A Computational Model of Jazz Improvisation Inspired by Language

Cody Kommers (cydeko@ucla.edu)
Department of Psychology, University of California, Los Angeles
Los Angeles, CA 90095 USA

Alan Yuille (yuille@stat.ucla.edu)
Department of Statistics, University of California, Los Angeles
Los Angeles, CA 90095 USA

Abstract

This paper presents a novel computational model of jazz improvisation based on n-gram language models. Recent functional neuroimaging studies suggest that the brain processes structural elements of improvised music and conversational language in a similar manner. We hypothesized that if musical improvisation and language share a common cognitive and neurological foundation, then statistical techniques for modeling one domain should be capable of successfully modeling the other domain. Accordingly, we demonstrate that n-grams (an archetypal language model) can successfully model jazz improvisation when trained on a large corpus of expert-level jazz saxophone solos. Furthermore, we propose perplexity as a novel method of evaluation of jazz improvisation models.

Keywords: computational models; jazz improvisation; music; language; n-grams

Introduction

Which cognitive faculty does the following list of attributes describe? (1) It is a form of communication; (2) It is governed by rules; (3) It is acquired through experience; and (4) It requires the production of finite strings from infinite possibilities. This is a paradigmatic description of human language, and these attributes are among the characteristics that make language a salient subject for studying the mind. However, these attributes also aptly characterize jazz improvisation (Culicover, 2005).

This paper presents a novel computational model of jazz improvisation using techniques developed for language modeling–specifically, n-grams (Figure 1). The success of the present model, paired with the success of n-gram language models, suggests that the same types of computational architectures can provide basic accounts of production of both improvised music and language. This success provides computational support for the hypothesis that production of musical improvisation relies on the same areas of the brain as language production. This hypothesis, in its strongest form, states that these regions are not domain-specific for language, but domain-general for communication (Donnay et al., 2014).

The hypothesis of a shared neurological foundation between improvised music and language derives from previous functional imaging studies of improvising musicians. Limb and Braun (2008) suggest that the medial prefrontal cortex is recruited to produce narrative structure in both musical improvisation and linguistic improvisation (e.g., conversation or storytelling). Similarly, Berkowitz and Ansari (2008) suggest that the inferior frontal gyrus is recruited to produce syntactic structure in both musical improvisation and linguistic improvisation. Other functional imaging studies of musical improvisation include Liu et al. (2012); Brown, Martinez, and Parsons (2006); and Bengtsson, Csikszentmihalyi, and Ullen (2007). These studies are summarized in Beaty (2015).

The hypothesis of shared cognitive traits between improvised music and language is supported largely by qualitative observation. Johnson-Laird (2002) claims that the best analogy to jazz improvisation is conversation (see also, Johnson-Laird, 1991). Likewise, Culicover (2005) provides a list of similar characteristics between jazz improvisation and conversational language, both of which are: improvised, rule-governed, processed in real-time, creative, acquired through experience, and used for communication. These descriptions are both aligned with the phenomenological accounts of many professional musicians who liken their collaborative improvisation to language or conversation (Berliner, 2009).

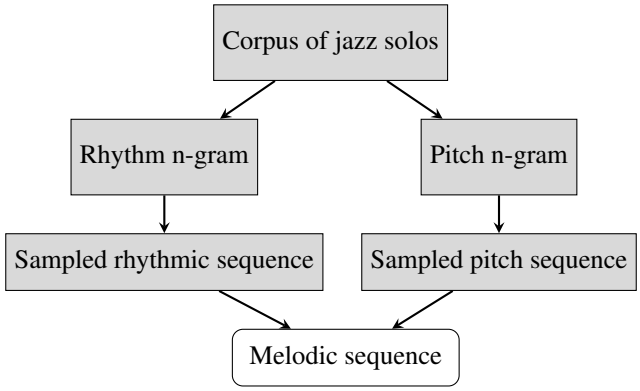

Figure 1: Basic structure of the present model. Melodic sequences are produced by independently sampling rhythms and pitches from respective n-grams. The n-gram probabilities are learned from a corpus of expert jazz saxophone solos; these probabilities reflect the likelihood of note n given the previous $n-1$ notes.

