BebopNet: Deep Neural Models for Personalized Jazz Improvisations

Shunit Haviv Hakimi*, Nadav Bhonker*, Ran El-Yaniv

Computer Science Department
Technion – Israel Institute of Technology

shunithaviv@gmail.com, nadavbh@gmail.com, rani@cs.technion.ac.il

Samples and supplementary material can be found in https://shunithaviv.github.io/bebopnet/

Abstract

Personalized content is all around us. Tech giants provide to their users personalized streams of **existing** content. However, the challenge of **generating new personalized content** has been examined only rarely in the literature.

We propose a novel pipeline for personalized music generation that learns and optimizes user-specific musical taste.

We focus on the task of generating symbol-based, monophonic, harmony-constrained jazz improvisations.

Personalization Pipeline

Our Personalization Pipeline consists of three major parts:

Train BebopNet

Generate symbolic saxophone jazz improvisations to any chord progression

Learn User Preference Metric

Assemble a personal dataset and train a personal metric to predict user preference

Optimize BebopNet to User

Use beam search to optimize the generation process for our user

Dataset

Our dataset, consisting of jazz solo transcriptions purchased as XML files from saxsolos.com, comprises jazz improvisations by:

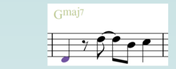
- Charlie Parker (1920–1955)
- Cannonball Adderley (1928–1975)
- Sonny Stitt (1924–1982)
- Phil Woods (1931–2015)
- Sony Rollins (1930-)
- Stan Getz (1927–1991)
- Gene Ammons (1925–1974)
- Dexter Gordon (1923–1990)

BebopNet

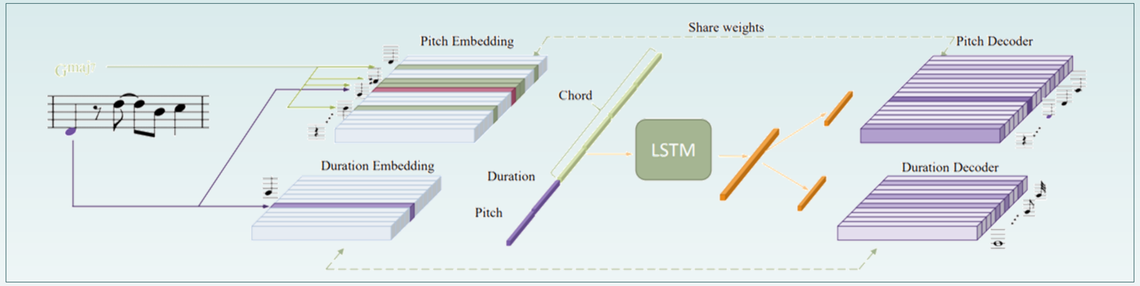
BebopNet predicts the next note to be played given some past notes and the past harmonic progression as well as the forthcoming harmony (over which the next note should be played).

BebopNet is based on a symbolic note representation that closely resembles the standard music notation system used to communicate music by musicians. Each object represents a musical note with its pitch, duration, offset within a measure, and harmonic context. The harmonic context includes the four notes of the current chord.

Our neural network begins with learned embeddings for pitch and duration, which are then used to construct the input to the main part of the network that can be either an LSTM or a transformer. The chord is represented using the concatenation of the embeddings of the 4 pitches that consist it. The output is decoded using the same embedding weights to predict the duration and pitch of the note to be played next.

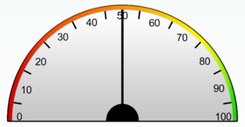


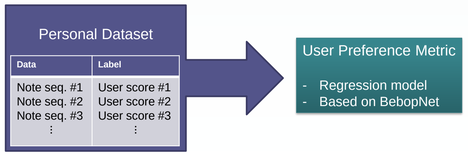
Pitch	62	128	74	71	72
Duration	4	2	4	2	4
Offset	0	12	18	30	36
Chord	67	67	67	67	67
	71	71	71	71	71
	74	74	74	74	74
	78	78	78	78	78



User Preference Learning

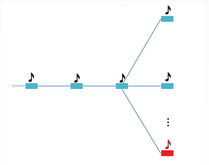
In the second stage, we collect a training set for a specific user and train a personal preference metric.
Each user is presented with jazz improvisation which they are required to rate according to their preference. As the music plays, the user adjusts the meter to display the level of satisfaction to the currently heard jazz solo.
Thereafter, we train a regression model to predict the user's tastes.



