Sequential modeling of Sessions using Recurrent Neural Networks for Skip Prediction

Sainath Adapa
FindHotel
adapasainath@gmail.com

ABSTRACT

Recommender systems play an essential role in music streaming services, prominently in the form of personalized playlists. Exploring the user interactions within these listening sessions can be beneficial to understanding the user preferences in the context of a single session. In the Spotify Sequential Skip Prediction Challenge[1], WSDM, and Spotify are challenging people to understand the way users sequentially interact with music. We describe our solution approach in this paper and also state proposals for further improvements to the model. The proposed model initially generates a fixed vector representation of the session, and this additional information is incorporated into an Encoder-Decoder style architecture. This method achieved the seventh position in the competition[2], with a mean average accuracy of 0.604 on the test set. The solution code is available on GitHub[3].

CCS CONCEPTS

• **Information systems** → *Personalization*; *Music retrieval*; Recommender systems.

KEYWORDS

Deep learning, recurrent neural networks, music, recommender systems, user modelling

1 INTRODUCTION

In many of the music streaming services such as Spotify[4], personalized music recommendation systems play a prominent role. These recommendation systems allow the user to listen to suggested music based on a particular song, the user's mood, time or location. The vast amount of available music and diverse interests exhibited by various users, as well as by the same user during different situations pose considerable challenges to such systems. As part of WSDM Cup 2019[5], the Spotify Sequential Skip Prediction Challenge mainly explores the sequential nature of user interactions during music listening sessions.

Spotify has provided 130 million listening sessions for training for this challenge. Another 30 million sessions are provided as the test set[1]. Each session is divided into two nearly equal halves, with the information about tracks available for both halves of a session. However, the user interaction features are available only for the first half[6]. The task is to predict if the user skipped any of the tracks in the second half.

The length of each session varies from 10 to 20 tracks. This means the model has to predict skipping behavior for five tracks for the shortest sessions, and ten tracks for the longest. Metadata such as duration, release year, and US popularity estimate is provided for every track. Also, audio features such as acousticness, tempo, loudness are provided. For each track that the user was presented within the session, interactions such as seek forward/backward, short/long pause before play are available. Finally, session information such as the time of the day, date, premium user or not, context type of playlist is present.

In the dataset, skipping behavior is classified into four types:

(1) *skip_1*: Boolean indicating if the track was only played very briefly
(2) *skip_2*: Boolean indicating if the track was only played briefly
(3) *skip_3*: Boolean indicating if most of the track was played
(4) *not_skipped*: Boolean indicating that the track was played in its entirety

The objective of the challenge is limited to predicting just the *skip_2* behavior.

Tables 1 to 3 present distributions for some of the features in the dataset[7]. From Table 2, it can be inferred that the *skip_2* variable is fairly balanced between the true and false classes. We can also observe, from Table 3, that the skipping behavior remained consistent from track 4.

The test set distributions of features were found to be very similar to that of the training set. Hence, neither sampling nor other modifications were made to the training set before training the model.

2 RELATED WORK

Deep learning based recommendation methods have been extensively studied during recent years [13]. Session-based music recommender systems specifically try to infer user's preferences and context using information from a single session. This differs from *session-aware* systems which use previous interactions of the same user in the recommendation process [9]. The current task shares many aspects of session-based music recommender systems. Hence, understanding the approaches that are employed to such systems is useful for the current task. A survey and evaluation of various approaches to Session-based recommendation systems was presented in [7]. Recurrent Neural Networks (RNNs) have been shown to work exceedingly well with sequential modeling tasks [4]. As such, in [5], a new architecture named GRU4REC that employs Gated Recurrent Units (GRUs) was proposed to predict the probability of subsequent events given a session beginning. A data augmentation

