

TOWARDS MUSIC CAPTIONING: GENERATING MUSIC PLAYLIST DESCRIPTIONS

Keunwoo Choi, György Fazekas, Mark Sandler

Centre for Digital Music
Queen Mary University of London
keunwoo.choi@qmul.ac.uk

Brian McFee, Kyunghyun Cho

Center for Data Science
New York University
{first.last}@nyu.edu

ABSTRACT

Descriptions are often provided along with recommendations to help users' discovery. Recommending automatically generated music playlists (e.g. personalised playlists) introduces the problem of generating descriptions. In this paper, we propose a method for generating music playlist descriptions, which is called as music captioning. In the proposed method, audio content analysis and natural language processing are adopted to utilise the information of each track.

1. INTRODUCTION

Motivation: One of the crucial problems in music discovery is to deliver the summary of music without playing it. One common method is to add descriptions of a music item or playlist, e.g. *Getting emotional with the undisputed King of Pop*¹, *Just the right blend of chilled-out acoustic songs to work, relax, think, and dream to*². These examples show that they are more than simple descriptions and even add value to the curated playlist as a product.

There have been attempts to automate the generation of these descriptions. In [8], Eck et al. proposed to use social tags to describe each music item. Fields proposed a similar idea for playlist using social tag and topic model [9] using Latent Dirichlet Allocation [1]. Besides text, Bogdanov introduced music avatars, whose outlook - hair style, clothes, and accessories - describes the recommended music [2].

¹ Michael Jackson: Love songs and ballads by Apple Music ☞

² Your Coffee Break by Spotify ☞



© Keunwoo Choi, György Fazekas, Mark Sandler, Brian McFee, Kyunghyun Cho. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Keunwoo Choi, György Fazekas, Mark Sandler, Brian McFee, Kyunghyun Cho. "Towards Music Captioning: Generating Music Playlist Descriptions", Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference, 2016. This work was part funded by the FAST IMPACT EPSRC Grant EP/L019981/1 and the European Commission H2020 research and innovation grant AudioCommons (688382). Mark Sandler acknowledges the support of the Royal Society as a recipient of a Wolfson Research Merit Award. Brian McFee is supported by the Moore Sloan Data Science Environment at NYU. Kyunghyun Cho thanks the support by Facebook, Google (Google Faculty Award 2016) and NVidia (GPU Center of Excellence 2015-2016). The work is done during Keunwoo Choi is visiting Center for Data Science in New York University.

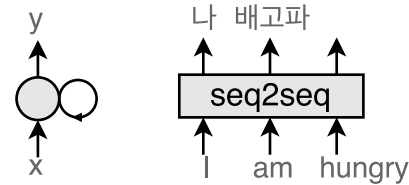


Figure 1. A block diagram of an RNN unit (left) and sequence-to-sequence module that is applied to English-Korean translation (right).

Background: • *RNNs*: RNNs are neural networks that have a unit with a recurrent connection, whose output is connect to the input of the unit (Figure 1, left). They currently show state-of-the-art performances in tasks that involve sequence modelling. Two types of RNN unit is widely used: Long Short-Term Memory (LSTM) unit [10] and Gated Recurrent Unit (GRU) [3].

• *Seq2seq*: Sequence-to-sequence (seq2seq) learning indicates training a model whose input and output are sequences (Figure 1, right). Seq2seq models can be used to machine translation, where a phrase in a language is summarised by an encoder RNN, which is followed by a decoder RNN to generate a phrase in another language [4].

• *Word2vec*: Word embeddings are distributed vector representations of word that aim to preserve the semantic relationships among words. One successful example is *word2vec* algorithm, which is usually trained with large corpora in an unsupervised manner [13].

• *ConvNets*: Convolutional neural networks (ConvNets) have been extensively adopted in nearly every computer vision task and algorithm since the record-breaking performance of AlexNet [12]. ConvNets also show state-of-the-art results in many music information retrieval tasks including auto-tagging [5].

The Problem: The problem of *music captioning* can be defined as *generating a description for a set of music items using on their audio content and text data*. When the set includes more than one item, it can be also called as *music playlist captioning*.

