Title	Data-efficient playlist captioning with musical and linguistic knowledge
Author(s)	Gabbolini, Giovanni; Hennequin, Romain; Epure, Elena
Publication date	2022-12-07
Original citation	Gabbolini, G., Hennequin, R. and Epure, E. (2022) 'Data-Efficient Playlist Captioning With Musical and Linguistic Knowledge', EMNLP 2022, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, UAE, 7-11 Dec., Association for Computational Linguistics, pp. 11401-11415.
Type of publication	Conference item
Link to publisher's version	https://preview.aclanthology.org/emnlp-22-ingestion/2022.emnlp-main.784/ Access to the full text of the published version may require a subscription.
Rights	**© 2022** https://creativecommons.org/licenses/by/4.0/
Item downloaded from	http://hdl.handle.net/10468/14062

Data-Efficient Playlist Captioning With Musical and Linguistic Knowledge

Giovanni Gabbolini
Insight Centre for Data Analytics
School of Computer Science & IT
University College Cork, Ireland
giovanni.gabbolini@insight-centre.org

Romain Hennequin and **Elena Epure**
Deezer Research
Paris, France
research@deezer.com

Abstract

Music streaming services feature billions of playlists created by users, professional editors or algorithms. In this content overload scenario, it is crucial to characterise playlists, so that music can be effectively organised and accessed. Playlist titles and descriptions are proposed in natural language either manually by music editors and users or automatically from pre-defined templates. However, the former is time-consuming while the latter is limited by the vocabulary and covered music themes. In this work, we propose PLAYNTELL, a data-efficient multi-modal encoder-decoder model for automatic playlist captioning. Compared to existing music captioning algorithms, PLAYNTELL leverages also linguistic and musical knowledge to generate correct and thematic captions. We benchmark PLAYNTELL on a new editorial playlists dataset collected from two major music streaming services. PLAYNTELL yields 2x-3x higher *BLEU@4* and *CIDEr* than state of the art captioning algorithms.

1 Introduction

Playlists are a popular feature of music streaming services. Users consume playlists for the 31% of their listening time (Schedl et al., 2018). And, 55% of users create their own playlists (Muligan, 2017). Playlists are also created for users by professional editors or algorithms. For instance, the popular music streaming service, Spotify, was hosting more than four billion playlists in 2021(Dean, 2021).

In this content overload scenario, it is crucial to characterise playlists, so that music can be effectively organised and accessed (Choi et al., 2016). A common approach is to rely on automatic playlist tagging with tags such as music genres or decades. However, tagging solutions are limited because they often resort to a pre-defined set and fall short of filling the semantic gap between audio and human-like descriptions (Choi et al., 2020). In contrast, natural language could be used to characterise playlists with less ambiguity and using a richer vocabulary. In fact, curators often provide a title and/or a description of their created playlist in natural language. However, describing playlists, especially the ones created by recommendation algorithms, is very labor-intensive and time-consuming when done manually (Doh et al., 2021). When automatic captions are proposed, these often rely on pre-defined templates thus cannot cover all kinds of cases, similar to tags (Afchar et al., 2022).

To address the above limitations, Choi et al. (2016) introduce the task of playlist captioning, that is automatically describing a playlist using natural language. Playlist captioning can enable several useful applications, such as assisting curators in the process of finding an appropriate caption for a playlist; enabling search and discovery of playlists through human-like queries (Manco et al., 2021); assigning captions to algorithm-generated playlists, that could be also used as explanations for automatic playlist recommendations (Afchar et al., 2022). Even so, playlist captioning is still an under-researched topic. As of today, we are aware of only two contributions in the field: Choi et al. (2016) and Doh et al. (2021). Although promising, these are afflicted by two main limitations.

The first limitation is poor data quality. Public datasets for playlist captioning rely on playlists created by users for personal use (Doh et al., 2021; Zamani et al., 2019). Instead, editorial playlists are created by professional editors for a public audience. Cunningham et al. (2006) find that user playlists may not have a strictly defined theme, while the editorial ones usually do. Therefore, user playlists may not be an optimal source to learn *representative* captions, especially considering that the theme is regarded as a common playlist descriptor (Kamehkhosh et al., 2020).

The second limitation is the semantic gap. Algorithms for playlist captioning consider as input embeddings of tracks, and strive to generate a correct caption as output. For example, Choi et al. (2016) represent tracks as audio embeddings; and Doh et al. (2021) as random embeddings indexed by track id and updated by back-propagation. In both cases, the low-level input embeddings and the high-level output caption are separated by a "semantic gap" (Celma Herrada et al., 2006) that algorithms are tasked to close. Closing the semantic gap is challenging, especially because playists may be built around themes not easily deducible from embeddings alone. For example, an Ireland and a UK thematic playlist could be confused given the cultural similarity these countries share. Artist thematic playlists, e.g. "100% The Beatles", are even more difficult to caption due to data sparsity. In the whole dataset, an artist can be absent or be mentioned only in some playlists, making it difficult for existing models to learn tracks-to-artists mappings that could be exploited to produce relevant captions (Shimorina and Gardent, 2018).

In this work, we propose PLAYNTELL, a new multi-modal, data-efficient (Adadi, 2021) playlist captioning model that overcomes the above limitations by leveraging linguistic and musical knowledge, to generate English-language thematic captions. First, PLAYNTELL narrows the semantic gap with musical knowledge. In particular, it leverages tags, e.g. "Ireland", "rock" and "90s", that provide information at the same high semantic level as the expected captions. We also introduce an ad-hoc strategy to deal with artist thematic playlists, by masking artist mentions, and informing PLAYNTELL with an artist distribution vector at encoding.

Second, we train PLAYNTELL on a new high-quality dataset of editorial playlists assembled from two major music streaming services, which we release together with the code[1]. However, as editorial playlists have the drawback of sparsity, *i.e.* our dataset is composed by only few thousand samples, training PLAYNTELL from scratch to generate correct natural language captions is challenging (Wang et al., 2019). Inspired by existing work in computer vision (Chen et al., 2021), we address this limitation by warm-starting the decoder with a pre-trained GPT-2 (Radford et al., 2019).

We validate PLAYNTELL with extensive quantitative experiments: PLAYNTELL outperforms existing playlist captioning algorithms, achieving 2x higher *BLEU@4* and 3x higher *CIDEr*. Also, we observe via qualitative evaluation that it can generate relevant editorial-like captions. We report in Table 1 a caption generated by PLAYNTELL and its ground truth. More examples can be found in Table 5. In summary, our contributions are:

	Caption
PLAYNTELL	"80s smash hits the best tracks of the decade"
Ground truth	"all out 80s the biggest songs of the 1980s"

Table 1: Example of output generated by PLAYNTELL vs. the corresponding ground truth.

1. PLAYNTELL, a new multi-modal, data-efficient playlist captioning encoder-decoder model that leverages audio, linguistic and musical knowledge to generate thematic captions;
2. A new high-quality dataset of thematic playlist created by editors from two major music streaming services. We enrich each playlist with tags automatically collected at track level from the crowdsourced database Discogs[2];
3. An extensive evaluation of PLAYNTELL reporting 2x-3x higher *BLEU@4* and *CIDEr* than existing playlist captioning algorithms. We also provide a qualitative analyses of PLAYNTELL, as well as an ablation study, sensitivity analysis to validate the contribution of different modalities and model components, and a user study.

