MusCaps: Generating Captions for Music Audio

Ilaria Manco†‡, Emmanouil Benetos*, Elio Quinton† & György Fazekas*
†School of EECS, Queen Mary University of London, U.K.
i.manco@qmul.ac.uk, emmanouil.benetos@qmul.ac.uk, elio.quinton@umusic.com, g.fazekas@qmul.ac.uk

Abstract—Content-based music information retrieval has seen rapid progress with the adoption of deep learning. Current approaches to high-level music description typically make use of classification models, such as in auto-tagging or genre and mood classification. In this work, we propose to address music description via audio captioning, defined as the task of generating a natural language description of music audio content in a human-like manner. To this end, we present the first music audio captioning model, MusCaps, consisting of an encoder-decoder with temporal attention. Our method combines convolutional and recurrent neural network architectures to jointly process audio-text inputs through a multimodal encoder and leverages pre-training on audio data to obtain representations that effectively capture and summarise musical features in the input. Evaluation of the generated captions through automatic metrics shows that our method outperforms a baseline designed for non-music audio captioning. Through an ablation study, we unveil that this performance boost can be mainly attributed to pre-training of the audio encoder, while other design choices – modality fusion, decoding strategy and the use of attention – contribute only marginally. Our model represents a shift away from classification-based music description and combines tasks requiring both auditory and linguistic understanding to bridge the semantic gap in music information retrieval.

I. INTRODUCTION

Current music information retrieval (MIR) approaches to music description typically rely on single- or multi-label classification. A prominent example is music auto-tagging [1]–[3], in which descriptive keywords are assigned to a music clip so as to convey high-level characteristics of the input such as genre, instrumentation and emotion. While this offers a reasonable starting point to modelling high-level representations of music, its output is limited to a predefined set of categorical labels. This in turns limits its usefulness in applications such as music search and recommendation, which would benefit from being able to both process and generate more human-like, detailed and nuanced descriptions. It is in fact through natural language that we often query music collections and search for known and unknown music content. A significant part of our daily language is also devoted to describing the musical world, offering an interface between music as mere audio signals and the set of abstractions used to describe it. While a wealth of music information is encoded in text, research in machine listening and MIR has traditionally overlooked the relationship between audio and natural language.

In this study, we focus on the novel task of music captioning, which we define as the ability to extract, disentangle and reason about high-level musical concepts in music audio, and then map these to the text modality by generating syntactically and semantically correct sequences of words. This is significantly harder than other, more common music description tasks such as classification and recognition, but presents a solution to some of their limitations. Captioning systems not only need to recognise signal-level features such as instrumentation and high-level descriptors such as genre, but they must also encode the relationship between them, thus better capturing the nuances of musical content; they also produce fully formed, descriptive sentences, that more closely match human queries. Through its joint use and processing of audio and linguistic information, music captioning also provides a first step towards the development of audio-and-language models for music understanding.

Finally, music captioning has several useful applications, such as producing descriptions for items in large music catalogues or vast collections of amateur and user-generated content; automatically generating evocative descriptions of music in films and videos for deaf and hard-of-hearing people; enabling search and discovery of music through more human-like queries; and providing explanations for automatic music recommendations.

To the best of our knowledge, this is the first work on music audio captioning. In the absence of benchmark datasets and established research on this task, we build upon previous literature on image and audio captioning and compare our model performance to a baseline sequence-to-sequence model. Differently from pioneering work on neural architectures for audio captioning, typically composed of an audio encoder and a text decoder, we propose a multimodal encoder that learns a joint representation of both audio and text to better account for the need to capture high-level semantics and summarise information that emerges at different levels of granularity in the input. We also show that, while audio-text data in the music domain is hard to obtain, this data scarcity issue can be effectively alleviated in the context of music captioning by employing suitable pre-trained audio representations. To accomplish this, we leverage large-scale pre-training on a publicly available music dataset and investigate the role of this pre-training step when only a smaller corpus is available for the downstream task of music audio captioning.

