Nicholas Trieu, supervised by Prof. Keller
CSCI 186: Independent Study
Semester Report
Date: Friday May 5, 2017

Parsing roadmaps, and connections between story telling and music improvisation

1. Summary

This independent study explored 2 ways of furthering music improvisation: literary story generation methods, and brick decomposition [1]. A Python script was developed to parse pretty-printed songs into their bricks and chords. Of the resulting bricks, most (>70%) were used at most three times; however, this may have been due to the limited corpus (100 insight roadmaps). A 1st-order Markov analysis revealed that the most popular transitions actually kept to the same key, and just a couple of bricks (i.e. straight cadence) dominated the transitions.

2. Story generation methods

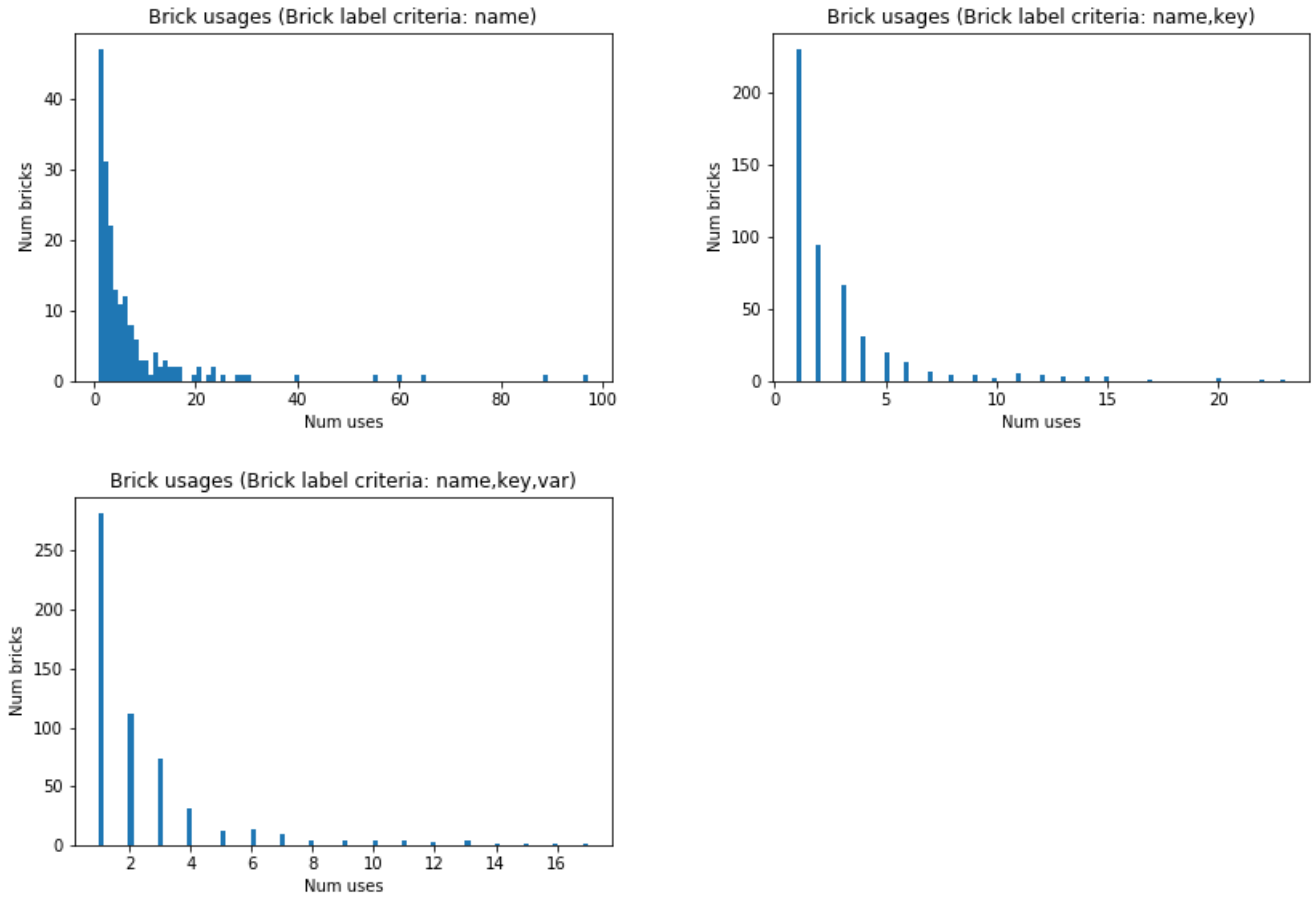
One of the challenges of music improvisation is generating music according to an overarching theme, giving the music consistency. To this end, I turned to the story generation field in the hopes of adapting their techniques to music. It was found that the literary story generation methods resembled the music improvisation methods [2]: various authors emulated neural networks, event graphs, genetic algorithms, and grammars. These methods attempted to generate plot rules, plot events, or plot goals (i.e. motivation, engagement, resolution). Due to their ambiguous nature, inevitably some author interference was necessary to define the rules used. No new and sophisticated generative methods were found; the story generation methods seemed to be hampered by the ill-defined nature of story plots, which have a much vaster search space (language) than monophonic music.