2 Related work

Automated captioning is an active research field that has attracted much attention in the recent years. We can find attempts to caption images (Stefanini et al., 2021), videos (Gao et al., 2017), audio (Drossos et al., 2020) and music (Manco et al., 2021). Here we review works in audio and music captioning, which are closest to our contribution.

Audio captioning focuses on identifying the human-perceived information in a general audio signal and expressing it through text, using natural language (Drossos et al., 2020). For example, an audio caption is: "a door creaks as it slowly revolves back and forth". Koizumi et al. (2020) propose a transformer architecture for audio captioning that is similar to the original transformer for machine translation (Vaswani et al., 2017). The input audio is embedded with a pre-trained Convolutional Neural Network (CNN) (Hershey et al., 2017). The embeddings are input to a transformer

[1] https://github.com/deezer/playntell

[2] https://data.discogs.com/

encoder. Then, a transformer decoder is tasked to generate the target caption.

While audio captioning is concerned with general audio signals, music captioning deals with music audio signals. Manco et al. (2021) propose a Recurrent Neural Network (RNN) encoder-decoder to tackle single song captioning. The multi-modal encoder takes as input the song audio embedding and the caption embedding up to token t. The song embedding is obtained with a pre-trained CNN (Pons et al., 2017). The caption embedding is obtained with a pre-trained word2vec (w2v)-like model. The decoder is tasked to generate the token $t + 1$.

Choi et al. (2016) propose a RNN encoder-decoder to tackle playlist captioning. However, their attempt is not successful, mainly due to overfitting on a small training set. Doh et al. (2021) frames playlist captioning as a machine translation task. The source playlist is treated as a sequence of song ids. The target caption is treated as a sequence of token ids. The authors apply the seq2seq (Bahdanau et al., 2015) and transformer (Vaswani et al., 2017) machine translation models to translate a playlist to a caption. They train the models on user playlists from the Million Playlist Dataset (Zamani et al., 2019).

As detailed in Section 1, current work on playlist captioning suffer from two main limitations: 1) poor data quality of existing datasets; 2) the semantic gap between the low-level input playlist embeddings and the high-level output caption. Here, we tackle the two limitations by designing a model that takes advantage of high-level musical and linguistic knowledge, apart from audio, and by training it on a newly collected dataset of high-quality editorial playlists, better suited to capture themes.

3 Dataset

Public playlist captioning datasets are limited to user playlists, known to be noisier and sometimes not focused on a theme (Doh et al., 2021; Cunningham et al., 2006). We address the above limitation by introducing a new dataset of thematic editorial playlists. We collected public editorial playlists from Spotify and Deezer, which are two major music streaming services (Muligan, 2017).

Each playlist consists of a sequence of track ids, a title and a description. Both tracks, title and description are curated by a professional editor. Each track is associated to at least one artist.

Some playlist are extremely short, e.g. just two tracks. We get rid of these outliers by filtering out playlists which number of tracks is below the fifth percentile. Some other playlists are extremely long, e.g. more than 200 tracks, or have long captions, e.g. more than 50 words. In practice, such outliers have the effect of increasing, respectively, the memory requirements and inference time of algorithms. For these reasons, we filter out playlists which number of tracks is above the 95th percentile, or which caption length in words is above the 95th percentile. We end up with 5467 Deezer playlists and 1104 Spotify playlists. We observe high data quality, so no further pre-processing is needed. We release both raw and pre-processed datasets with the code. We present dataset statistics in Appendix A.

We consider both title and description to be part of a caption. In particular, a caption is defined by the template: *<title> [the title] <description> [the description]*. An example is: *<title> 100% The Beatles <description> The best music from The Beatles.* Other captions are the Ground truths in Table 5. The choice of the template follows recent advances in few-shot learning, where the samples are enriched by task-specific tokens (Li and Liang, 2021; Zhao et al., 2021). Their results do not extend to our fine-tuning setting, as we experiment with other templates, *e.g. [the title] <sep> [the description]*, with no difference in performance.

The captions we consider account for a variety of playlist themes, such as musical genres, moods, activity, events and artist, and were annotated by a number of professional editors coming from different cultural backgrounds. Also, the dataset statistics we report in Table 7 (in Appendix) provide further evidence of the captions' linguistic diversity. For example, the Deezer playlists are 5467, and their captions contain 20312 unique words.

We partition the Deezer playlists randomly in training, validation and test, accounting respectively for 60%, 20% and 20% of the total. Validation and test splits allow internal evaluation, *i.e.* on in-distribution samples. We use the Spotify playlists as an additional test set for an external evaluation, *i.e.* on out-of-distribution samples.

PLAYNTELL, the model we propose, is designed to bridge the semantic gap between a playlist and the relative caption by leveraging different sources of musical knowledge, among which tracks audio and tags. As for audio, we retrieve 30-seconds audio previews of playlist tracks. Audio previews are convenient because they can be freely accessed

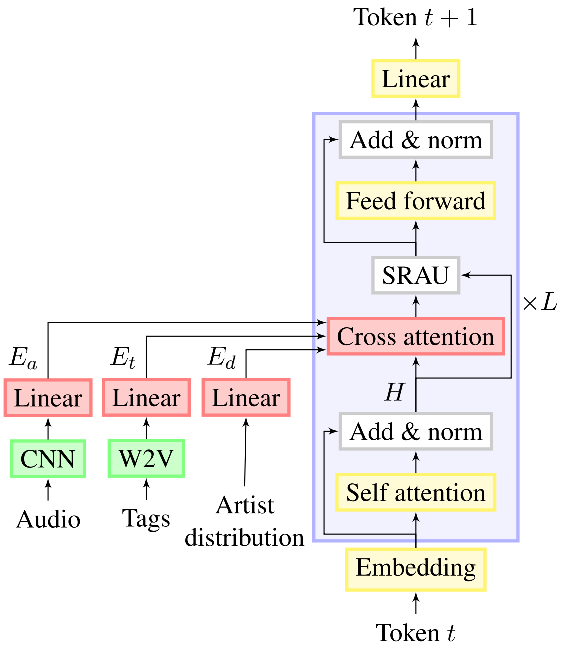


Figure 1: PLAYNTELL - model architecture.

from music streaming public API. As for tags, we resort to the crowdsourced database Discogs, which offers tags at album level. We propagate album tags to every track in the album. The tags cover a range of aspects: genres, countries, years and moods (e.g. "pop", "italy", "2020" and "happy"). We release the tracks tags with the captioned playlists.

Tracks audio and tags differ in availability. While the existence of audio is implied by the existence of a track, the availability of tags depends on the crowdsourcing databases, such as Discogs. Thus, tags may not be available for a new release, just because no crowdsourcer have tagged it yet. The model we propose leverages tags if available, and, if not, can rely on audio only to generate sensible captions, as we show in Section 5.3.

4 Method

Editorial data are available in small quantities, as their existence depends on the expensive work of professional editors. In our case, we can rely on a few thousand editorial playlists, as we detail in Section 3. In this setting, it may be difficult to learn a model from scratch. We introduce PLAYNTELL, a new data-efficient playlist captioning model that leverages linguistic and musical knowledge to generate editorial-like captions. PLAYNTELL is composed of an encoder and decoder, shown in the left and right hand-sides of Figure 1.