The main contributions of this work can be summarised as follows: (i) we propose the first music captioning model for...
II. RELATED WORK

A. Vision & Language

Within the field of machine perception, image caption generation has long been studied [4], [5]. Thanks to enormous progress in natural language processing (NLP) and computer vision, coupled with the increased availability of large-scale image-text datasets, research in this field has recently focussed on developing models to solve several vision-and-language (V&L) tasks. Most V&L models can be classified under two paradigms: CNN-RNN [4]–[6] and Transformer-based [7]–[10]. The former encompasses a family of fusion-based models in which convolutional neural network (CNN) architectures are employed as feature extractors, while recurrent neural networks (RNN), are employed as language models conditioned on the visual input. More recently, the success of BERT [11] has motivated research into its adaptation to V&L tasks, allowing to achieve state-of-the-art (SOTA) performance in all of them, both for still image and video [12], [13].

B. Audio & Language

While multimodal tasks have long been studied in the visual domain, audio-and-language research has only recently started to emerge, with the first audio captioning model proposed in [14]. Following a similar development to its visual counterpart, audio captioning has seen a rapid progress over the last few years [15]–[22], greatly encouraged by the recently introduced DCASE Challenge dedicated to the task[2] Most prior audio captioning methods make use of encoder-decoder models, frequently including sequence modelling modules, such as RNNs [20] or variants like gated recurrent units (GRU) [14], [17] and long short-term memory (LSTM) networks [18], in their encoder to take care of the temporal structure of audio inputs. Most of these works also make use of attention mechanisms to align the audio and text modalities [14], [15], [18], [21]. More recently, following the success of self-attention in V&L models, a small body of work has also started exploring the use of Transformer-based models in audio captioning [19], [23].

Our work is inspired by CNN-RNN architectures developed for image and audio captioning, but focusses on how such approaches can be extended to the music domain for the first time. The only prior works that attempt a similar goal can be found in [24] and [25]. However, the method proposed in [24] fails to generate grammatically correct sentences, while [25] simplifies the task by reframing it as the generation of a sequence of tags. Similarly, prior work on audio-text representation learning also makes use of tags [26], while we stress that our approach focusses on natural language.

C. Transfer Learning for MIR Tasks

Finally, one of the underlying ideas of our study is to leverage pre-training on music data to alleviate the issue of data scarcity in tasks such as captioning where parallel data across modalities is required. Tasks such as music tagging have been shown to successfully extract salient characteristics and learn musically relevant concepts such as mood, genre, era, and emotional content [1], [3]. This suggests that the data representations learnt by networks designed for these tasks can act as useful descriptors for more complex tasks that are highly dependent on similar properties of the input data.

\[^{2}\text{http://dcase.community/challenge2020/task-automatic-audio-captioning}\]
Previous work has demonstrated the benefit of using these pre-trained features in a transfer learning setting for regression and classification tasks in the music domain [27], [28], but this has not yet been explored in a multimodal setting.

### III. Proposed Method

Our model is composed of five main building blocks: a text embedding module, an audio feature extractor, a multimodal encoder, an attention mechanism and a natural language decoder. An overview of the architecture is presented in Fig. 1. In what follows, we offer a detailed description of each component.

#### A. Text Embedding

The text input is tokenized and encoded through an embedding matrix of dimensions $V \times d$, where $V$ is the vocabulary size and $d$ is the word embedding dimension. This is initialised with 300-dimensional GloVe word embeddings [29], pre-trained on Wikipedia 2014 and Gigaword 5 data, and kept frozen while training our model. Each sentence is thus encoded as a sequence $S = \{w_1, ..., w_T\}$, $w_t \in \mathbb{R}^d$, where $T$ is the variable length of the sentence.

#### B. Audio Feature Extractor

The audio feature extraction module is designed to take a variable-length raw audio input, split it into fixed-length chunks and extract features to be fed to the encoder. In summary, its role is to encode audio into a fixed-length representation or a sequence thereof. Inspired by the established practice of using CNN-based image feature extraction for vision-and-language tasks [7], [9], [12] and further motivated by the development of similar feature extraction networks specifically designed for music audio data [3], [28], we make use of convolutional neural networks for music auto-tagging as our feature extractors.

Among several architectures, we select two in our study, based on their performance on the music tagging task and their musically informed design: Musicnn [3], which combines vertical and horizontal convolutional filters to capture both timbral and temporal features, and Harmonic CNN [30], which uses trainable filters to model harmonic structures in the audio. A comprehensive comparison of these two networks, among other architectures, is presented in [31]. We discard the classification layers of the networks, using only the front-end and convolutional layers in our feature extraction, and use pre-trained weights from [31].