[1] https://www.crowdai.org/challenges/spotify-sequential-skip-prediction-challenge
[2] Team name: Sainath A
[3] https://github.com/sainathadapa/spotify-sequential-skip-prediction
[4] https://www.spotify.com
[5] http://www.wsdm-conference.org/2019/wsdm-cup-2019.php
[6] Also referred to as *session log features* in this document
[7] Owing to the size of the dataset, a random sample of data was used to calculate the distributions for Table 2 and 3

Table 1: Distribution of number of tracks in sessions

Number of tracks	10	11	12	13	14	15	16	17	18	19	20
Percentage	8.8%	7.7%	6.7%	5.9%	5.2%	4.5%	4.0%	3.5%	3.1%	2.8%	47.7%

Table 2: Overall skipping behavior

Type	True percentage
skip_1	41.52%
skip_2	50.89%
skip_3	63.86%
not_skipped	34.41%

technique that improves upon [5] via sequence pre-processing was proposed in [11]. To model the changes in user behavior based on context, a new recurrent architecture was proposed in [10].

3 APPROACH

3.1 Modified Encoder-Decoder Architecture

In 2014, Cho et al. [3] proposed the Encoder-Decoder architecture, that consisted of two recurrent neural networks (RNN). The Encoder RNN strives to encode the input sequence into a fixed length representation, and from this representation, the Decoder RNN generates a correct, variable length target sequence. The architecture was proposed for the statistical machine translation, for which it was shown to be a significant improvement over previous methods.

The current task of sequential skip prediction shares the variable length and the sequential dependency aspects of the statistical machine translation. However, differing significantly from the statistical machine translation task, the current data set contains information about the tracks in the second half. There is a direct one-to-one correspondence between the track and the output skip prediction. In the following sections, We describe a modified Encoder-Decoder architecture that takes advantage of the unique characteristics of the present dataset.

3.2 Base transformation of input data

The raw input feature vector space might not be an ideal representation for processing by the Long Short-Term Memory (LSTM) cells in the model. Hence, a single Fully-Connected (FC) layer was used to transform the user interaction features and other metadata about the session into a higher dimensional representation. Similarly, a separate FC layer was used to process the acoustic and other metadata associated with tracks. Note that the same FC layer mutates tracks from both the first and the second half of the session. Both the FC layers were equipped with the rectification (ReLU) non-linearity. Only the transformed features were used subsequently in the model.

3.3 Compact representation of the session

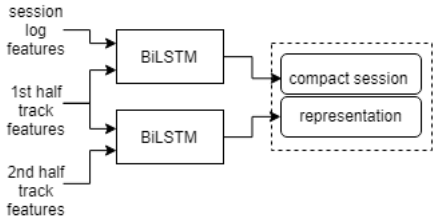
As shown by Cho et al. [2], Encoder-Decoder architectures exhibit weakness in handling long sentences. The fixed-length vector representation that is transferred from encoder to decoder may not have enough capacity to encode the complicated relationships within the data. Many solutions have been proposed to mitigate this issue [8] [12]. In the current approach, a compact representation of the session is generated to preserve information for use by Decoder. This fixed length vector is then concatenated with each track's features, before being fed into the Encoder or Decoder.

The track features corresponding to the first half of the session were concatenated with the first half's user interaction features. This was then fed into a Bi-directional Long Short-Term Memory (BiLSTM) network. The final output from this Bi-directional LSTM can be considered to contain aspects such as overall user behavior in the session and user behavior with respect to specific track features within the session.

During the previous computation, we have only considered the first half of the session. However, we also have the track information for the second half of the session. Usage of track information from both the first and second halves can lead to a better understanding of the nature of the playlist. Hence a second BiLSTM layer was used to transform the long sequence of all the tracks within the session into a fixed length vector.

The combined output from both the Bi-directional LSTMs can now be considered as a compact representation of all the information that is available to the model as input.

Figure 1: Computing a fixed vector representation of the session



3.4 Encoder

At every track's prediction, the Decoder now has easy access to a representation of the session from the input. Hence, the only goal of the Encoder currently is to create the initial state for the decoder. The Encoder consists of an FC layer and a subsequent BiLSTM layer.

