BEAT AND DOWNBEAT TRACKING WITH GENERATIVE EMBEDDINGS

Haokun Tian[1,2,*] **Kun Liu**[2] **Magdalena Fuentes**[1]
[1] New York University, New York, USA [2] Xiaohongshu Inc., Shanghai, China
ht2208@nyu.edu

ABSTRACT

It is standard practice to use spectrograms as input features for discriminative MIR tasks. However, recent research showed using representations produced by Jukebox (a music language model) led to better model performance. This was tested on music tagging, genre classification, key detection, emotion recognition, and music transcription. In this paper, we test it on beat and downbeat tracking. Specifically, we compare compressed Jukebox embeddings with spectrograms as input to a model that jointly predicts beat, downbeat, and tempo. Experiments show that the two inputs bring comparable results for beat tracking, while using Jukebox embeddings leads to significant improvements for downbeat tracking.

1. INTRODUCTION

Beat tracking and downbeat tracking are two tasks in the field of Music Information Retrieval (MIR) that can be undertaken independently or jointly. These tasks aim to develop systems that automatically detect the timing of beats and downbeats (the first beat of each measure) in music signals. The typical approach is to train a neural network that converts audio features into beat and downbeat activations. These activations are then fed into a probabilistic graphical model to make beat and downbeat predictions.

Earlier models designed for downbeat tracking employed a variety of input features, including spectral flux, chroma, CQT (Constant-Q Transform), and low-frequency spectrogram [1, 2]. These features were thoughtfully selected; for instance, chroma was chosen because chord changes often occur at downbeats. However, as model learning capabilities advanced and multi-task learning was incorporated, spectrograms have become the standard input [3–8]. This shift can be attributed to the models' ability to learn meaningful transformations during training, making hand-crafted features unnecessary. As an explicit example, trainable harmonic filters can be applied to convert spectrograms into tailored harmonic representations [8]. In parallel, the model architecture has evolved from recurrent neural networks (RNN) and convolutional neural networks (CNN) [1–3], to temporal convolutional networks (TCN) [4–6], and finally to Transformer [7, 8].

Recent research brought new possibilities for music audio representations. Dhariwal *et al.* introduced Jukebox, a music language model trained on 1.2 million songs [9]. They compressed raw audio into discrete codes via VQ-VAE and trained Transformers on these codes for autoregressive generation. Following this, Castellon *et al.* demonstrated embeddings produced by Jukebox were strong representations for downstream MIR tasks [10]. It was tested on music tagging, genre classification, key detection, and emotion recognition, utilizing time-averaged Jukebox embeddings for these non-temporal tasks. Subsequently, Donahue *et al.* developed Sheet Sage, a system for transcribing audio into lead sheets, employing Jukebox embeddings for melody and chord transcription [11]. In this paper, we further investigate this idea by using Jukebox embeddings for beat and downbeat tracking [1].

2. EXPERIMENTS

2.1 Data

We used Ballroom [12, 13], Hainsworth [14], HJDB [15], and SMC [16] for training using 8-fold cross validation, with GTZAN [17, 18] served as our test dataset. Note that models were trained on a combined dataset containing 7 of the 8 splits from each training dataset.

We followed Sheet Sage [11] to choose layer 53 for embedding extraction. This decision was based on the fact that beat and downbeat tracking as a temporal task aligns more closely with music transcription than non-temporal tasks such as music tagging, for which layer 36 was proved to be a better choice [10].

To compute Jukebox embeddings, the audio was segmented into 20-second chunks. This was necessary because the Jukebox model we used can handle a maximum audio length of approximately 23.8 seconds. These audio segments were converted into Jukebox embeddings with a sampling rate of approximately 345 Hz and an embedding size of 4800, using an open-source implementation [2]. Subsequently, these embeddings were resampled to 100 Hz, and the embedding size was reduced from 4800 to 10 via 1D average pooling, aiming to provide appropriate input data for our model. Taking a 20-second audio chunk for

[*] This work was performed while interning at Xiaohongshu.

[1] code available at https://github.com/tiianhk/jukebeat

[2] https://github.com/rodrigo-castellon/jukemirlib

	Beat			Downbeat		
	F1	CMLt	AMLt	F1	CMLt	AMLt
			Ballroom			
S	.956	.929	.957	.907	.903	.957
S$_a$	.961	.942	.960	.914	.911	.960
J	.956	.932	.960	.961	.951	.966
J$_a$	.961	.944	.961	.963	.958	.971
			Hainsworth			
S	.874	.793	.925	.661	.619	.815
S$_a$	.888	.819	.936	.685	.650	.836
J	.885	.817	.926	.772	.720	.883
J$_a$	.903	.856	.942	.794	.760	.891
			HJDB			
S	.972	.971	.987	.949	.957	.972
S$_a$	.979	.977	.988	.955	.960	.972
J	.985	.984	.992	.964	.973	.980
J$_a$	.988	.986	.990	.974	.980	.986
			SMC			
S	.545	.439	.629			
S$_a$	.544	.452	.641			
J	.551	.435	.627			
J$_a$	.568	.470	.651			
			GTZAN			
S	.860	.756	.923	.590	.547	.794
S$_a$	.862	.767	.919	.623	.577	.802
J	.863	.763	.922	.736	.671	.880
J$_a$	.878	.795	.926	.749	.689	.881
[7]	.885	.800	.922	.714	.665	.844
[8]	.887	.812	.920	.756	.715	.881

