier, R. Serizel, and C. Cerisara, “Automated audio captioning by fine-tuning bart with audioset tags,” in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 170–174.
70. X. Liu, X. Mei, Q. Huang, J. Sun, J. Zhao, H. Liu, M. D. Plumbley, V. Kılıç, and W. Wang, “Leveraging pre-trained BERT for audio captioning,” *arXiv preprint arXiv:2203.02838*, 2022.
71. S. Perez-Castanos, J. Naranjo-Alcazar, P. Zuccarello, and M. Cobos, “Listen carefully and tell: An audio captioning system based on residual learning and gammatone audio representation,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 150–154.
72. H. Won, B. Kim, I.-Y. Kwak, and C. Lim, “Transfer learning followed by Transformer for automated audio captioning,” in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 221–225.
73. C. Chen, N. Hou, Y. Hu, H. Zou, X. Qi, and E. S. Chng, “Interactive audio-text representation for automated audio captioning with contrastive learning,” *arXiv preprint arXiv:2203.15526*, 2022.
74. C. Narisetty, E. Tsunoo, X. Chang, Y. Kashiwagi, M. Hentschel, and S. Watanabe, “Joint speech recognition and audio captioning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7892–7896.
75. 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.
76. 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 (CVPR)*, June 2020.
77. L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: Sequence generative adversarial nets with policy gradient,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
78. W. Zhu, X. Wang, P. Narayana, K. Sone, S. Basu, and W. Y. Wang, “Towards understanding sample variance in visually grounded language generation: Evaluations and observations,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8806–8811.
79. 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.
80. C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Association for Computational Linguistics, 2004, pp. 74–81.
81. S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.
82. 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.
83. 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.
84. 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.
85. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating text generation with BERT,” in *International Conference on Learning Representations*, 2020.
86. N. Reimers, I. Gurevych, N. Reimers, I. Gurevych, N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, N. Reimers, I. Gurevych *et al.*, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019.
87. Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, “Can audio captions be evaluated with image caption metrics?” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 981–985.
88. J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser, “Why we need new evaluation metrics for NLG,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2241–2252.
89. F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013, pp. 411–412.
90. A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13.
91. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.
92. L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, “Unified vision-language pre-training for image captioning and VQA,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.
93. 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.
94. J. Gao, X. Meng, S. Wang, X. Li, S. Wang, S. Ma, and W. Gao, “Masked non-autoregressive image captioning,” *arXiv preprint arXiv:1906.00717*, 2019.
95. B. Dai, S. Fidler, R. Urtasun, and D. Lin, “Towards diverse and natural image descriptions via a conditional gan,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2970–2979.
96. Y. Tian, C. Guan, J. Goodman, M. Moore, and C. Xu, “An attempt towards interpretable audio-visual video captioning,” *arXiv:1812.02872*, 2018.
97. “A better use of audio-visual cues: Dense video captioning with bi-modal Transformer,” in *British Machine Vision Conference (BMVC)*, 2020.
98. X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, “Language-based audio retrieval with pre-trained models,”

DCASE2022 Challenge, Tech. Rep., July 2022.

99. H. M. Fayek and J. Johnson, "Temporal reasoning via audio question answering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2283–2294, 2020.
100. X. Liu, T. Iqbal, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Conditional sound generation using neural discrete time-frequency representation learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1–6.
101. X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate what you describe: Language-queried audio source separation," *arXiv:2203.15147*, 2022.