Rhythm, Chord and Melody Generation for Lead Sheets using Recurrent Neural Networks

Cedric De Boom, Stephanie Van Laere, Tim Verbelen, and Bart Dhoedt

IDLab, Department of Information Technology at Ghent University – imec
Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium
`firstname.lastname@ugent.be`

Abstract. Music that is generated by recurrent neural networks often lacks a sense of direction and coherence. We therefore propose a two-stage LSTM-based model for lead sheet generation, in which the harmonic and rhythmic templates of the song are produced first, after which, in a second stage, a sequence of melody notes is generated conditioned on these templates. A subjective listening test shows that our approach outperforms the baselines and increases perceived musical coherence.

Keywords: Music generation · lead sheets · neural networks

1 Lead Sheets

Lead sheets are widely used to represent the fundamental musical information about almost any contemporary song: they contain a chord scheme, a melody line, some navigation and repetition markers, and sometimes lyrics. They seldom contain information about the instrumentation or accompaniment, so any band can take a lead sheet as a guideline and make the song their own, sometimes even by improvising over the chord schemes. In this paper we focus on generating chords and melody lines for lead sheets from scratch.

A major difficulty in music generation is that harmony, melody and rhythm all influence each other. For example, a melody note can change whenever the underlying harmony changes, and vice versa. Rhythmic patterns can influence which notes are played, and rhythm and harmony together define the overall groove of the piece. To tackle this issue, we split the generation process into two stages. First, we generate a harmonic progression using chord sequences, while simultaneously picking the most appropriate rhythmic patterns. And in a second step the melody is generated on top of this harmonic and rhythmic template.

We are, however, not the first to tackle the problem of lead sheet generation and, in general, music generation. Briot et al. provide a recent and extensive overview of all deep learning based techniques in this field [?]. Regarding lead sheets specifically, Liu et al. use GAN-based models on piano roll representations, but the melody and chords are still predicted independently by different generators [?]. Roy et al. devise a lead sheet generator with user constraints defined by Markov models, and harmonic synchronization between melody and

i	c_i	r_i	m_i
1	C	quarter	G4
2	C	eighth	E4
3	C	eighth	G4
4	C	half	C5
5	|	|	|
6	G	quarter	D5
7	G	quarter	B4
8	G	half	G4
9	|	|	|

Fig. 1: Example of a lead sheet decomposition into chords, rhythms and melodies.

chords through a probabilistic model that encodes which melody notes fit on which chords [**?**]. There have also been many efforts in the past that learn to generate chords for a given melody [**?**,**?**,**?**] or the other way round [**?**]. In this paper we want to show that, on the one hand, a two-stage generation process greatly improves the perceived quality of the music. And, on the other hand, we show that melodic coherence improves when the melody generator gets to look ahead at the entire harmonic template of the song.

We formally define a lead sheet $x_{1:n}$ of length n, characterized by a sequence of chords $c_{1:n}$, rhythms $r_{1:n}$ and melody pitches $m_{1:n}$. At each time step i in the piece, each of these quantities take some value:

$$x_{1:n} = \{c_{1:n}, r_{1:n}, m_{1:n}\}, \quad x_i = \{c_i, r_i, m_i\}. \tag{1}$$

In this equation, the compact $x_{j:k}$ notation denotes the sequence $(x_j, x_{j+1} \ldots x_k)$. Figure **??** shows an example of this decomposition. Notice that the chords are repeated until there is a change in harmony, thereby allowing us to model the entire lead sheet using a single shared time scale. We also point out that there is only one melody note per time step, which is not a severe restriction, since most lead sheets only contain monophonic melodies. Finally, we choose to treat the barlines as separate elements in the sequence, which is indicated by the vertical bars in Figure **??**.

2   The Wikifonia Dataset

In this paper we will make use of the Wikifonia dataset, a former public lead sheet repository hosted by wikifonia.org. It contains more than 6,500 lead sheets in MusicXML format, and in all sorts of modern genres. This section goes over the different preprocessing and encoding steps that are executed on the dataset in order to obtain a clean collection of lead sheets.

2.1   Preprocessing

Eliminate polyphony Whenever multiple notes sound at the same time, we only retain the note with the highest pitch, as it is often the note that characterizes the melody.