Compared to the music improvisation algorithms, the story generation methods seemed far less sophisticated and consistent. However, some interesting similarities arose. Most generation methods focused on the high-level plot structure; the inner details were mostly author-written. Interestingly, some authors created low-level ”bricks” by constructing sets of character and event templates that a program could use to fill in the details of a generated high-level plot structure. Unfortunately, the method of constructing those templates seemed ad-hoc rather than automated.

Evaluation metrics were just as hard for the story generation crowd as for the music crowd. The idea of story goals was proposed to help alleviate the story ambiguity, though analogous efforts at decomposing music into emotions seemed to fail. However, a new British music improvisation company by the name of “Jukedeck” seems to have successfully trained neural nets based on desired features, i.e. descriptions such as “calm” or “peaceful”. Other goals such as climax-timing can apparently be generated by the Jukebox nets. This would be an interesting direction for further research.

3. Parsing roadmaps

Figure 1: Distributions of brick usages, untransposed. Brick label feature examples: name: "POT", key: C, variant: var1.

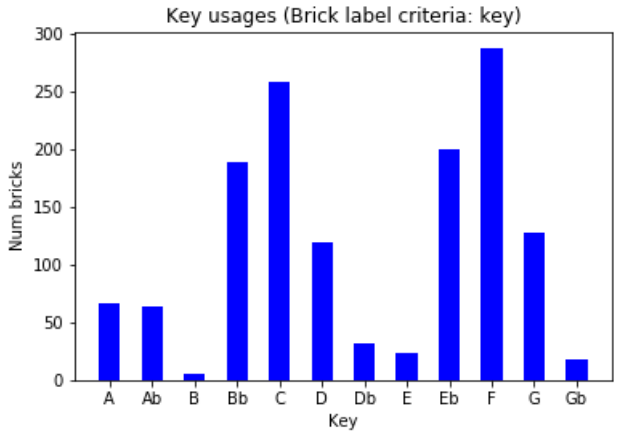


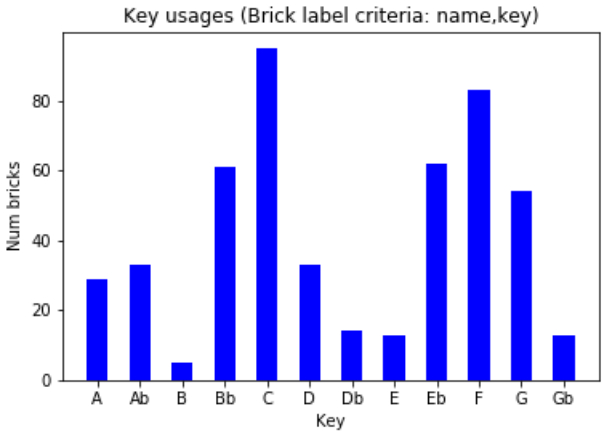
After hitting a dead-end with the story generation techniques, I turned to brick decomposition. The hope was that brick decomposition and prediction could help guide music improvisation at a higher level than note-to-note.

The roadmap and brick corpus was predefined by a collaboration between John Elliot and Prof. Keller. I used a corpus of 100 insight roadmaps, which resulted in $\sim$ 350 to $\sim$ 500 distinct bricks depending on the features considered. After parsing the roadmaps into their respective bricks, 1st-order Markov chaining of the bricks was performed. Given the sparsity of the dataset, the transition probabilities are most certainly not representative of the actual brick usages, but they show a clear preference towards certain bricks nonetheless.