Many researchers have previously attempted to model production of jazz improvisation, because jazz improvisation offers a paradigmatic example of musical improvisation, which is a fascinating cognitive expertise independently of its relation to language. Examples of techniques used in these models include genetic algorithms (Biles, 1994), artificial neural networks (Toiviainen, 1995; Bickerman et al., 2010), grammars (Keller & Morrison, 2007), and tailor-made sets of im-

provisatory formulae (Grachten, 2001). A common advantage among these systems is that they produce novel strings of melodies that qualitatively resemble jazz. A significant disadvantage of these systems is that none are accompanied by a metric for quantifying the success of the produced melodies. Furthermore, many of these models rely on highly domain-specific algorithms, and it is unclear how they might generalize to language or other domains of creativity.

In contrast, the field of modeling human language is far more developed than the field of modeling jazz improvisation. For example, some language models have been shown to achieve human-level performance on tests of analogies (Demski et al., 2015). The archetypal technique for language modeling is characterized by learning an n-gram model from a large corpus of data (Manning, 1999). Intuitively, n-grams capture the statistical likelihood that, in English, if one says "My name" they are likely to follow it with "is" and not "are". In other domains, n-grams are referred to as Markov models or finite-state models.

The application of n-grams to music dates back to Pinkerton (1956) and Cohen (1962). A more recent example of using Markov models to predict music is Conklin (2003). The present model is differentiated from previous statistical models of musical generation namely by (1) its novel corpus of data, without which a model cannot be trained or tested, and (2) its explicit relationship to cognitive and neurological hypotheses.

Thus, with the qualitative and quantitative similarities between improvised music and language, the demonstrated success of n-gram language models, and the comparatively large potential for improvement in jazz improvisation models, it follows naturally to apply modeling techniques developed for language to jazz improvisation.

N-gram Background

The following procedures were used to learn the present n-gram model. These techniques can apply to learning probabilities of strings in the context of both language and music. In language modeling, the goal of an n-gram is to calculate the probability of a given sentence. In musical modeling, the goal of an n-gram is to calculate the probability of a given melodic phrase:

$$P(W) = P(w_1, w_2, ..., w_n) \tag{1}$$

Where W is a sentence or, in the case a music, a melodic phrase and $w_1, ..., w_n$ is the string of words making up the sentence or the string of notes making up the phrase.

This calculation is often framed in the following way; the probability of word n given the previous $n-1$ words:

$$P(w_n|w_1, w_2, ..., w_{n-1}) \tag{2}$$

For instance, a language model may capture that "are" is likely to follow from "My children" in English. Similarly, a musical model may capture that G is likely to follow from E in the key of C.

Ideally, a model would be sophisticated enough to learn the specific probabilities associated with complex strings. However, this is not feasible in practice. To account for this, n-gram models typically make the Markov assumption:

$$P(w_n|w_1, w_2, ..., w_{n-1}) \approx P(w_n|w_{n-1}) \tag{3}$$

The base rate probability $P(w_n)$ is called a unigram. $P(w_n|w_{n-1})$ is called a bigram. $P(w_n|w_{n-2}, w_{n-1})$ is called a trigram. In the present model, these n-gram probabilities were learned using data from a training set with 385,305 note events.

As n increases, the percentage of possible strings that have not been observed increases as well. According to the model, the probabilities of these unseen strings are 0, even if many of them could be legal. To account for these unseen possible strings, we used a method of smoothing called simple linear interpolation. The simple linear interpolation equation for trigrams:

$$\begin{aligned} \hat{P}(w_n|w_{n-1}, w_{n-2}) = {} & \lambda_1 P(w_n|w_{n-1}, w_{n-2}) \\ & + \lambda_2 P(w_n|w_{n-1}) \\ & + \lambda_3 P(w_n) \end{aligned} \tag{4}$$

Where λ_i is a weight for the associated n-gram and $\Sigma_n \lambda_i = 1$. These weights are learned by maximizing the likelihood of strings in a held-out set, which, for the present model, contained 22,665 note events. This interpolated model affords appropriate probability mass to strings which have not yet been seen, but could be in the future.

Perplexity is a metric for evaluating the performance of n-gram models. It is proportional to the probability that the learned n-gram assigns to each string W seen in a test set. The test set for the present model contained 45,300 note events. The perplexity equation for bigrams:

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \tag{5}$$

Where W is a string of words and N is the number of words in string W. For more information on n-grams or language modeling, see Manning (1999).

Language models typically learn and evaluate n-gram probabilities with a large corpus of sentence data. The present jazz improvisation model employs the same methodology by training and testing on a large corpus of appropriate musical phrases.