For each $n$-second chunk, where $n$ is the input length of the feature extraction network (3s for musicnn and 5s for Harmonic CNN, both at a sampling rate of 16 kHz), we obtain a feature vector of dimension $k$, such that the whole input sequence is encoded by a variably sized set of $L$ audio features $A = \{a_1, ..., a_L\}$, $a_i \in \mathbb{R}^k$. When no attention mechanism (Section III-D) is included in the model, we then obtain track-level features by applying average pooling on the feature maps across the time dimension. This yields $\tilde{a} \in \mathbb{R}^k$, a track-level vector representation of the input audio.

#### C. Multimodal Encoder

In its baseline variant, our multimodal encoder consists of a 1-layer unidirectional LSTM. Using the index $t = 1, ..., T$ to denote the $t$-th word in a caption, the $t$-th hidden state of the encoder LSTM is given by

$$ h_t^{enc} = \text{LSTM} \left( x_t^{enc}, h_{t-1}^{enc}, m_{t-1} \right), $$

where $x_t^{enc}$ is the input and $m_t$ is the cell state at step $t$.

In order to investigate the effect of using a multimodal input, we choose to perform modality fusion at different stages of the encoding procedure. In the early fusion approach, embeddings of the audio-text input pair are concatenated and passed to the LSTM as an input at each step:

$$ x_t^{enc} = [a, w_t], $$

where $a$ is obtained by passing the track-level audio embedding $\tilde{a}$ generated as described in the previous section through a fully connected layer. In the late fusion approach, the encoder LSTM only takes the text as input ($x_t^{enc} = w_t$) and its hidden state $h_t^{enc}$ is then concatenated with $a$ only prior to being passed to the decoder:

$$ x_t^{enc} = [a, h_{t-1}^{enc}]. $$

In this case, similarly to what was done for early fusion, $a$ is obtained by passing the audio embeddings through an additional linear layer, matching the dimension of the encoder hidden state ($W_a \tilde{a} \in \mathbb{R}^{H_{enc}}$). As illustrated in Fig. 2 in the early-fusion approach the audio content is therefore available to the encoder LSTM, while in the late-fusion approach the audio input does not influence the sequence dynamics modelled by the encoder.

---

D. Attention Mechanism

A key challenge in our multimodal encoder is to appropriately align and summarise the elements in the input audio to which an associated textual description is most sensitive to. We reduce this problem to tackling two distinct aspects: identifying the salient acoustic characteristics emerging from a global description of the input track and localising the temporal segments that are most strongly responsible for the associated description. An optimal approach would provide both and we note that the encoder described in the previous section may fall short of these requirements. In light of the shortcomings of compressing the whole audio input into a static representation, we adopt a soft attention mechanism to dynamically weigh different temporal sections in the audio input. For this, we obtain time-localised audio embeddings \( \alpha_i \) by using the audio features produced by the audio encoder over \( n \)-second segments prior to average pooling, as described in Section III-B.

The attention model used is a soft top-down attention mechanism, similar to approaches first proposed for the image captioning task in [5], [32]. This mechanism allows the decoder to attend to different sections of the input audio according to attention weights \( \beta_{ti} \), which are based on an alignment score \( e_{ti} \) learned from both the audio features \( \alpha_i \) and the hidden state of the encoder \( h_t^{enc} \):

\[
\begin{align*}
    e_{ti} &= w_{att}^T \tanh \left( W_{att}^a \alpha_i + W_{att}^h h_t^{enc} \right) \\
    \beta_{ti} &= \frac{\exp(e_{ti})}{\sum_{k=1}^N \exp(e_{tk})}
\end{align*}
\]

where \( w_{att} \in \mathbb{R}^{H_{dec}}, W_{att}^a \in \mathbb{R}^{H_{dec} \times k}, W_{att}^h \in \mathbb{R}^{H_{dec} \times H_{enc}} \) are parameters to be learnt and \( H_{enc} \) and \( H_{dec} \) are the number of hidden units in the encoder and decoder LSTM.

From (4), we obtain the attended vector \( \hat{a}_t \) as a weighted sum over the \( L \) \( n \)-second chunks forming the audio input:

\[
\hat{a}_t = \sum_{i=1}^L \beta_{ti} \alpha_i.
\]

E. Language Model Decoder

The input to the decoder LSTM varies depending on the fusion approach used at the encoding stage and whether the attention module is used. In the case of early fusion, it simply consists of the hidden representations produced by the encoder LSTM, while in the late-fusion case it is given by the audio features concatenated with the hidden states as in (3).