Table 1. Results of beat and downbeat tracking using 8-fold cross validation. GTZAN is an unseen dataset for test only. Input data types are denoted as follows: 'S' for spectrogram and 'J' for Jukebox embedding. The subscript 'a' signifies the use of data augmentation.

example, its Jukebox embedding shape was (2000, 4800) after resampling and (2000, 10) after dimension reduction. The resulting embeddings were concatenated along the time axis. Thus, for a 30-second audio clip, the Jukebox embedding shape was (3000, 10). Additionally, to implement the data augmentation technique proposed in [6], Jukebox embedddings were further resampled to 95, 97.5, 102.5, and 105 Hz to generate more training data.

2.2 Model

We experimented with two different inputs: the spectrogram and the Jukebox embedding. The spectrogram was produced by firstly computing the STFT with a window and FFT size of 2048 samples, and a hop size of 441 samples. With an audio sampling rate of 44.1 kHz, this resulted in 100 frames per second for the STFT. The STFT was then filtered using the FilteredSpectrogramProcessor in madmom [19] with default parameters. Lastly, the magnitudes were converted into the logarithmic scale. As mentioned, the Jukebox embedding was resampled to 100 Hz. Thus, the two inputs were time-aligned.

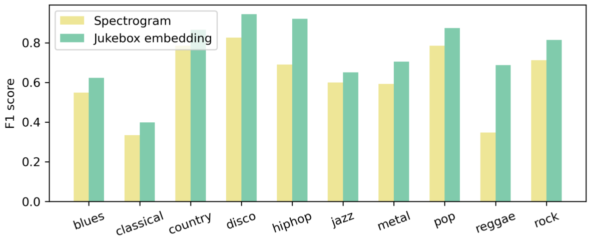

Figure 1. F1 scores of downbeat tracking on GTZAN sub-genres. Models are trained using spectrogram (left) and Jukebox embedding (right) with data augmentation.

We used the lightweight multi-task model proposed in [6] and adapted its open-source implementation available in [20]. The model architecture comprised CNN layers for feature extraction, succeeded by TCN layers for temporal modeling. The model was designed to simultaneously predict beat, downbeat, and tempo. For the spectrogram input, the CNN layers compressed the frequency dimension from 81 to 1 while expanding the channel dimension from 1 to 20. The output of the CNN was then fed into the TCN. In comparison, the Jukebox embedding was directly fed into the TCN, as we believed it required no additional feature extraction. More details about data augmentation, model architecture, and training process could be found in [5, 6, 20]. After training, we used DBNBeatTrackingProcessor and DBNDownBeatTrackingProcessor, two dynamic Bayesian networks from madmom [19], to infer beat and downbeat from corresponding frame-level activations.

2.3 Results

We report beat and downbeat tracking results for two inputs (the spectrogram and the Jukebox embedding) in Table 1. For GTZAN we additionally include two Transformer-based models [7, 8] for comparison, as they both use GTZAN as test-only data. As shown, the beat tracking results are comparable for both inputs. However, by using the Jukebox embedding, the downbeat tracking results are significantly improved over the spectrogram: F1 score improved by about 5% on Ballroom and over 10% on Hainsworth and GTZAN. Notably, our best downbeat F1 score on GTZAN is close to the state-of-the-art result with a difference smaller than 1%. In Figure 1, we further present downbeat tracking results across GTZAN sub-genres. We observe that using the Jukebox embedding doubles the F1 score for reggae, but gives only a limited boost for classical.

3. DISCUSSION

The model we used is lightweight, but the Jukebox model's inference is computationally intensive. For future research, one avenue is to optimize the inference time by focusing only on generating the compressed embedding, rather than the full embedding. Another option is to explore the use of uncompressed Jukebox embeddings with larger models like Transformer, aiming for maximum accuracy.

4. REFERENCES

[1] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of common design choices in deep learning systems for downbeat tracking," in *ISMIR*, 2018.

[2] S. Durand, J. P. Bello, B. David, and G. Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 76–89, 2016.

[3] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks." in *ISMIR*, 2016, pp. 255–261.

[4] E. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[5] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other." in *ISMIR*, 2019, pp. 486–493.

[6] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation." in *ISMIR*, 2020, pp. 574–582.

[7] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *ISMIR*, 2022.

[8] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 401–405.

[9] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[10] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *ISMIR*, 2021.

[11] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *ISMIR*, 2022.

[12] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[13] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio." in *ISMIR*, 2013, pp. 227–232.

[14] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, pp. 1–11, 2004.

[15] J. Hockman, M. E. Davies, and I. Fujinaga, "One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass." in *ISMIR*, 2012, pp. 169–174.

[16] A. Holzapfel, M. E. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[17] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[18] U. Marchand and G. Peeters, "Swing ratio estimation," in *Digital Audio Effects (DAFx)*, 2015.

[19] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1174–1178.

[20] M. F. Matthew E. P. Davies, Sebastian B ock, *Tempo, Beat and Downbeat Estimation*. https://tempobeatdownbeat.github.io/tutorial/intro.html, Nov. 2021. [Online]. Available: https://tempobeatdownbeat.github.io/tutorial/intro.html