4.1 Encoder

The encoder has 3 branches that handle musical knowledge as audio, tags and artist distribution.

Audio Tracks audio is commonly used in playlist tagging (Choi et al., 2020) or playlist captioning (Choi et al., 2016). For every track, we retrieve 30-seconds audio previews as detailed in Section 3 and create an audio embedding by using a pre-trained CNN. The CNN architecture is VGGish (Pons and Serra, 2019). The CNN extracts a 256-dimensional embedding vector for every three seconds of audio. That is, ten embeddings for one track, which we average. For a playlist with P tracks, we then obtain a $P \times 256$ matrix, which we transform into a $P \times h$ matrix using a learned linear layer.

Tags Audio and captions are separated by a wide semantic gap that algorithms may struggle to close. Therefore, we propose to inform PLAYNTELL with tags, e.g. "pop", "happy", which provide information at the same semantic level as captions. For every track in a playlist, we retrieve tags from Discogs, as in Section 3. We consider as playlist tags all the distinct tracks tags. We embed tags with a pre-trained word2vec-like model specific for music-related text, called music-w2v (Doh et al., 2020). Music-w2v embeddings are 300-dimensional. We embed all playlist tags to obtain a 300-dimensional matrix, which we transform to a h-dimensional matrix using a learned linear layer.

Artist distribution Some playlists are artist thematic, *e.g.* "100% The Beatles". Kamalzadeh et al. (2012); Cunningham et al. (2006) report that artists are a common organisation principle among playlist-makers. While audio and tags inform the model with "general" musical knowledge, *e.g.* genres and styles, they may not help to detect artist thematic playlists, which is challenging even with large scale datasets (Royo-Letelier et al., 2018). As a remedy, we introduce the artist distribution vector. It has as i^{th} value the share of tracks in the playlist authored by the i^{th} most popular artist in the playlist. For example, if a playlist has one track by John Lennon and 99 tracks by The Beatles, the vector is $[0.99, 0.01]$. In tracks authored by multiple artists, we consider only the main artist for simplicity. We limit the length of the vector to ten and pad it when necessary. We experimented with higher vector length, without any performance gain. We project the vector to a h dimensional space using a learned linear layer.

The output of each encoder branch is 0-padded to a common number of rows N, to obtain the three embedding matrices E_a, E_t, $E_d \in \mathbb{R}^{N \times h}$.

CNN and w2v embeddings are frozen *i.e.* not updated during training. This is similar to the state of the art in image captioning, where the input image is embedded by means of a pre-trained and frozen CNN, before being fed to a transformer-like model tasked to generate the output caption (Cornia et al., 2020; Herdade et al., 2019; Guo et al., 2020).

4.2 Decoder

The decoder is very similar to a transformer decoder (Vaswani et al., 2017). An attention function is at its core. Given matrices $Q \in \mathbb{R}^{n_q \times d}$, $K \in \mathbb{R}^{n_k \times d}$ and $V \in \mathbb{R}^{n_k \times d}$, representing query, key and value, the attention is defined as:

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V. \quad (1)$$

Attention computes a weighted sum of V rows according to the similarity between Q and K rows. In practice we implement the multi-head variant of attention (Vaswani et al., 2017).

The decoder is composed of three parts: input, hidden and output. In the input part, we use learned token embeddings to convert a caption token to an embedding of dimension h. In the output part, we use a learned linear layer with softmax to produce predicted next token probabilities. In practice, we provide as input all captions tokens shifted right, and we predict all next tokens in parallel. The hidden part is made up of L identical layers. Every layer is composed of three sub-layers: self-attention, cross-attention, and feed-forward.

Self-attention & feed forward We apply the attention function using the same matrix as query, key and value in the self-attention layer. The attention function is modified so to avoid the prediction of token t to depend on subsequent tokens. The feed-forward layer consists of a fully connected neural network (Vaswani et al., 2017). We wrap each layer around a residual connection (He et al., 2016) and a normalisation layer (Ba et al., 2016).

Cross-attention We apply the attention function using the encoder output as key and value, and the self-attention output H as query. Our encoder has three outputs. We apply the attention function to every output separately and then sum the results, similar to Zhao et al. (2019). Then:

$$\begin{aligned} Crs\text{-}att(H, E_a, E_t, E_d) = {} & Att(H, E_a, E_a) \\ & + Att(H, E_t, E_t) \quad (2) \\ & + Att(H, E_d, E_d). \end{aligned}$$

We wrap the cross-attention layer around a SRAU layer, proven effective in data-efficient image captioning (Chen et al., 2021).

We inject linguistic knowledge in the decoder, similar to Chen et al. (2021). GPT-2 (Radford et al., 2019) has a transformer-like architecture, compatible with PLAYNTELL. We load pre-trained GPT-2 (small) weights in the layers of the decoder: embedding, self-attention, feed-forward and linear. These layers are fine-tuned during training. The choice of GPT-2 follows previous work (Chen et al., 2021). Other pre-trained decoders, such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020), could be used. The choice of GPT-2 fixes the decoder hyper-parameters to $N = 12$ transformer layers, 12 attention heads and hidden size $h = 768$. As a result, the three linear layers in the decoder are tasked to project from respectively 256, 300 and 10 dimensional spaces to a 768 dimensional space.

4.3 Artists masking

Correctly generating artists mentions is important, as artist-thematic is a common category of editorial playlists (Kamalzadeh et al., 2012; Cunningham et al., 2006). However, correctly generating artists mentions is a particularly challenging task, mainly due to data sparsity. In the whole dataset, an artist can be absent or be mentioned only in some playlists, making it difficult for algorithms to learn a mapping between input data and artist mentions.

We introduce artist masking as a remedy to data sparsity, similar to Zhao et al. (2019). We pre-process the training captions by substituting artist mentions with placeholders. For example, the caption “100% The Beatles” is pre-processed as “100% artist1”. If a caption mentions more than one artist, we will have more than one placeholder (“artist1”, “artist2”, ...). The mapping between placeholders and artists is decided in advance. We use popularity within the playlist: the author of most tracks in the playlist has placeholder “artist1”; the second most popular artist has placeholder “artist2”; and so on. We post-process the output captions by substituting placeholders with actual artist mentions. The artist masking strategy we present is designed to be used in conjunction with the artist distribution

	Deezer								Spotify							
	B@1	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*	*B@1*	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*
NN	18.7	6.2	3.4	2.4	16.6	17.4	11.4	81.2	15.8	3.5	1.2	0.5	17.4	17.1	4.2	81.5
MusCaps	18.7	8.9	4.9	2.6	17.8	23.8	12.2	77.2	19.0	6.3	2.3	0.9	17.4	19.6	6.8	75.8
DohRNN	19.6	8.5	4.7	2.5	16.7	21.1	7.0	80.0	18.5	5.0	1.7	0.5	17.3	18.5	2.6	80.5
DohTra	20.6	8.8	5.1	3.2	17.2	20.5	12.0	81.1	18.3	4.4	1.4	0.4	17.4	18.0	2.4	81.1
PlayNTell	**29.0**	**18.9**	**14.4**	**11.8**	**22.9**	**32.7**	**114.9**	**84.4**	**23.4**	**9.0**	**4.3**	**2.3**	**19.8**	**22.9**	**26.3**	**82.8**

Table 2: State of the art accuracy as measured on the Deezer and Spotify test sets. Bold indicates the most accurate algorithm. PlayNTell score 2x-3x higher *BLEU@4* and *CIDEr* than the baselines.