Ignore ties Connections between two notes with the same pitch that extend the first note's duration are ignored. The two notes are therefore treated as two separate notes with their original duration.

Delete anacruses Incomplete bars that often appear at the start of a piece, are removed from all lead sheets.

Unfold repetitions Lead sheets can contain repetition and other navigation markers. If a section should be repeated, we duplicate that particular section, thereby unfolding the piece into a single linear sequence.

Remove ornaments Since such ornaments do not contribute much to the overall melody, we leave them out.

2.2   Data encoding and features

After preprocessing, we encode the melody, rhythm and chord symbols into feature vectors such that they can be used as input to our generators.

Encoding rhythms We retain the 12 most common rhythm types in the dataset, which are given in Appendix **??**. We remove 184 lead sheets from the dataset that contain other than these 12 types. Together with the representation for a barline, we encode rhythm into a 13-dimensional one-hot vector r_i.

Encoding chords A chord is described by both its root and its mode. There are 12 possible roots (C, C♯, D, D♯, ..., B) and we choose to convert all accidentals to either no alteration or one sharp.We count 47 different modes in the dataset, which we map to one of the following four: major, minor, diminished or augmented. This mapping only very slightly reduces musical expressivity and interestingness. The mapping table can be found in Appendix **??**. The 12 roots and 4 modes give 48 chord options in total, resulting in a 49-dimensional one-hot vector c_i if we include the barline.

Encoding melody The MIDI standard defines 128 possible pitches. We assign two additional dimensions for rests and barlines, resulting in a 130-dimensional one-hot encoded melody vector m_i.

3   Recurrent Neural Network Design

As mentioned in Section **??**, the lead sheet generation process happens in two stages: in stage one the rhythm and chord template of the song is learned, and in stage two the melody notes are learned on top of that template. We will use separate LSTM-based models for both stages [**?**]; the models are trained independently of each other, but they are combined at inference time to generate an entire lead sheet from scratch. Figure **??** shows the complete architecture.
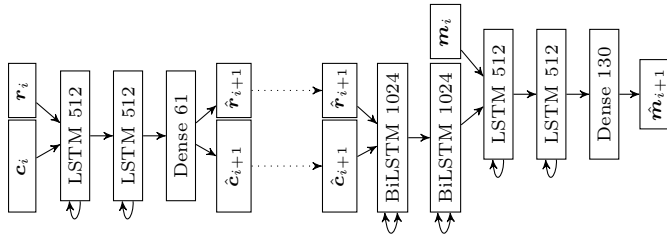
Fig. 2: The RNN architecture for both stage one (left) and stage two (right). The output dimensionality for every layer is written in each of the blocks. Whenever two blocks appear next to each other, the (output) vectors are concatenated.

Stage one In this stage, the rhythm and chord vectors are first concatenated and are subsequently given as inputs to two LSTM layers followed by a dense layer. All LSTM layers have a output dimensionality of 512 states, as indicated in the figure. The output of the dense layer is cut in two vectors, both on which we apply a softmax nonlinearity with temperature τ, controlling the concentration of the output distribution. This way we are effectively modeling a distribution over the chord and rhythm symbols that come next in the sequence.

Stage two The second model will process the generated sequence of predicted chords and rhythms. To this end, each chord and rhythm vector is again concatenated before being processed by two BiLSTM layers. These BiLSTM states allow the pitch generator to look back and also ahead at the harmonic sequence, as inspired by [?]. The dimensionality of the BiLSTM layers is 512 in both directions, adding up to a total of 1024 states. After concatenation with the previous melody vector, the BiLSTM states are fed through another stack of two LSTM layers. The output of the last dense layer is used to predict the next melody note, again using a softmax nonlinearity controlled by a temperature parameter.