Several graphs showcase the patterns found in the brick usages. Whether or not all the songs were transposed into C, brick usage was very sparse and key usage was unevenly distributed. Most bricks were used at most 3 times. Introducing the key into the label drastically reduces the maximum number of uses, as expected.

Figure 2: Distributions of key usages, untransposed. Brick label feature examples: name: "POT", key: C, variant: var1.





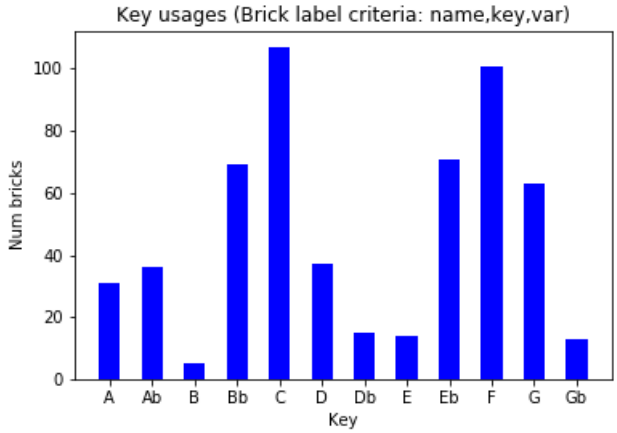


Figure 3: Distributions of brick usages, transposing all songs into C. Brick label feature examples: name: “POT”, key: C, variant: var1.

