The Supplementary Document for XAI-Lyricist: Improving the Singability of AI-Generated Lyrics with Prosody Explanations

Qihao Liang[1] , **Xichu Ma**[1] , **Finale Doshi-Velez**[2] , **Brian Lim**[1] and **Ye Wang**[1] ,
[1]School of Computing, National University of Singapore
[2]Harvard University
qihao.liang@u.nus.edu, ma_xichu@nus.edu.sg, finale@seas.harvard.edu, {brianlim, wangye}@comp.nus.edu.sg

1 Details of Tracking Melody Rhythm and Lyrics Prosody

1.1 Melody Rhythm Tracking

This section elaborates on the method of extracting melody rhythm. We discuss 4/4 time music, which is commonly used in contemporary pop songs. In musicology, the rhythmic pattern for 4/4 time music within each measure is typically "strong, weak, sub-strong, weak". Following [Zhang *et al.*, 2023] and Schenkerian music theory, we generalise this rhythmic pattern based on different note types. Figure 1 displays the rhythmic pattern for different note types in 4/4 time, where the strength of a melodic note depends on both its onset (viz., its beginning time) and its duration.

To extract the rhythm of melodies, we need to procure the measure number m_n, beat number b_n and the duration of each melodic note n to determine the strength and length of n. For each n, we parse the MIDI file and obtain its duration $|n|$ and onset t_n (viz., the beginning time of n). The measure number m_n and beat number b_n of n are computed as follows:

$$\begin{aligned} m_n &= \frac{\mathrm{quant}(t_n)}{|\hat{M}|} \\ b_n &= \frac{|m_n|\hat{M}| - \mathrm{quant}(t_n)|}{|n|} \end{aligned} \tag{1}$$

Here, $|\hat{M}|$ denotes the standard length of a measure in the MIDI melody; $\mathrm{quant}(\cdot)$ is a function that quantises the onset t_n to the nearest beat. It is worth noting that in the MIDI format, time is discretised into ticks, viz., the minimum unit of time. All time-related information retrieved from MIDI, such as $|n|$, t_n, $|\hat{M}|$, is measured by ticks. The strength of n is classified by the following $\mathrm{NoteStrength}(\cdot)$ function:

$$\mathrm{NoteStrength}(n) = \begin{cases} \text{strong}, & \text{if } b_n \bmod 4 \in \{1, 3\} \\ \text{weak}, & \text{otherwise} \end{cases} \tag{2}$$

where mod here denotes the modulo operation. Similarly, we also categorise n based on its duration:

$$\mathrm{NoteLength}(n) = \begin{cases} \text{long}, & \text{if} |n| > |\bar{n}| \\ \text{short}, & \text{otherwise} \end{cases} \tag{3}$$

Here, $|\bar{n}|$ denotes the average note duration of the melody.

1.2 Lyrics Prosody Extraction

We use the Python package named Prosodic [1] to query the IPA symbol of each word (w) in lyrics, denoted by $\boldsymbol{I^w}$:

$$\boldsymbol{I^w} = \{\mathbf{i}_k^w\}_{k=1}^{K} \tag{4}$$

In this equation, $\mathbf{i}_k^w$ denotes the IPA symbol for the k-th syllable in the word w. K is the total number of syllables in this word. Each $\mathbf{i}_k^w$ is represented as a string:

$$\mathbf{i}_k^w = \{\mathbf{c}_l^{w,k}\}_{l=1}^{L} \tag{5}$$

where $\mathbf{c}_l^{w,k}$ denotes the l-th character in $\mathbf{i}_k^w$, and L is the total number of characters in $\mathbf{i}_k^w$. Then the syllable strength of $\mathbf{i}_k^w$ can be defined as:

$$\mathrm{SyllableStrength}(\mathbf{i}_k^w) = \begin{cases} \text{strong}, & \text{if } \exists\, \mathbf{c}_l^{w,k} \in \{ˈ, ˌ\} \\ \text{weak}, & \text{otherwise} \end{cases} \tag{6}$$

The syllable length of $\mathbf{i}_k^w$ can be defined as:

$$\mathrm{SyllableLength}(\mathbf{i}_k^w) = \begin{cases} \text{long}, & \text{if } \exists\, \mathbf{c}_l^{w,k} \in \{ː\} \cup \mathbb{D} \\ \text{short}, & \text{otherwise} \end{cases} \tag{7}$$

where $\mathbb{D}$ is the set of IPA symbols for diphthongs.

2 Example Demonstrations in the Human-Subjects Study

This section details the demonstrations for lyrics with and without explanations used in the human-subjects study. Figure 2 displays an example demonstration with singability explanations based on the "*Hey Jude*" song by Beatles. It visualises the correspondence between melody notes and lyrics words, rendering notes in different colours, and words in lower and upper cases based on musical prosody.