3.1   Optimization details

In this paper we train both stages separately; it is possible to jointly train both models through reparameterization tricks [?], but we leave this as future work. While training is done separately, inference of lead sheets from scratch is easily done by feeding the output of the first model to the input of the second model. We use a standard cross-entropy loss function on all outputs. In stage one we sum the losses on both the chord and rhythm outputs with hyperparameter α:

$$\mathcal{L}_{\text{stage 1}}(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}) = \alpha \cdot \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{c}}) + (1 - \alpha) \cdot \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{r}}). \tag{2}$$

In this equation, $\mathcal{L}_{\text{CE}}(\cdot)$ indicates the cross-entropy loss. In stage two the loss function is equal to the cross-entropy loss on the melody output.

$$\mathcal{L}_{\text{stage 2}}(\hat{\boldsymbol{m}}) = \mathcal{L}_{\text{CE}}(\hat{\boldsymbol{m}}). \tag{3}$$

We use Adam with learning rate λ to optimize both models [**?**]. During optimization, the training data is augmented by shifting the pitches and chords with a random number of semitones between -12 and 12, that is, between plus or minus one octave. This virtually increases the amount of training data by a factor of 25, thereby aiding model generalization towards less frequently appearing keys.

4   Experiments

Hyperparameters In all experiments we use a batch size of 128 sequences, each of length 100. The learning rate λ is set fixed to 0.001. We empirically found that a value α of 0.5 leads to good results, so we set it fixed to that value. We also set the temperature τ slightly lower than 1 during inference of the melody, which helps to improve the perceived quality of the generated music; we varied it between 0.75 and 1.0 during the experiments. For the rhythm and chord patterns a temperature of 1.0 gives the most pleasing results.

Baselines We will compare our model against two baselines:

1. An unconditioned LSTM-based model similar to the stage one model in Figure **??**, but now the melody is also concatenated to the input and output. The melody is no longer conditioned on the entire chord and rhythm sequence. We also add an extra LSTM layer, adding up to a total of three.
2. A two-stage model where the BiLSTM layers are replaced by regular LSTM layers, so that the melody cannot look ahead at the harmonic sequence. We keep all other parameters identical to the original model.

Subjective listening test We conducted an online listening test in which we asked 40 participants to score 12 short audio clips, each of approximately one minute long. The following songs were included in the test[1]:

- 3 pieces generated by the two-stage model from scratch,
- 3 pieces generated by the two-stage model, but conditioned on the chord and rhythm scheme of existing songs: *I Have a Dream* (Abba), *Autumn Leaves* (jazz standard) and *Colors of the Wind* (Alan Menken),
- 2 pieces generated by the one-stage baseline model,
- 2 pieces generated by the two-stage baseline model,
- 2 (relatively unknown) human-composed songs: *You Belong to my Heart* (Bing Crosby) and *One Small Photograph* (Kevin Shegog).

As stated in Section **??**, a lead sheet only encodes the basic template of a song, and it ideally needs to be played by a real musician. We therefore gave all lead sheets to a semi-professional pianist; the pianist stayed true to the sheet music, but was free to create an accompaniment that suited the piece. In our regards, this evaluation method reflects best how a lead sheet, produced by an AI model, would in practice be used and experienced by musicians and listeners.

---

[1] Listen to the audio clips at `https://users.ugent.be/~cdboom/music`

Model	Pleasing $\bar{Z}$	Coherence $\bar{Z}$	Turing $\bar{Z}$
One-stage	-0.23 ± 0.34	-0.24 ± 0.31	-0.22 ± 0.29
Two-stage, without BiLSTM	-0.04 ± 0.35	-0.07 ± 0.33	-0.09 ± 0.33
Two-stage, with BiLSTM	-0.01 ± 0.36	0.01 ± 0.34	-0.02 ± 0.34
Two-stage, with existing chords	$\mathbf{0.15} \pm 0.37$	0.09 ± 0.36	$\mathbf{0.13} \pm 0.36$
Human-composed songs	0.03 ± 0.34	$\mathbf{0.13} \pm 0.36$	$\mathbf{0.13} \pm 0.34$

Table 1: Results of the subjective listening experiments. We report averaged $\bar{Z}$-scores for each of the questions, along with the standard deviations.

The audio clips were presented to each user in randomized order. For each clip we asked the user to rate on a scale of 1 to 5 how much he likes the piece, if the melody is musically coherent, and whether the piece is composed by a computer (1) or a human (5). We also asked the user to indicate if he recognizes the piece. Since each user has his own rating bias and spread [?,?], we converted the ratings for each user to a standardized Z score between -0.5 and 0.5:

$$Z_{c,u} = \frac{R_{c,u} - \mu_u}{\max_{c'} R_{c',u} - \min_{c'} R_{c',u}}. \tag{4}$$

In this formula, $R_{c,u}$ is the rating of user u for clip c, μ_u is the average rating of user u, and $Z_{c,u}$ is the associated standardized score. Table ?? reports the average $\bar{Z}$ score across the audio clips for each of the three questions in the survey, along with the standard deviation. A negative score means that the ratings are below average overall, and a positive score indicates an overall above-average rating.