3.5 Decoder

The track features of the second half along with fixed vector representation of the session is first sent to an FC layer with ReLU non-linearity. The purpose of this layer can be seen as a context setting mechanism - the track features are being transformed using the current session as the context. Note that the weights of this FC layer are shared with that of the FC layer from the Encoder.

Table 3: "skip_2 = true" percentage by session position

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
37.4	46.7	49.4	51.6	52.4	53.2	53.1	53.0	52.1	50.3	50.7	51.3	51.7	52.2	52.5	53.0	53.3	53.5	53.6	53.6

Figure 2: Encoder architecture

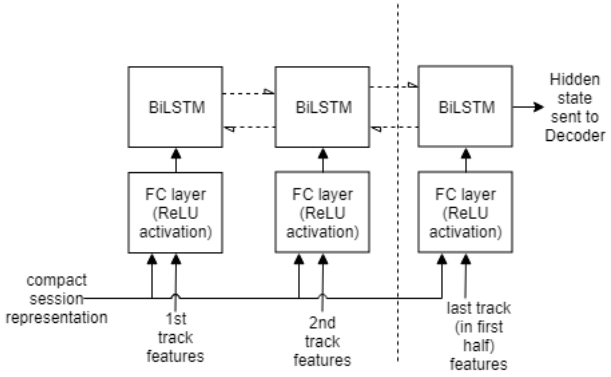
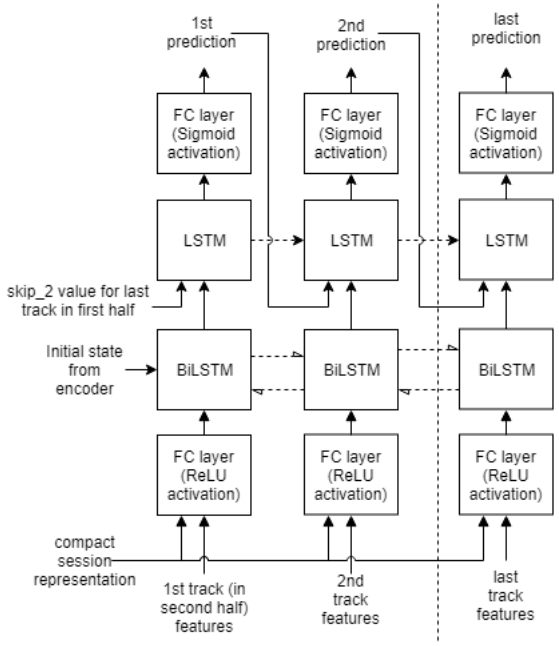


Figure 3: Decoder architecture



The output from the previous layer is now sent to a BiLSTM layer. For the very first track in the second half, the final state from the BiLSTM layer of the encoder is used as the hidden state for this layer.

Skipping behavior exhibited by the user during the previous track is a significant predictor for the current track. Hence, the output from the Decoder for the previous track is combined with the output from the previous layer. In case of the first track, the actual *skip_2* value of the last track in the first half is used. This combined vector is sent to an LSTM layer. Finally, the output from the LSTM layer is fed into a Fully Connected layer with Sigmoid non-linearity to generate the prediction.

4 MODEL TRAINING

4.1 Evaluation metric

For evaluation, Mean Average Accuracy (MAA) was selected as the primary metric for the challenge. Here, the average accuracy is defined as

$$AA = \sum_{i=1}^{T} \frac{A(i)L(i)}{T}$$

where:

- T is the number of tracks to be predicted for the given session
- $A(i)$ is the accuracy at position i of the sequence
- $L(i)$ is the Boolean indicator for if the i'th prediction was correct

The motivation for the metric is the notion that the immediate track's prediction is most important. As a tie-breaking secondary metric, the average accuracy of the first prediction was used.