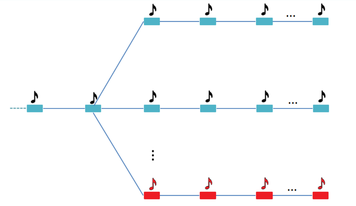


Optimized Generation - Beam Search

Having a trained user scoring model, we employ beam-search over BebopNet.
The beam search uses the criterion of the score of the user preference model.
At each step, BebopNet generates multiple options for the next note to be played. Then, we use the personal metric to calculate the user-score for each option and select the best one according to the score.
We can also consider a deeper search, in which we use the personal metric after generating multiple sequences as continuation options and choose the preferred sequence.



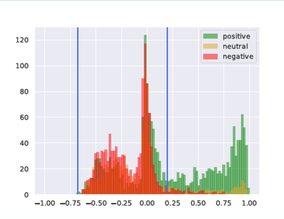
Greedy Beam-Search



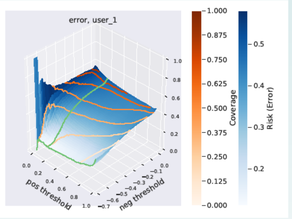
Deeper Beam-Search

User Analysis

We analyze the quality of our preference metric by plotting a histogram of the network's predictions applied on a validation set. We colored the musical sequences according to the user label, green for a positive score, yellow for neutral and red for negative scores. While the overall error is high, it is still useful since we are interested in its predictions in the positive (green) spectrum for the optimization stage. While trading-off coverage, we increase prediction accuracy using selective pre-diction by allowing our classifier to abstain when it is not sufficiently confident. To this end, we ignore predictions whose magnitude is between two rejection thresholds. We present a risk-coverage plot for user-1. The risk surface is computed by moving two thresholds across the histogram and calculating the risk (error of classification to positive, neutral and negative) and the coverage (percent of data maintained).



Histogram of predictions of the preference model on sequences from a validation set (thresholds in blue)



Risk-coverage plot for the predictions of the preference model (selected thresholds in green)

Plagiarism

In our paper, we also present a plagiarism analysis to evaluate originality.
We measure the length of the largest common musical sequence between any two artists in our training set, as well as with BebopNet. The average level of plagiarism for BebopNet lies within the averages of other artists.

Name	Adderley	Gordon	Getz	Parker	Rollins	Stitt	Woods	Ammons	Mean	Ours
Adderley	4	6	6	4.7	5.7	4.5	5	4.5	5	6.2
Gordon	3.4	6.4	5.1	4.2	4.6	3.8	3.5	4.2	4.4	4.6
Getz	3.3	4.6	5.7	4.4	4.2	3.8	3.5	4.2	4.2	4
Parker	3.7	5	5.1	6	5.1	4.1	3.6	5	4.7	4.3
Rollins	3.6	4.9	4.6	4.4	4.7	3.8	3.5	4.2	4.2	4.1
Stitt	4	7	7.2	5.6	5	10.3	4.1	5.6	6.1	4.7
Woods	4.1	5	5.8	4.8	5.4	4.4	5.4	5.1	5	3.8
Ammons	3.3	4.8	4.8	3.9	4.3	4	3.6	5.3	4.2	3.9
Mean	3.7	5.5	5.5	4.7	4.9	4.8	4	4.8	-	4.4
Ours	2.7	3.9	3.8	3	3.4	2.8	2.8	3.6	3.3	3.8

Each element in the table is the average largest subsequence in a solo of artist A (row names) found in any solo of artist B (column names).