We observe that the scores are, by far, better for the two-stage models compared to the unconditioned one-stage model. This shows that first sampling a harmonic and rhythmic sequence, and conditioning the melody on top of this sequence, is more beneficial than sampling all quantities simultaneously. Next to this, we also notice that adding the BiLSTM layers improves the score for all three questions. And although by a small margin, we can conclude that the musical quality improves when the melody generator can look ahead in the harmonic sequence. When we condition the melody generator on an existing chord and rhythm scheme, it is remarkable that the human-composed and AI-composed songs perform almost on par. The AI-composed songs are even considered most pleasing. Related to this observation, 4 participants indicated having recognized a piece from the two-stage model, 5 recognized a piece that was generated based on existing chords, and 3 participants recognized a human-composed song.

Finally, we also want to point out that the standard deviations are very substantial, which shows that there is a high level of disagreement between the reviewers. It is however interesting to point out that the standard deviation is slightly higher for better performing models. This might indicate that there is more consensus on what it means for music to sound 'badly', but that the definition of 'good' music is more subjective and person-dependent.

5   Conclusion

We have proposed a two-stage LSTM-based model to generate lead sheets from scratch. In the first stage, a sequence of chords and rhythm patterns is generated, and in the second stage the sequence of melody notes is generated conditioned on the output of the first stage. We conducted a subjective listening test of which the results showed that our approach outperformed the baselines. We can therefore conclude that conditioning helps the quality of the generated music, and that this approach can be explored further in the future.

References

1. Briot, J.P., et al.: Deep Learning Techniques for Music Generation - A Survey. arXiv.org (2017)
2. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
3. Huang, C.Z.A., et al.: Music Transformer: Generating Music with Long-Term Structure. In: ICLR (2019)
4. Jang, E., et al.: Categorical Reparameterization with Gumbel-Softmax. In: ICLR (2017)
5. Jin, R., Si, L.: A study of methods for normalizing user ratings in collaborative filtering. In: SIGIR (2004)
6. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
7. Koren, Y., et al.: Matrix Factorization Techniques for Recommender Systems. Computer (2009)
8. Lim, H., et al.: Chord Generation from Symbolic Melody Using BLSTM Networks. In: ISMIR (2017)
9. Liu, H.M., Yang, Y.H.: Lead Sheet Generation and Arrangement by Conditional Generative Adversarial Network. arXiv.org (2019)
10. Pachet, F., Roy, P.: Non-Conformant Harmonization - the Real Book in the Style of Take 6. In: ICCC (2014)
11. Roy, P., et al.: Sampling Variations of Lead Sheets. arXiv.org (2017)
12. Yang, L.C., et al.: MidiNet - A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation. In: ISMIR (2017)

A   Mode Mapping for Chords

In Table **??** we show how different chord modes are mapped to one of the following four options: major, minor, diminished or augmented.

Original mode	Mapped mode	Original mode	Mapped mode
6	major	major-6-9	major
7	major	major-7	major
9	major	major-9	major
augmented	augmented	major-minor	major
augmented-7	augmented	minor	minor
augmented-9	augmented	minor-11	minor
diminished	diminished	minor-13	minor
diminished-7	diminished	minor-6	minor
dominant	major	minor-7	minor
dominant-11	major	minor-7-b5	diminished
dominant-13	major	minor-9	minor
dominant-7	major	minor-major	minor
dominant-9	major	minor-major-7	minor
half-diminished	diminished	power	major
major	major	sus2	major
major-13	major	sus4	major
major-6	major	sus4-7	major

Table 2: Chord modes are mapped to one of four options.

B   Rhythm types

Table **??** provides an overview of the twelve rhythmic figures that are used.

Textual description	Musical symbol
32nd note	
32nd dotted note	
16th note	
8th triplet note	
8th note	
quarter triplet note	
8th dotted note	
quarter note	
quarter dotted note	
half note	
half dotted note	
whole note	

Table 3: The rhythm types that are considered in this paper.