Finally, if the attention module is added to the model architecture, the attended audio feature replaces the mean-pooled vector in (4):

\[
x_t^{dec} = \left[ \hat{a}_t, h_t^{enc} \right].
\]

The overall architecture of the LSTM encoder, attention module and decoder is illustrated in Fig. 3.

A linear layer, with parameters \( W_d \) and biases \( b_d \), is then appended to the recurrent layers, followed by a softmax nonlinearity, thus acting as a classifier, where each class represents one of the words in the vocabulary. Its output is \( Y = \{ y_1, ..., y_L \} \), \( y_t \in \mathbb{R}^V \), a sequence of vectors representing the probability distribution over the word vocabulary at each step \( t \) in a \( T \)-long sequence:

\[
P(y_t | y_{t-1}) = \text{softmax}(W_d h_t^{dec} + b_d),
\]

such that the joint distribution

\[
P(Y | A) = \prod_{t=1}^T P(y_t | y_{t-1})
\]

gives the probability distribution over full text sequences \( Y \), given an audio input \( A \).

During training, a caption is generated through greedy decoding by selecting the word with the highest probability at step \( t = 1, ..., L \). At inference time, beam search decoding is also explored as a post-processing step. In this case, a fixed number, equal to the beam size \( b \), of most likely hypotheses up to step \( t \) is considered to generate the output at \( t + 1 \).

IV. Experiments

A. Dataset

In our experiments, we use a private production music\(^4\) dataset consisting of audio-caption pairs. We denote such a dataset \( D = \{ A_i, c_i \}_{i=1}^P \), where \( P \) is the number of pairs, \( A_i \) an audio track and \( c_i \) the corresponding caption. We filter out unsuitable examples by (i) keeping only tracks of length between 30 and 360 seconds; (ii) keeping only captions that contain between 3 and 22 tokens and that are not duplicated across the dataset. Through this cleaning procedure we retain

\(^4\)Production music is written and recorded with the aim of being licensed for synchronisation in audio or audiovisual productions such as films and adverts. It is typically organised in catalogues and provided with metadata and descriptions to facilitate discovery of suitable content.
forcing captions using standard automatic metrics. These can not significantly affect its performance.

Our model with smaller, randomly selected audio chunks does data splits as in the training and evaluation of our model. To 30s segment from each of our audio tracks, using the same the model is tuned to take as input audio samples between 15s decoder, with no alignment mechanism between the two. Since operating on the audio input, and one bidirectional GRU as the model consisting of three bidirectional GRUs as the encoder, our music captioning dataset. This is a sequence-to-sequence of the DCASE 2020 Challenge available baseline system used for the audio captioning task.

C. Evaluation

Training takes roughly one day on a single RTX 2080 Ti GPU.

B. Experimental Setting

Our caption generation model is trained using teacher forcing [33]. At each step t, the t − 1 element of the target caption is supplied as text input during training, while at test time this is replaced by the previous decoder output. In practice this is achieved by providing the special start token (sos) at t = 0 and stopping when the predicted output corresponds to the special end token (eos) or when the maximum length (22 in our experiments) is reached.

All our captioning models are trained by minimising a Cross Entropy Loss between the probability distribution over the candidate sentence and the ground-truth caption.

We set the number of hidden units in both the encoder (Henc) and the decoder (Hdec) to 256. As part of the training procedure we use Adam as an optimiser, with a simple learning rate schedule to linearly reduce the learning rate from an initial value of 10^-4. We train for a maximum of 200 epochs with a batch size of 16, making use of dropout with rate 0.25 and early stopping with a patience of 10 epochs for regularisation. Training takes roughly one day on a single RTX 2080 Ti GPU.

C. Evaluation

1) Baseline: We compare our method to the publicly available baseline system used for the audio captioning task of the DCASE 2020 Challenge[4] trained and evaluated on our music captioning dataset. This is a sequence-to-sequence model consisting of three bidirectional GRUs as the encoder, operating on the audio input, and one bidirectional GRU as the decoder, with no alignment mechanism between the two. Since the model is tuned to take as input audio samples between 15s and 30s at a sampling rate of 16 kHz, we randomly select a 30s segment from each of our audio tracks, using the same data splits as in the training and evaluation of our model. To ensure a fair comparison, we empirically verify that training our model with smaller, randomly selected audio chunks does not significantly affect its performance.