	Deezer		Spotify	
	% novel	*Vocab*	*% novel*	*Vocab*
NN	0.0	1750	0.0	2100
MusCaps	100.0	66	100.0	60
DohRNN	100.0	838	100.0	764
DohTra	100.0	2015	100.0	1767
PlayNTell	97.6	2585	98.3	2147

Table 3: State of the art diversity as measured on the Deezer and Spotify test sets.

vector, which provides useful knowledge on how to generate artist placeholders.

5 Experiments

5.1 Experimental setting

Metrics We adopt eight accuracy metrics: *BLEU@1* to *4* (*B@1* to *4*), *METEOR* (*M*), *ROUGE-L* (*R-L*), *CIDEr* (*C*) and *BERT-Score* (*B-S*) (Celikyilmaz et al., 2020) . The first seven are a function of n-grams precision and recall of the ground truth with respect to the generated caption. *BERT-Score* exploits pre-trained BERT embeddings to represent and match the tokens in the ground truth with respect to the generated caption. We use *BERT-Score* with recall and *idf* weighting, which is the suggested configuration (Zhang et al., 2020).

We also adopt two diversity metrics. *% novel* is the percentage of generated captions that are not among the training captions. *Vocab* is the number of unique words used in all the generated captions.

Following Dror et al. (2018), we set up a t-test for *ROUGE-L*, *CIDEr* and *BERT-Score*, and a paired bootstrap test for *BLEU@1* to *4* and *METEOR*. Following Koehn (2004), we fix the number of bootstrap replicas to 1000.

Implementation details We convert a caption to tokens using Byte Pair Encoding (BPE) (Sennrich et al., 2016). We use learned positional embeddings, so to distinguish the order of tokens.

We optimise the model using simple cross-entropy loss, computed on the generated caption against the ground truth. During inference, the prediction of the previous time step is fed to the input of the next time step. We use a beam search of size three to compute the most likely output sequence.

We train the models with the AdamW optimizer (Loshchilov and Hutter, 2018). We use a learning rate equal to 10^{-4} and a batch size equal to 10. We use early stopping with patience equal to 40. We set threshold τ of the SRAU gate to 0.2, as recommended in (Chen et al., 2021).

5.2 Comparison with state of the art

Baselines We compare PlayNTell with a simple NN (Nearest Neighbor) baseline and state of the art music captioning algorithms:

NN Given a test playlist p, find the training playlist $\tilde{p}$ closest (wrt. cosine distance) to the test playlist, and output p's caption equal to $\tilde{p}$'s caption. A playlist is represented as the average audio embedding of its tracks, as previews work show no benefit on leveraging the sequential nature of playlists (Choi et al., 2020). Tracks audio embeddings are computed as in Section 4.1.

MusCaps We adopt the model proposed in Manco et al. (2021) to caption playlists. We replace song audio embedding with playlist audio embedding. A playlist audio embedding is the average audio embeddings of the songs in the playlist. We use the authors' implementation with default parameters;

DohRNN and **DohTra** seq2seq and transformer models for playlist captioning proposed in Doh et al. (2021). We use the authors implementation with default parameters;

We use as training set the Deezer dataset training split. We use as test sets the Spotify dataset and the Deezer dataset test split. We assess the algorithms according to accuracy and diversity metrics.

The results on accuracy are reported in Table 2. PlayNTell largely outperforms all baselines, achieving 2x higher *BLEU@4* and 3x higher *CIDEr*. The huge improvement on *CIDEr* may be due to the artist mentions, as *CIDEr* is particularly sensitive to un-frequent words (Vedantam et al., 2015). While the baselines may struggle, PLAYNTELL takes advantage of the musical knowledge in terms of artist distribution to correctly generate artist mentions. We can notice smaller differences of *BERT-Score*. This is expected, as *BERT-Score* is known to assume values in a narrow range (Zhang et al., 2020). We test the significance of differences in accuracy, as explained in Section 5.1. The differences are statistically significant ($p < 10^{-4}$).

The Spotify dataset is used for external validation. We observe quite high values of the metrics, but lower overall. As expected, this is due to differences in "style" between the two platforms. Artist thematic playlists in Spotify are captioned as *e.g.* "This is The Beatles"; in Deezer, we would have "100% The Beatles". Similarly, the most common words in Spotify captions are: "cover", "from", "tracks"; in Deezer are: "by", "best", "music".

The results on diversity are reported in Table 3. NN scores 0 in *% novel*, as NN can only generate captions of the training set. Instead, music captioning baselines only generate novel captions, as they score 100 in *% novel*. This may seem surprising, as captioning algorithms are known to generate a share of novel captions, and replicate a share of training captions (Stefanini et al., 2021). However, we believe that the above statistic is due to problems music captioning baselines have for correct text generation. For example, a caption generated by MUSCAPS is: *<title> 100% jazz <description> the best of the best of the best music*. We provide more evidence of such problems in Section 5.5. Baselines score modest *Vocab*s. For example, MUSCAPS's *Vocab* in the Deezer dataset is only 2% of PLAYNTELL's *Vocab*. MUSCAPS's *Vocab* may be low because the hyper-parameters are not optimised for the dataset. Instead, PLAYNTELL can replicate a share of training captions *i.e.* *% novel* < 100, and has the largest *Vocab*.

Finally, we assess the algorithms considering titles and descriptions separately. That is, we isolate title and description from the generated caption and ground truth, and we compute the metrics. The results, reported in the Appendix, corroborate the results presented in this Section.

5.3 Ablation study

PLAYNTELL is informed by three sources of musical knowledge: audio, tags, and artist distribution. We consider two variations to the architecture of PLAYNTELL, so to check if the three sources are actually exploited. The first variation has only the audio branch; the second has both audio and artist distribution branches. We compare the two variations with the original PLAYNTELL, which features also the tags branch. In the first variation we do not use the artist masking strategy, while in the second variation and PLAYNTELL we do.

We measure accuracy with the same training-evaluation setup as Section 5.2. The results are in Table 4. The accuracy increases with the number of input modalities, on both datasets. The differences are statistically significant in the Deezer dataset and in the Spotify dataset ($p < 0.05$). Then, we have a good indication that PLAYNTELL can successfully leverage all three sources of musical knowledge.

5.4 Sensitivity analysis

We investigate the impact of the GPT-2 initialisation on the generated captions. We consider a variation of PLAYNTELL with random decoder initialisation, which we name RAND INIT. We measure accuracy with the same training-evaluation setup as Section 5.2. The results are in Table 4.