2. THE PROPOSED METHOD

Both of the approaches use sequence-to-sequence model, as illustrated in Figure 2. In the sequence-to-sequence model,

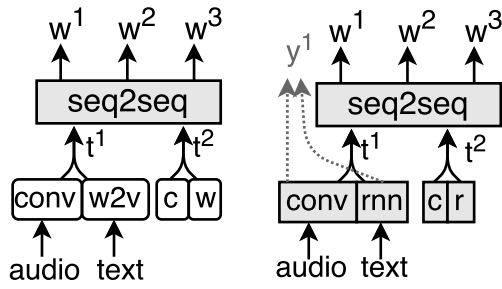


Figure 2. The diagrams of two proposed approaches, where coloured blocks indicate trainable modules. The first approach uses a pre-trained ConvNet (*conv*) and word2vec (*w2v*) and only sequence-to-sequence model is trained. In the second approach, the whole blocks are trained - a ConvNet to summarise the audio content, an RNN to summarise the text data of each track. An additional labels (*y*) such as genres or tags can be provided to help the training.

the encoder consists of two-layer RNN with GRU and encodes the track features into a vector, i.e., the encoded vector summarises the information of the input. This vector is also called *context vector* because it provides context information to the decoder. The decoder consists of two-layer RNN with GRU and decode the context vector to a sequence of word or word embeddings. The models are written in Keras and uploaded online³ [6].

2.1 Pre-training approach

This approach takes advantage of a pre-trained word embedding model⁴ and a pre-trained auto-tagger⁵. Therefore, the number of parameters to learn is reduced while leveraging additional data to train word-embedding and auto-tagger. Each data sample consists of a sequence of N track features as input and an output word sequence length of M , which is an album feature.

Input/Output⁶: A n -th track feature, $\mathbf{t}^n \in \mathbf{R}^{350}$, represents one track and created by concatenating the audio feature, $\mathbf{t}_a^n \in \mathbf{R}^{50}$, and the word feature, $\mathbf{t}_w^n \in \mathbf{R}^{300}$, i.e. $\mathbf{t} = [\mathbf{t}_a; \mathbf{t}_w]$. For computing \mathbf{t}_a , a convolutional neural network that is trained to predict tags is used to and output 50-dim vector for each track [5]. \mathbf{t}_w is computed by $\sum_k \mathbf{w}_k / K$, where \mathbf{w}_k refers to the embedding of k -th word in the metadata⁷. The word embedding were trained by *word2vec* algorithms and Google news dataset [13].

³<http://github.com/keunwoochoi/ismir2016-ldb-audio-captioning-model-keras>

⁴<https://radimrehurek.com/gensim/models/word2vec.html>

⁵https://github.com/keunwoochoi/music-auto_tagging-keras, [5]

⁶The dimensions can vary, we describe in details for better understanding.

⁷ Because these word embeddings are distributed representations in a semantic vector space, average of the words can summarise a bag of words and was used as a baseline in sentence and paragraph representation [7].

An playlist feature is a sequence of word embeddings of the playlist description, i.e. $\mathbf{p} = [\mathbf{w}^m]_{m=0,1,\dots,m-1}$.

2.2 Fully-training approach

The model in this approach include the training of a ConvNet for audio summarisation and an RNN for text summarisation of each track. The structure of ConvNet can be similar to the pre-trained one. The RNN module is trained to summarise the text of each track and outputs a *sentence vector*. These networks can be provided with additional labels (notated as y in the figure 2) to help the training, e.g., genres or tags. In that case, the objective of the whole structure consists of two different tasks and therefore the training can be more regulated and stable.

Since the audio and text summarisation modules are trainable, they can be more relevant to the captioning task. However, this flexibility requires more training data.

3. EXPERIMENTS AND CONCLUSIONS

We tested the pre-training approach with a private production music dataset. The dataset has 374 albums and 17,354 tracks with descriptions of tracks, albums, audio signal and metadata. The learning rate is controlled by ADAM [11] with an objective function of *1-cosine proximity*. The model was trained to predict the album descriptions.

The model currently overfits and fails to generate correct sentences. One example of generated word sequence is *dramatic motivating the intense epic action adventure soaring soaring soaring gloriously Roger-Deakins.cinematography Maryse-Alberti*. This is expected since there are only 374 output sequences in the dataset – if we use early stopping, the model underfits, otherwise it overfits.

In the future, we plan to solve the current problem – lack of data. The sentence generation can be partly trained by (music) corpora. A word2vec model that is trained with music corpora can be used to reduce the embedding dimension [14]. The model can be also modified in a sense that the audio feature is optional and it mainly rely on metadata. In that case, acquisition of training data becomes more feasible.

4. REFERENCES

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [2] Dmitry Bogdanov, MartiN Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management*, 49(1):13–33, 2013.
- [3] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

- [4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [5] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. In *International Society of Music Information Retrieval Conference*. ISMIR, 2016.
- [6] François Chollet. Keras. *GitHub repository*: <https://github.com/fchollet/keras>, 2015.
- [7] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.
- [8] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In *Advances in neural information processing systems*, pages 385–392, 2008.
- [9] Ben Fields, Christophe Rhodes, Mark d’Inverno, et al. Using song social tags and topic models to describe and compare playlists. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [11] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [13] T Mikolov and J Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 2013.
- [14] Sergio Oramas, Luies Espinosa-Anke, Shuo Zhang, Horacio Saggion, and Xavier Serra. Natural language processing for music information retrieval. In *17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016.