2) Captioning: In line with previous work in image and audio captioning, we provide an evaluation of the generated captions using standard automatic metrics. These can be divided into two groups: the first, comprising of BLEU [34], METEOR [35] and ROUGE_L [36], is a set of metrics first proposed to evaluate text generated through machine translation; the second, including CIDEr [37]. SPICE [38] and SPIDeR [39], was instead introduced for the evaluation of image captioning models. BLEU is computed from the geometric mean of the n-gram-based precision, using either only unigrams (BLEU_1), or combinations of n-grams up to 4, while METEOR and ROUGE_L are instead computed from an F-score based on matches between the candidate sentence and the reference one. Among the second set of metrics, CIDEr computes the cosine similarity between tf-idf weighted n-grams and is observed to better correlate with human judgement compared to previous metrics [37]. SPICE is based on a comparison of semantic similarity of scene graphs parsed from the candidate and reference caption, and SPIDeR is a linear combination of the two. We also note that these metrics have been found to be more accurate when more than one reference sentence is provided [37], while only one caption per example is available in our dataset.

3) Text-based Music Retrieval: We also evaluate our model on music audio retrieval based on a natural language query. The goal of this task is to retrieve the correct audio among a pool of candidate tracks, given its corresponding textual description. We note that our model was not trained on retrieval and its objective is not explicitly designed to optimise text-audio ranking, and therefore does not allow for a direct way to retrieve audio given a caption (or vice versa). However, the model can be easily adapted to perform retrieval by ranking each candidate audio track according to the conditional probability of generating the text query given the audio input, corresponding to the joint probability over the T words in the query sentence, as can be seen in (8). For this experiment, we use the 1207 audio tracks in our test set and 100 test queries from the set of ground-truth captions. Following prior work on image description [4], [6], we provide results for the following retrieval metrics: Recall @ K (higher is better) with k = {1, 5, 10}, the percentage of correctly retrieved items within the top-K results, and the median rank (lower is better) of the ground-truth items across all queries.

D. Results & Analysis

In this section, we discuss our experimental results for the captioning and retrieval tasks, comparing several variants of our model. We also perform an ablation study (Table [1]) to tease apart the relative contributions of the main design choices. In order to account for variance due to random initialization, we repeat each experiment in the ablation study three times and provide confidence intervals on our results.
For the captioning task, we look at: (1) comparison with the baseline; (2) the effects of large-scale pre-training of the audio feature extractor module; (3) the choice of modality fusion in the encoder and the role of the attention mechanism in providing temporal alignment between the audio embeddings and the representations learnt by the decoder; (4) additionally, we provide a comparison of model performance and caption quality with two different decoding strategies.

1) Comparison with the Baseline: In our first experiment, we focus on the comparison between different audio feature extractors (musicnn and Hcnn), pre-trained on the MagnaTagATune dataset [40] and compare the two resulting variants of our model, denoted as MusCaps-Musicnn and MusCaps-Hcnn. All experiments were run with the same settings, as detailed in Section III-B using early fusion and keeping the pre-trained audio feature extractor module frozen during training. For a fair comparison to the baseline, which does not include an attention mechanism, we only include results of our attention-free model in this section. Our experimental results, shown in Table IV, indicate that both variants of our model significantly outperform the DCASE baseline across all metrics, suggesting that the use of a musically informed audio feature extractor brings a considerable performance boost. Illustrative examples of the qualitative differences in output captions between our model and the baseline are also reported in Table IV

2) Effect of Pre-training: To further support our claim that pre-training the audio feature extractor module on the music auto-tagging task is indeed crucial to obtaining a good model performance, we empirically verify this by running experiments without initialising the audio encoder parameters with pre-trained weights, training them instead from scratch with the rest of the model. Due to memory constraints, we randomly select 20s chunks from the full track as our audio input. As hypothesised, our experimental results demonstrate that removing pre-training of the CNN component results in a significant performance drop, as shown in Table II and slower convergence. This can be attributed to the lack of sufficient training data to successfully accomplish both feature extraction and sentence generation end-to-end, which quickly leads to overfitting, and to a possible difference in the rates at which these two model components generalise. Similarly to what was done when comparing our model with the baseline, we also train MusCaps with pre-training using shorter, randomly selected chunks as the audio input, to ensure that this does not significantly affect performance, and find that only a small decrease in some of the scores is observed in this case.