Table 4: Average Accuracy values for one simulated sample of length 5

Ground truth sequence	Predicted sequence	AA
1, 1, 1, 1, 1	0, 1, 1, 1, 1	0.543
1, 1, 1, 1, 1	1, 0, 1, 1, 1	0.643
1, 1, 1, 1, 1	1, 1, 0, 1, 1	0.710
1, 1, 1, 1, 1	1, 1, 1, 0, 1	0.760
1, 1, 1, 1, 1	1, 1, 1, 1, 0	0.800

4.2 Loss function and weights

Table 4 shows Average Accuracy values for various predictions on a sequence of length 5. As illustrated by the table, a wrong prediction of the first track decreases the AA value by 0.457 whereas an incorrect prediction of the last track would reduce the AA by just 0.2. One way to interpret this is that the first prediction is (0.457/0.2 =) 2.285 times as important as the last prediction in this example. Incorporating this information into the loss function is useful. Hence, session positions were allocated weights in proportion to the decrease in Average Accuracy value when the prediction for that session position alone is incorrect. Log Loss with weights assigned for each session position was used for optimizing the parameters of the model.

4.3 Training

Due to time and resource constraints, we were not able to train the model on the whole data set. Three mutually exclusive sets of sessions were sampled from the data - Training, Validation, and Test. The Model was trained on the training set, using the validation set to determine the stopping point. The parameters were tuned until the loss on validation set plateaued. The third sample - Test set was used to perform a local evaluation of the model.

4.4 Results

Before the results of the proposed method are presented, we introduce the result of a baseline model. The baseline model uses the skipping behavior of the last track in the first half as the prediction for all the tracks in the second half. This model scored 0.537 on MAA and 0.742 on First prediction accuracy. Predictions from the proposed model in this paper resulted in an MAA score of 0.604 and First Prediction accuracy of 0.792 on the hold-out set, thus achieving the seventh position in the competition.

5 FUTURE WORK

Only around 20% of the data was used for training the model because of resource constraints. Instead, training the model on 80% of the data (leaving 20% of the sessions for validation and test sets) might improve accuracy.

During the exploration phase, a Random Forest (RF) model was built to predict the first track of the second half. Another RF model was trained to predict the last track in the second half. This model was built using the session log features until the last track, which is unlike the setup of this challenge where session log features are only available for the first half. Both the models achieved similar levels of accuracy. This reveals that fundamentally, user behavior during the end of the second half is not much more variable (and thus not harder to predict) than during the beginning of the second half. If complete data is available until the previous track, then any session position can be predicted with reasonable accuracy.

In the absence of complete data, we can try to generate predictions for missing features by building a model that predicts those features. Hence if a model was trained to predict all types of user interactions (*skip_1*, *2*, and *3*, *hist_user_behavior_reason_start*, and other remaining features), it is possible that the resulting model might be better at predicting *skip_2*. Similar to the proposed model in this paper, this model can use the previous time step's predictions while predicting the current time step.

Another option is to employ transfer learning. Transfer learning has been useful in many areas, with one prominent example being the use of ImageNet-trained models[6]. As described in the previous paragraph, we can build and train a model to predict all the user interaction features available in the data. Such a model theoretically would have inferred more aspects of the user behavior than a model that is solely trained for *skip_2* prediction. We can then fine-tune the top layers of this model specifically for *skip_2* resulting in better *skip_2* prediction accuracy.

ACKNOWLEDGMENTS

We want to thank WSDM, Spotify, and CrowdAI for organizing the challenge. Special thanks to Google for providing the coupons for using Google Cloud Compute resources. Considering the large size of the dataset, the coupons were especially helpful.

REFERENCES

[1] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The Music Streaming Sessions Dataset. In *Proceedings of the 2019 Web Conference*. ACM.
[2] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).
[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[6] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. 2016. What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614* (2016).
[7] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *arXiv preprint arXiv:1803.09587* (2018).
[8] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).
[9] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *arXiv preprint arXiv:1802.08452* (2018).
[10] Elena Smirnova and Flavian Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. ACM, 2–9.
[11] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 17–22.
[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
[13] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).