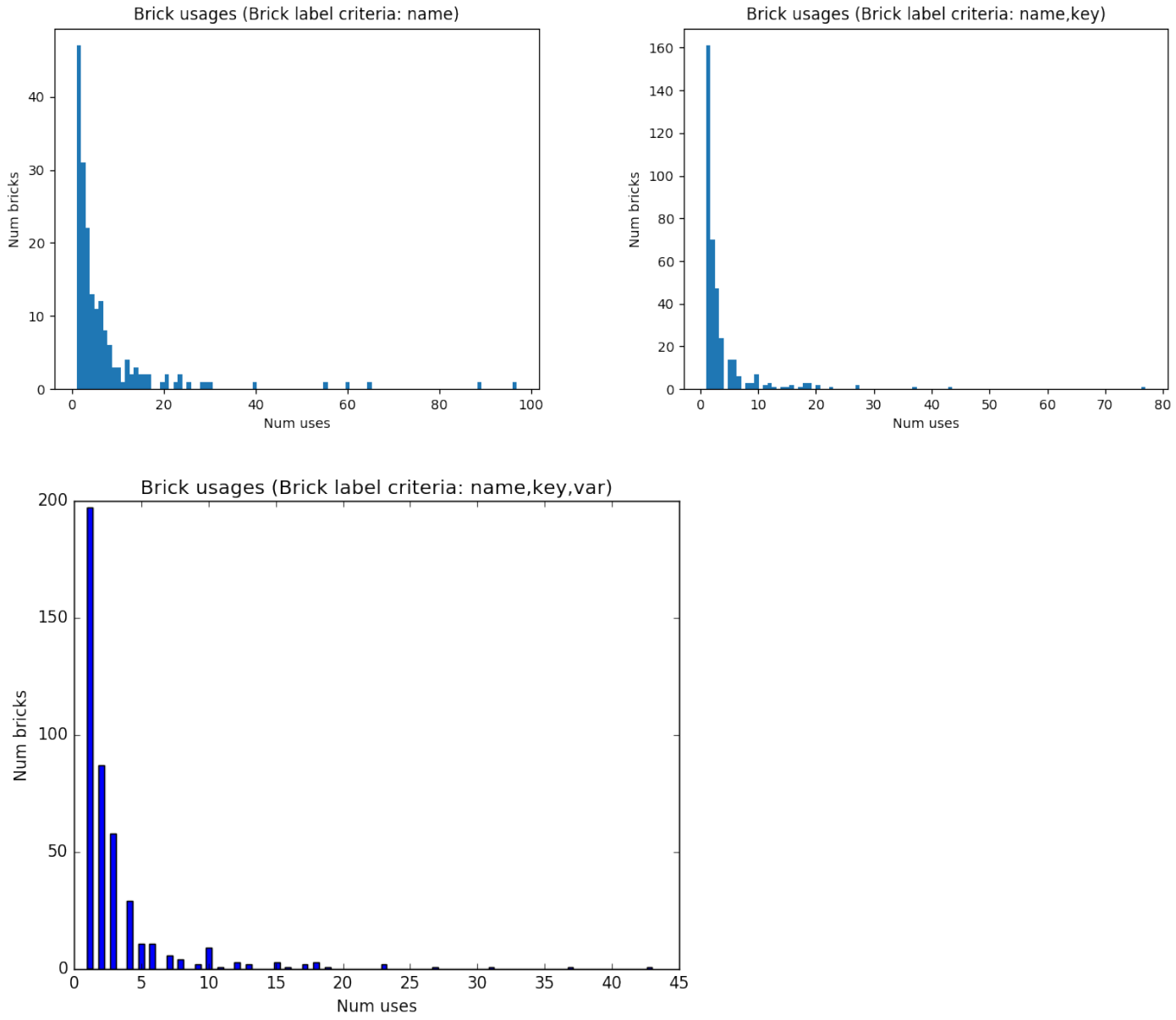
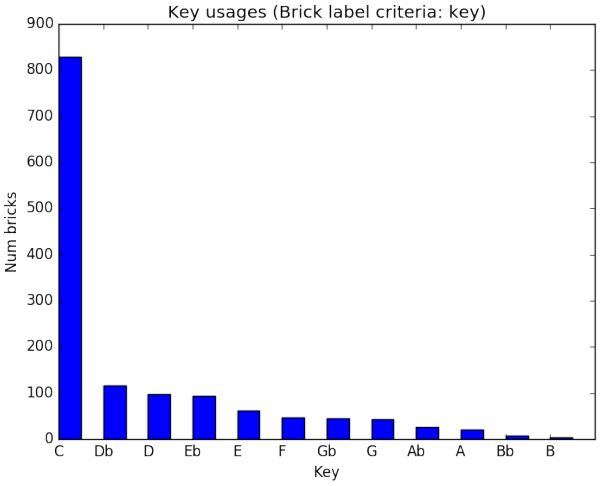
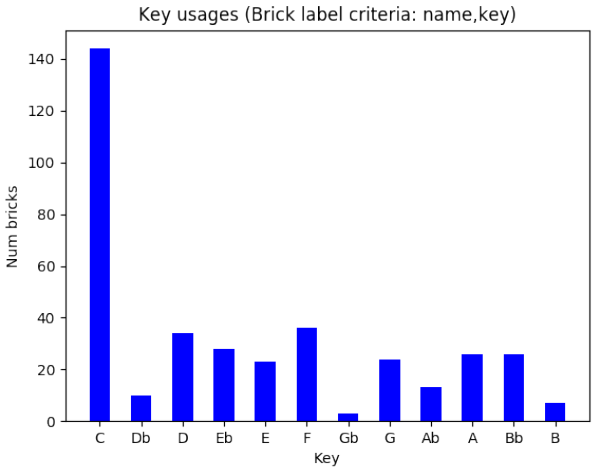
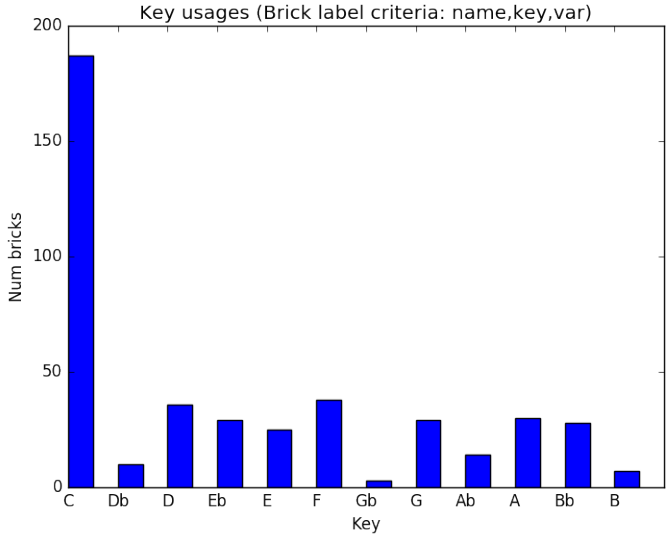


Figure 4: Distributions of key usages, transposing all songs into C. Brick label feature examples: name: “POT”, key: C, variant: var1.







References

[1] R. Keller et. al. A creative improvisational companion based on idiomatic harmonic bricks. *International Conference on Computational Creativity*, 2012.

[2] Ben Kybartas and Rafael Bidarra. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games*, 2016.