PLAYNTELL largely outperforms RAND INIT, achieving 63% higher *BLEU@4* and 71% higher *CIDEr*. The differences are statistically significant on both Deezer and Spotify datasets ($p < 10^{-4}$). We observe that PLAYNTELL produces syntactically correct captions, while RAND INIT does not, *e.g.* for one playlist PLAYNTELL generates the caption *<title> relaxing piano <description> relax with calm classical tunes*, while RAND INIT generates the caption: *<title> relaxing music < piano>description <> the of music the playlist classicalind*. Thus, we have a clear indication that the GPT-2 initialisation has a positive impact on the generated captions, and that the linguistic knowledge acquired by GPT-2 is leveraged by PLAYNTELL in the playlist captioning task.

5.5 Discussion

The baselines differ from PLAYNTELL along three axes: input data, linguistic knowledge and architecture. The baselines leverage only tracks audio or learned embeddings. PLAYNTELL accommodates three input modalities, being informed by tracks audio, tags and artists distribution. Comparing the results in Table 4 with the baselines in Table 2, we notice that PLAYNTELL, when informed by audio only, is still consistently superior to the baselines.

	Deezer								Spotify							
	B@1	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*	*B@1*	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*
AUDIO	23.2	11.9	7.6	5.2	19.5	25.7	27.8	82.4	20.8	6.2	2.5	1.1	18.7	20.9	12.9	82.0
+ ARTIST	27.3	17.4	13.2	10.8	22.1	30.7	105.7	84.0	22.9	8.6	4.0	2.0	19.5	22.2	24.1	82.7
RAND INIT	21.0	12.0	8.8	7.0	18.5	24.4	72.6	82.0	17.9	5.4	2.4	1.2	17.2	17.9	16.3	81.0
PLAYNTELL	**29.0**	**18.9**	**14.4**	**11.8**	**22.9**	**32.7**	**114.9**	**84.4**	**23.4**	**9.0**	**4.3**	**2.3**	**19.8**	**22.9**	**26.3**	**82.8**

Table 4: Accuracy with one and two input modalities and with random initialisation compared to PLAYNTELL (3 modalities and initialized with GPT-2) on the Deezer and Spotify test sets. Bold indicates highest accuracy.

	Caption
NN	*<title> best of calm piano 2020 <description> the best of the neo classical scene*
MUSCAPS	*<title> 100 % jazz <description> the best of the best of the best music*
DOHRNN	*<title> boogie # 2016 <description> all the best songs from the best recordings*
DOHTRA	*<title> rock 101 <description> the best of the decade in this playlist # pulse # pulse*
PLAYNTELL	*<title> relaxing piano <description> relax with calm classical tunes*
Ground truth	*<title> peaceful piano <description> relax and indulge with beautiful piano pieces*
NN	*<title> top trending hip hop <description> a collection of hip hop hits and viral trends that 's updated weekly*
MUSCAPS	*<title> 100 % rock <description> the best of the best of the best of the best of the best of the best rock*
DOHRNN	*<title> chill fi <description> <description> the hottest hottest tracks to the hottest focus %*
DOHTRA	*<title> an introduction to the best of <description> the best british music that have made the great playlists %*
PLAYNTELL	*<title> 100 % ramones <description> enjoy the kings of rock this is ramones collection*
Ground truth	*<title> this is ramones <description> this is ramones the essential tracks all in one playlist*
NN	*<title> trap & bass by spinnin records <description> trap & bass by the finest 808 suppliers*
MUSCAPS	*<title> top hits hits <description> the best tracks of the best tracks*
DOHRNN	*<title> tiktok party <description> <description> the playlist <description> the the the most party hits*
DOHTRA	*<title> 100 % queen latifah <description> all the best songs of the amazing lebanese tracks*
PLAYNTELL	*<title> rising african hits <description> the best african tracks of our favorite african pop music in one playlist*
Ground truth	*<title> desi hits <description> desi hits from south asia cover badshah*

Table 5: Ground truth and generated captions on the Spotify dataset.

Similarly, compared to the baselines, PLAYNTELL leverages linguistic knowledge held by GPT-2 weights. However, when comparing the performance of PLAYNTELL without GPT-2 weights (RAND INIT in Table 4) with the baselines in Table 2, we notice that RAND INIT outperforms the baselines with in-distribution data (Deezer dataset). With out-of-distribution data (Spotify dataset), RAND INIT outperforms all the baselines for some metrics (e.g. *BLEU@1*, *CIDEr*), but only some baselines for some other metrics (e.g. *BLEU@1*). Thus, there is evidence that our novel encoder, which is the main architectural difference between PLAYNTELL and the baselines, is enough to outperform the baselines, at least with in-distribution data. With out-of-distribution data, the contribution of the encoder may be blurred by overfitting in the small in-distribution dataset, as PLAYNTELL has a far more complex decoder than the baselines. Hence, we expect that, when trained on a larger in-distribution dataset, PLAYNTELL can outperform the baselines in both in and out-of-distribution data, without using GPT-2 weights.

We report captions generated by the algorithms on the Spotify test set when trained on the Deezer training set and the corresponding ground truths in Table 5. Captions produced by music captioning baselines are not syntactically correct, probably because the dataset is too small to learn a sound language model. On the other hand, PLAYNTELL can take advantage of the pre-trained GPT-2 weights and generate syntactically correct captions.

Captions produced by the baselines are not always semantically correct. In the first example, none of the baselines matches the playlist theme. In the second example, MUSCAPS aligns with the playlist theme, while DOHRNN and DOHTRA do not. MUSCAPS holds pre-trained audio knowledge that may help in latter. Conversely, PLAYNTELL takes advantage of musical knowledge through audio, tags, and artist distribution, and matches the theme in both cases. The third example features a playlist of “desi”[3] (traditional south-asian music). None of the algorithms matches this latter theme, probably beacause the input data do not properly represent the concept of “desi” music. For example, the “desi” tag is not among the playlist tags we use as input to PLAYNTELL. This example witnesses the western-centered perspective of Music Information Retrieval research, which is a debated topic (Huang et al., 2021).

[3] https://en.wikipedia.org/wiki/Desi

The NN baseline only outputs captions from the training set. So, by design, NN produces syntactically correct captions. NN is aligned with the playlist theme in one case over three. Although simple, NN appears as a competitive baseline.

Supplementary examples are in the Appendix.

5.6 User study

The user study is based on a survey in which the participants were asked to rate playlists captions on three different aspects: (1) *Content match*: Do the title and description match the playlist content? (2) *English correctness*: Are the title and description correct in English? (3) *Appeal*: Are the title and description appealing? The participant could respond to each statement with five options on a Likert scale. We used as inspiration for these questions user trial designs in image captioning, where (1) and (2) were often assessed (Zhao et al., 2019).

During the study, the participants were shown the ten first songs of curated playlists. The songs were shown as a playlist page on the Deezer website thus including the song title, the artist name and the album title (see appendix E). Those playlists were sampled randomly from the Deezer test set. Each playlist was presented with several captions (title and description). The captions were either the ground truth caption (original caption assigned to the playlist by a human editor) or generated by one of the methods evaluated in this paper: the baselines MUSCAPS, DOHTRA, NN or PLAYNTELL. 7 different playlists were presented to each participants (with no overlapping between participants). The comparison of PLAYNTELL with the ground truth and baselines serves to assess how far is our algorithm from, respectively, human editors, considered as the golden standard, and the state of the art in playlist captioning. In order to guarantee that the participants had sufficient musical knowledge to accurately assess a playlist caption, the 7 participants were exclusively recruited among Deezer playlist editors.