3) Effect of Fusion & Attention: We next experiment with the two fusion strategies described in Section III-C and the use of a simple attention mechanism for temporal alignment (Section III-D), denoting the respective models as MusCaps\textsubscript{Early}, MusCaps\textsubscript{Late}, and MusCaps\textsubscript{Attn}. Overall, we do not observe a significant difference across these three variants. Remarkably, we find that, even when including the attention mechanism, the model performs similarly. We identify two main explanations for this. Firstly, the attention mechanism used assumes an alignment between audio segments and word tokens at a consistent temporal scale. In music captioning, however, an exact alignment between discrete units in the audio and text modalities may be too strong of an assumption and, if present, is likely to occur at different timescales. Secondly, as shown in Fig. II the learning curves of the ablation models reveal that MusCaps\textsubscript{Attn} converges faster than all other variants. This behaviour is more pronounced for CIDEr, but a similar trend is observed on all metrics. This suggests that the model may be overfitting and ultimately warrants further investigation.

4) Effect of Decoding Strategies: Finally, we compare performance when using greedy decoding and beam search. We find that a beam size of 5 brings the highest performance gain on the captioning metrics for MusCaps\textsubscript{Early}, while in
most other variants performance saturates quickly and even
degrades with a beam size larger than 3. This is consistent with
observations from previous studies [41], which attribute the
performance degradation to overfitting or discrepancy between
optimisation of the metrics and of the learning objective.

To better understand how caption quality if affected by the
use of different decoding algorithms beyond what is captured
by automatic metrics, we extract some statistics from the
predicted captions. When greedily decoding the text output,
we observe that, although the generated sentences generally
capture the audio content well, they often present undesirable
features, such as a particularly high occurrence of the 'unk'
token and the frequent repetition of n-grams ("and strings and
strings", "rock rock", "bass bass and bass"). When comparing
the output of greedy decoding to that obtained through beam
search, we observe a significant reduction in both of these.
On MusCapsEarly, beam search decoding with \( b = 5 \)
produces 20\% fewer 'unk' tokens and 37\% fewer repetitions.
A similar trend is observed in other architectural variants, such
as MusCapsAttn, although the improvement is more modest
(−11\% and −18\% respectively). As shown in Table [IV]
this results in more readable captions and we therefore argue that,
although not consistently reflected in the evaluation metrics,
beam search improves structure and is therefore beneficial to
caption generation overall.

5) Retrieval Performance: In Table [III] we report the audio
retrieval performance of the 4 model variants analysed in the
ablation study, comparing them to the baseline. The results
generally mirror what is observed in the captioning task:
pre-training of the audio feature extractor brings the highest
performance gain, while changes due to design choices such
as fusion strategy and the inclusion of the attention mechanism
are less substantial. Surprisingly, unlike in the captioning
results, the recall performance of the DCASE baseline model
is comparable to MusCaps without pre-training, while its
median rank is similar to the top-performing MusCaps vari-
ants. Overall, although our experiments demonstrate that the
model can be used for text-based retrieval, the scores are not
particularly high and it remains unclear whether this is due
to genuinely incorrect audio-text matching or to limitations of
the ranking procedure. It is worth noting, for example, that the
task itself may be partially ill-defined in this context, since
multiple textual descriptions may be acceptable for a given
audio clip (and vice versa), while our dataset, and therefore
our evaluation procedure, does not account for this. A more
thorough evaluation protocol may include similarity measures
between the retrieved and ground-truth items and user studies.

V. Conclusions & Future Work

In this paper, we presented the first music audio cap-
tioning model, MusCaps, a simple encoder-decoder network
consisting of a multimodal CNN-LSTM encoder with temporal
attention and an LSTM decoder. We formulate the problem
in a transfer learning setting and highlight the benefit of
leveraging music audio feature extractors pre-trained on large-
scale, publicly available datasets. Our experiments show that
our model can successfully generate descriptive sentences of a
music track and outperforms a general-purpose baseline model
for audio captioning, even when training on a small audio-
text corpus. This confirms that both a musically informed
architecture and the use of music tagging as a pretext task are
beneficial to music captioning. Although our results are
encouraging, our work offers several avenues for further
study. Among these, data-related limitations warrant particular
attention and we emphasise that additional data collection
or data augmentation techniques may be required to obtain
several reference captions per example and thus make metric-
based evaluation more reliable.
REFERENCES