In contrast, Figure 3 displays an example visualisation for Vanilla and Prosody model without singability explanations, where only plain text captions are added to each melody sentence for participants' reference.

[1]https://pypi.org/project/prosodic/

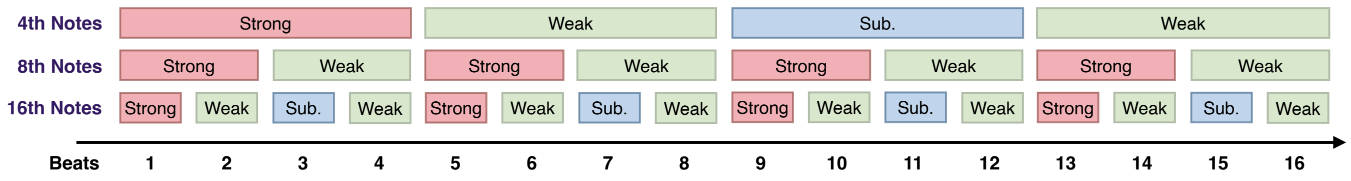

Figure 1: The rhythmic pattern for different note types in 4/4 time music. “Sub.” is short for “sub-strong”.

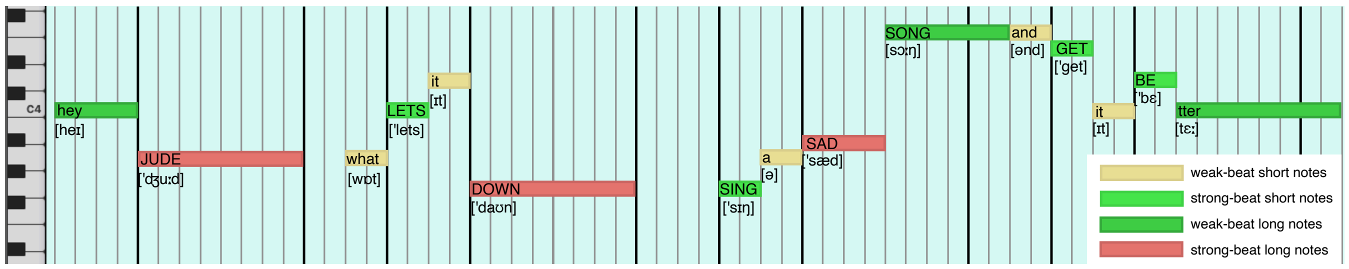

Figure 2: An example demonstration **with** singability explanations based on “*Hey Jude*” by Beatles.

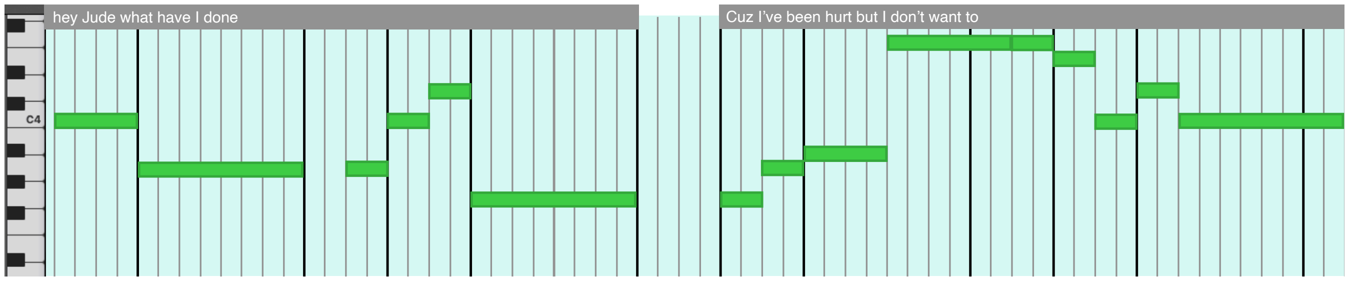
hey Jude what have I done
Cuz I’ve been hurt but I don’t want to
C4

Figure 3: An example demonstration **without** singability explanations based on “*Hey Jude*” by Beatles.

References

[Zhang *et al.*, 2023] Kejun Zhang, Xinda Wu, Tieyao Zhang, Zhijie Huang, Xu Tan, Qihao Liang, Songruoyao Wu, and Lingyun Sun. Wuyun: Exploring hierarchical skeleton-guided melody generation using knowledge-enhanced deep learning. *arXiv preprint arXiv:2301.04488*, 2023.