Results are shown in Table 6. The first thing to be noted is that the ground truth captions get scores that are quite far from the perfect score of 5. One reason that could explain this phenomenon is that playlist captioning is intrinsically subjective. Then, the NN baseline has the best results in terms of *appeal* and *English correctness* with metrics close to the ground truth. This is not surprising as NN is the only baseline that does not generate captions but outputs captions attributed by humans to other playlists. However, the NN baseline is not able to output captions that match the content of the associated playlist as shown by the *content match* metrics. On the contrary, PLAYNTELL outperforms all baselines in terms of *content match* with a score that is significantly higher than all other baselines but not significantly lower than the ground truth. Also, PLAYNTELL is the second captioning method in terms of *appeal* and *English correctness* after NN with no significant difference for *appeal*. PLAYNTELL brings a significant improvement in terms of *appeal* compared to other generative models, and over MUSCAPS in terms of *English correctness*.

Method	Mean	Std.	p-value
Content match			
PLAYNTELL	3.20408	1.27442	Ref
NN	1.63265	0.83401	<1e-3
MUSCAPS	2.04082	0.99915	<1e-3
DOHTRA	1.55102	0.86750	<1e-3
GroundTruth	3.665306	1.199844	0.066
English correctness			
PLAYNTELL	3.14286	1.32288	Ref
NN	3.67347	0.92168	0.019
MUSCAPS	2.34694	1.09070	0.002
DOHTRA	2.73469	1.15064	0.094
Ground Truth	3.83673	1.23063	0.004
Appeal			
PLAYNTELL	2.87755	1.14805	Ref
NN	3.22449	1.00551	0.071
MUSCAPS	1.97959	0.87773	<1e-3
DOHTRA	2.30612	0.96186	0.005
Ground Truth	3.44898	1.19131	0.013

Table 6: Means and standard deviations of Likert scores (in $[1, 5]$) and p-values obtained from a paired t-test between the considered method and PLAYNTELL scores.

These results tend to confirm that PLAYNTELL is able to generate realistic captions matching the content of the playlists with a significant improvement over the state-of-the-art.

6 Conclusion and future work

With the contribution at hand, we add to the literature on playlist captioning, as we present a new dataset and a new model that sets a new state-of-the-art. PLAYNTELL leverages general linguistic knowledge from a pre-trained language model to generate coherent captions, and musical knowledge to make the captions consistent with the playlist content. Planned future work include the extension to non-English captions, and to apply PLAYNTELL in the field of music recommender systems, where captions could explain automatically generated playlists (Afchar et al., 2022).

7 Limitations

We identify multiple limitations of our work. First, our work is biased towards western music. The dataset we employ features playlists of mainly western music. As such, PLAYNTELL can find the right caption for a playlist of e.g. UK pop-rock made in the early '00s, but it struggles for a playlist of e.g. traditional south-asian music, or "desi" music. We provide evidence of this cultural bias in Section 5.

A second type of bias is with regard to the language of the produced output which targets only English. Moreover, the captions are written in a manner aligned with editorial guidelines of the Deezer music streaming. We attempted to mitigate this bias by testing PLAYNTELL on an external dataset from Spotify. Nonetheless, while we show in the experiments that the two corpora employ a diverse, partially non-overlapping vocabulary, we expect editorial styles to be somewhat aligned across these two music streaming platforms.

These limitations can be overcome by ensuring that playlists from different parts of the world are equally represented. Though possible, finding such datasets can be challenging given the general western-centered perspective of the whole field of Music Information Retrieval (Huang et al., 2021).

Another limitation is the offline evaluation. We set-up a standard experimental procedure, where the quality of algorithms is determined by how well they can replicate test captions unseen at training time, as measured by several metrics, such as *BLEU*. Though widely accepted, the above experimental procedure neglects the user perspective.

To address this, we set up a user trial with music editors in order to take into account the end user perspective. While we ensured that the editors involved in the study had various music backgrounds, the fact that they came only from one organisation is a limitation. Also the number of participants was small (7 editors) and the data they assessed came only from Deezer playlists. This latter aspect entails a familiarity with the Deezer captioning style, which can bias the evaluation, especially of the ground-truth captions. However, although more extensive user trials need to be set up, this preliminary study gives already interesting insights that corroborate with the results of the quantitative evaluation.

References

Amina Adadi. 2021. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):1–54.

Darius Afchar, Alessandro B Melchiorre, Markus Schedl, Romain Hennequin, Elena V Epure, and Manuel Moussallam. 2022. Explainability in music recommender systems. *arXiv preprint arXiv:2201.10528*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Òscar Celma Herrada, Herrera Boyer, Xavier Serra, et al. 2006. Bridging the music semantic gap. In *Proceedings of the Workshop on Mastering the Gap, From Information Extraction to Semantic Representation, held in conjunction with the European Semantic Web Conference; 2006 Jun 11-14; Budva, Montenegro.[Aachen]: CEUR Workshop Proceedings; 2006.* CEUR Workshop Proceedings.

Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. 2021. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. *arXiv preprint arXiv:2102.10407*.

Jeong Choi, Anis Khlif, and Elena Epure. 2020. Prediction of user listening contexts for music playlists. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, pages 23–27.

Keunwoo Choi, George Fazekas, Brian McFee, Kyunghyun Cho, and Mark Sandler. 2016. Towards music captioning: Generating music playlist descriptions. *arXiv preprint arXiv:1608.04868*.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.

Sally Jo Cunningham, David Bainbridge, and Annette Falconer. 2006. 'more of an art than a science': Supporting the creation of playlists and mixes. *International Society for Music Information Retrieval Conference, ISMIR*.

Brian Dean. 2021. Spotify user stats.

Seungheon Doh, Jongpil Lee, Tae Hong Park, and Juhan Nam. 2020. Musical word embedding: Bridging the gap between listening contexts and music. In *Machine Learning for Media Discovery Workshop (ML4DL), International Conference on Machine Learning*. The International Conference on Machine Learning (ICML).

Seungheon Doh, Junwon Lee, and Juhan Nam. 2021. Music playlist title generation: A machine-translation approach. In *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MusA)*, pages 27–31, Online. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE.

Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. 2017. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055.

Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. 2020. Normalized and geometry-aware self-attention network for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10327–10336.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Rujing Huang, Bob LT Sturm, and André Holzapfel. 2021. De-centering the west: East asian philosophies and the ethics of applying artificial intelligence to music. *International Society for Music Information Retrieval Conference, ISMIR*.

Mohsen Kamalzadeh, Dominikus Baur, and Torsten Möller. 2012. A survey on music listening and management behaviours. *International Society for Music Information Retrieval Conference, ISMIR*.

Iman Kamehkhosh, Geoffray Bonnin, and Dietmar Jannach. 2020. Effects of recommendations on the playlist creation behavior of users. *User Modeling and User-Adapted Interaction*, 30(2):285–322.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Yuma Koizumi, Ryo Masumura, Kyosuke Nishida, Masahiro Yasuda, and Shoichiro Saito. 2020. A transformer-based audio captioning model with keyword estimation. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1977–1981. ISCA.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. 2021. Muscaps: Generating captions for music audio. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Mark Muligan. 2017. Announcing MIDiA's State Of The Streaming Nation 2 Report. `https://midiaresearch.com/blog/announcing-midias-state-of-the-streaming-nation-2-report`. [Online; accessed 15-March-2022].

Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. 2017. End-to-end learning for music audio tagging at scale. *International Society for Music Information Retrieval Conference, ISMIR*.

Jordi Pons and Xavier Serra. 2019. Musicnn: Pre-trained convolutional neural networks for music audio tagging. *International Society for Music Information Retrieval Conference, ISMIR*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Jimena Royo-Letelier, Romain Hennequin, Viet-Anh Tran, and Manuel Moussallam. 2018. Disambiguating music artists at scale with audio metric learning. *International Society for Music Information Retrieval Conference, ISMIR*.

Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7(2):95–116.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th international conference on natural language generation*, pages 360–370.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. 2021. From show to tell: A survey on image captioning. *arXiv preprint arXiv:2107.06912*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Chenguang Wang, Mu Li, and Alexander J Smola. 2019. Language models with transformers. *arXiv preprint arXiv:1904.09408*.

Hamed Zamani, Markus Schedl, Paul Lamere, and Ching-Wei Chen. 2019. An analysis of approaches taken in the acm recsys challenge 2018 for automatic music playlist continuation. *ACM Trans. Intell. Syst. Technol.*, 10(5).

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sanqiang Zhao, Piyush Sharma, Tomer Levinboim, and Radu Soricut. 2019. Informative image captioning with external sources of information. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6485–6494, Florence, Italy. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Dataset statistics

We further present some statistics of the preprocessed editorial public playlists collected from Deezer and Spotify in Table 7.

	Statistic	Value
Deezer playlists	Playlist number	5467
	Average caption length (words)	21.1
	Average playlist length (tracks)	47.8
	Unique word number	20312
	Unique track number	182638
Spotify playlists	Playlist number	1104
	Average caption length (words)	16.8
	Average playlist length (tracks)	69.1
	Unique word number	5551
	Unique track number	60765

Table 7: Dataset statistics.

B Comparison with state of the art

	Deezer		Spotify	
	% novel	*Vocab*	*% novel*	*Vocab*
NN	0.0	540	0.0	644
MusCaps	85.8	39	90.8	37
DohRNN	89.5	393	89.7	381
DohTra	41.9	900	34.6	785
PlayNTell	72.5	1257	65.9	959

Table 8: State of the art diversity as measured on the Deezer and Spotify test sets in titles only.

	Deezer		Spotify	
	% novel	*Vocab*	*% novel*	*Vocab*
NN	0.0	1608	0.0	1939
MusCaps	98.6	41	99.0	37
DohRNN	100.0	579	100.0	511
DohTra	100.0	1567	100.0	1375
PlayNTell	88.7	2025	93.2	1671

Table 9: State of the art diversity as measured on the Deezer and Spotify test sets in descriptions only.

We assess the algorithms of Section 5.2 considering titles and descriptions separately. That is, we use as training set the Deezer dataset training split; we use as test sets the Spotify dataset and the Deezer dataset test split; we isolate title and description from the generated caption and ground truth, and we compute accuracy and diversity metrics, in both datasets, for title and description separately. The results are reported respectively in Tables 10 and 11. The results corroborate with Table 2, where we considered the full captions, i.e. following the template: *<title> the title <description> the description.*

C Qualitative study

We analyse some captions generated by the algorithms on the Spotify test set when trained on the Deezer training set, as in Section 5.5. We report some additional generated captions and the corresponding ground truths in Table 12, which is to be considered as a continuation of Table 5.

D Computing devices

PlayNTell was trained with Graphics Processing Units (GPU). We used a 32-core Intel Xeon Gold 6134 CPU @ 3.20GHz CPU with 128GB RAM equiped with 4 GTX 1080 GPUs with 11GB RAM each. Each training phase was performed on a single GPU. Full training of a model (for a single hyper-parameter setting) was about 20 hours.

E User study

In Figure 2, an example of a survey form with the instructions, the asked question and the way the playlist was presented to the participant is displayed.

	Deezer								Spotify							
	B@1	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*	*B@1*	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*
NN	4.4	2.7	1.8	1.3	2.1	4.2	12.6	81.2	2.4	1.1	0.0	0.0	1.4	2.8	6.1	81.5
MusCaps	10.8	8.1	2.3	0.0	5.9	13.1	20.0	77.2	5.2	1.0	0.0	0.0	2.2	5.1	9.0	75.8
DohRNN	8.9	7.2	3.3	2.0	4.5	10.1	13.1	80.0	1.7	0.6	0.0	0.0	0.6	1.5	2.7	80.5
DohTra	8.7	6.3	2.6	1.4	4.1	8.7	12.4	81.1	0.8	0.3	0.0	0.0	0.3	0.8	1.4	81.1
PlayNTell	**34.6**	**30.3**	**27.9**	**25.4**	**19.0**	**34.4**	**218.7**	**84.4**	**12.8**	**7.5**	**4.0**	**2.3**	**6.8**	**12.8**	**42.7**	**82.8**

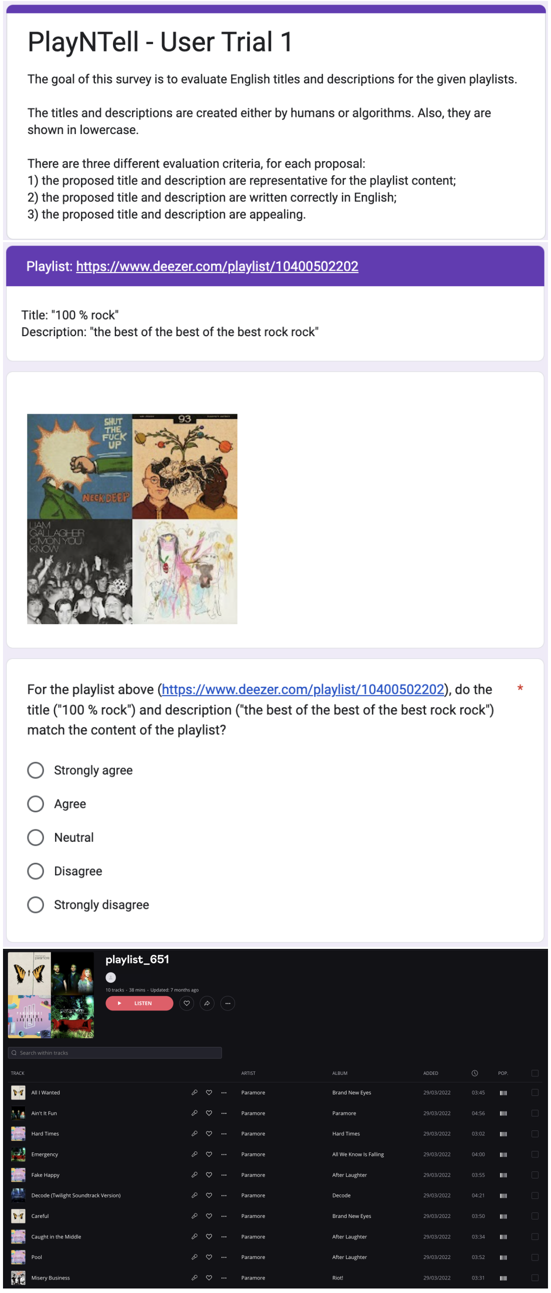
Table 10: State of the art accuracy as measured in the Deezer and Spotify test sets in titles only. Bold indicates the most accurate algorithm.

	Deezer								Spotify							
	B@1	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*	*B@1*	*B@2*	*B@3*	*B@4*	*M*	*R-L*	*C*	*B-S*
NN	10.4	4.3	2.6	1.9	3.9	8.8	11.7	81.2	7.6	2.0	0.7	0.3	3.1	7.4	3.7	81.5
MusCaps	10.5	5.0	2.6	1.5	4.2	14.3	13.9	77.2	9.2	3.6	1.4	0.5	3.1	9.8	5.3	75.8
DohRNN	10.7	4.6	2.4	1.5	3.0	11.3	6.7	80.0	8.4	2.7	1.0	0.3	2.8	8.0	2.5	80.5
DohTra	11.7	5.4	3.2	2.3	3.9	11.4	13.1	81.1	9.0	2.9	1.1	0.0	3.1	8.2	3.1	81.1
PlayNTell	**17.3**	**10.9**	**8.1**	**6.6**	**8.1**	**20.4**	**73.8**	**84.4**	**11.2**	**4.3**	**1.9**	**0.8**	**4.2**	**10.9**	**14.7**	**82.8**

Table 11: State of the art accuracy as measured in the Deezer and Spotify test sets in descriptions only. Bold indicates the most accurate algorithm.

	Caption
NN	*<title> best of dance 2020 <description> top dance tracks of 2020*
MusCaps	*<title> 80s hits <description> the best hits of the best hits*
DohRNN	*<title> best of <description> the best of of the best of the best of the %*
DohTra	*<title> acoustic pop <description> your favorite tracks %*
PlayNTell	*<title> 90s pride <description> the best tracks of the decade to celebrate all colors of love*
Ground truth	*<title> all out 90s <description> the biggest songs of the 1990s*
NN	*<title> mashups | club mix <description> smash hit club & chart remixes sure to get the party started*
MusCaps	*<title> dance hits <description> the best tracks of the best tracks*
DohRNN	*<title> workout workout <description> <description> the the love with the finest and love %*
DohTra	*<title> 100 % the 1975 <description> it 's the essential tracks to keep your little bit %*
PlayNTell	*<title> dance music <description> the biggest dance tracks out there*
Ground truth	*<title> power hour <description> tap it back or go for a spin with these uptempo tracks*
NN	*<title> best of classical 2020 <description> the best albums of 2020 in one playlist what a year*
MusCaps	*<title> 100 % best of <description> the best of the best of the best of the best of the world*
DohRNN	*<title> 100 % feist <description> <description> the essential tracks from one playlist %*
DohTra	*<title> 100 % shania twain <description> the essential tracks in one playlist %*
PlayNTell	*<title> 100 % the beatles <description> the essential tracks in one playlist*
Ground truth	*<title> the long and winding road <description> a journey through the beatles career*
NN	*<title> japanese k pop <description> listen to the japanese rendition of your favorite k pop songs*
MusCaps	*<title> top hits <description> the best hits of the best hits in one playlist*
DohRNN	*<title> lo fi <description> <description> the best to the best and best music %*
DohTra	*<title> 100 % the script <description> need by the best tracks from the legend %*
PlayNTell	*<title> k pop party <description> experience the best k pop music in hifi quality*
Ground truth	*<title> sing along k pop <description> fancy belting out your favourite korean songs*
NN	*<title> 100 % tinie <description> simply unstoppable this is the essential tinie collection*
MusCaps	*<title> 100 % the best of <description> the best of the best tracks from one playlist*
DohRNN	*<title> the house <description> the best of the best dance music in the %*
DohTra	*<title> 100 % the script <description> this is the essential lg collection %*
PlayNTell	*<title> indie essentials <description> high energy indie tracks to keep you motivated*
Ground truth	*<title> essential indie <description> all your indie favorites cover mac demarco*
NN	*<title> late night chill <description> get ready for the after party*
MusCaps	*<title> chill chill <description> the best of chill music <description> the best music*
DohRNN	*<title> chill hits <description> <description> the best hits hits for the latest hits %*
DohTra	*<title> 100 % testament <description> the essential tracks in one playlist %*
PlayNTell	*<title> chill beats <description> the best in chill beats to chill*
Ground truth	*<title> downtempo beats <description> let 's slow down*
NN	*<title> dance pop <description> the biggest edm pop crossover tracks out there*
MusCaps	*<title> dance hits <description> the best tracks of the best tracks*
DohRNN	*<title> classical piano <description> <description> working <description> enjoy the best of 2020 %*
DohTra	*<title> jazz pop <description> a selection of pop songs by the most relaxing moment %*
PlayNTell	*<title> dance pop <description> get ready for the party with this playlist to keep you moving*
Ground truth	*<title> cardio <description> upbeat dance pop to keep your heart pumping*

Table 12: Ground truth and generated captions on the Spotify dataset.



PlayNTell - User Trial 1

The goal of this survey is to evaluate English titles and descriptions for the given playlists.

The titles and descriptions are created either by humans or algorithms. Also, they are shown in lowercase.

There are three different evaluation criteria, for each proposal:
1) the proposed title and description are representative for the playlist content;
2) the proposed title and description are written correctly in English;
3) the proposed title and description are appealing.

Playlist: https://www.deezer.com/playlist/10400502202

Title: "100 % rock"
Description: "the best of the best of the best rock rock"

For the playlist above (https://www.deezer.com/playlist/10400502202), do the title ("100 % rock") and description ("the best of the best of the best rock rock") match the content of the playlist? *

○ Strongly agree

○ Agree

○ Neutral

○ Disagree

○ Strongly disagree

playlist_651

10 tracks - 38 mins - Updated: 7 months ago

TRACK	ARTIST	ALBUM	ADDED	🕓	POP.	□
All I Wanted	Paramore	Brand New Eyes	29/03/2022	03:45	▮▮▮▮	□
Ain't It Fun	Paramore	Paramore	29/03/2022	04:56	▮▮▮▮	□
Hard Times	Paramore	Hard Times	29/03/2022	03:02	▮▮▮▮	□
Emergency	Paramore	All We Know Is Falling	29/03/2022	04:00	▮▮▮▮	□
Fake Happy	Paramore	After Laughter	29/03/2022	03:55	▮▮▮▮	□
Decode (Twilight Soundtrack Version)	Paramore	Decode	29/03/2022	04:21	▮▮▮▮	□
Careful	Paramore	Brand New Eyes	29/03/2022	03:50	▮▮▮▮	□
Caught in the Middle	Paramore	After Laughter	29/03/2022	03:34	▮▮▮▮	□
Pool	Paramore	After Laughter	29/03/2022	03:52	▮▮▮▮	□
Misery Business	Paramore	Riot!	29/03/2022	03:31	▮▮▮▮	□

Figure 2: User survey instructions and questions and playlist presentation.