Data Acquisition Methods

Learning n-gram probabilities for jazz improvisation requires a substantial corpus of data from solos played by expert jazz musicians. To create a corpus of such data, we followed a three step process: (1) download publicly available transcriptions of jazz saxophone solos as portable document format

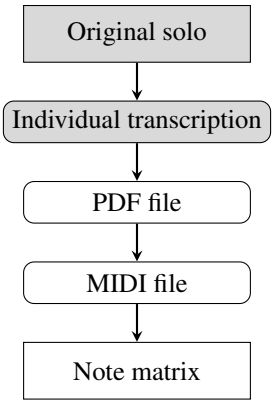

Figure 2: Process of data acquisition from original solo to note matrix. Individual transcriptions of professional saxophone solos were downloaded from an internet database. Conversion from PDF to MIDI to note matrix was done using third-party software.

(PDF) files, (2) transform these PDF files into musical instrument digital interface (MIDI) files, and (3) transform these MIDI files into note matrices for use in MATLAB (Figure 2).

First, we downloaded publicly available transcriptions of jazz saxophone solos as PDF files. Many online resources exist to amass collections of solos, because many neophyte jazz musicians learn their instrument through attempting to play expert solos verbatim, a practice technique referred to as transcription. We used 838 transcribed saxophone solos from the Saxopedia website, which is a wiki-based collection of transcribed jazz solos. These solos represent a comprehensive sample of the most influential jazz saxophonists from the 1940s to 2000s – over 150 musicians, including Stan Getz, Michael Brecker, Dexter Gordon, and Sonny Rollins. These data represent canonical jazz saxophone style and should generalize to other jazz instruments in as much as they are not bound by instrument-specific limitations.

Saxophone solos were chosen as data because (1) the saxophone plays only monophonic melodies, (2) notated transcriptions are most prevalent for saxophone, and (3) written musical notation can more accurately represent the music played by a saxophone than that of a trumpet or trombone.

Second, we used Myriad Software's PDFtoMusic program to transform PDF files into MIDI files. Due to individual discrepancies in notating by the original transcribers and occasional flaws in the transformation process, this transformation introduced intermittent and apparently unsystematic noise into the data, namely as approximations of rhythms (e.g., encoding 0.5 as 0.50001).

Third, we used the MIDI toolbox for MATLAB (Eerola & Toiviainen, 2004) to transform MIDI files into note matrices. MIDI files, unlike waveform-based files, explicitly represent individual characteristics of note events such as duration, pitch, and velocity. We were therefore able to represent distinct attributes (such as pitch and rhythmic duration) as vectors, indexed corresponding to the order in which the notes occurred.

In total, this resulted in 453,409 note events (including start and stop indicators), for which we were especially interested in the vectors for rhythmic duration and pitch.

Results

Rhythmic Duration

Rhythmic duration data were stored sequentially in a vector. Each rhythm is represented as a decimal percentage of a quarter note. For example, an eighth note is represented as 0.5 and a half note is represented as 2.0. Beginnings and endings of songs were indicated by start and stop indices.

Discrepancies in the process of transforming PDF files to note matrices introduced noise into the rhythmic duration data. We used two strategies to account for this noise: (1) numeric representations of rhythmic durations were truncated to four decimal places and (2) rhythmic durations were only included in the model if they occurred at least once out of every 4,500 notes (100 times of out the 453,409 notes in the data set). While some noise may still exist, the effect on the relevant data should be negligible.

With the rounded decimal and base rate threshold, the model includes 47 distinct rhythmic tokens. See Table 1 and Figure 3 for distribution of rhythmic durations.

Table 1: Most common unigram rhythmic durations. Rhythmic durations are displayed as decimal percentage of a quarter note. Forty-seven distinct rhythmic duration tokens were included in the model.

Base rate	Rhythm
57.0%	0.5
18.1%	0.25
5.4%	0.3333
5.9%	1.0
3.4%	0.125
2.3%	0.1667
2.3%	1.5
1.1%	2.0
0.6%	0.6667
0.5%	0.75

Pitch

Pitch data were stored sequentially in a vector. Each pitch was represented as an integer corresponding to its MIDI representation. For example, A0 was represented as 21, A4 was represented as 69, and A7 was represented as 105. Beginnings and endings of songs were indicated by start and stop indices. Pitches data were learned and evaluated without respect to the underlying harmonic structure.

The model included 67 distinct pitch tokens. See Figure 4 for distribution of pitches.

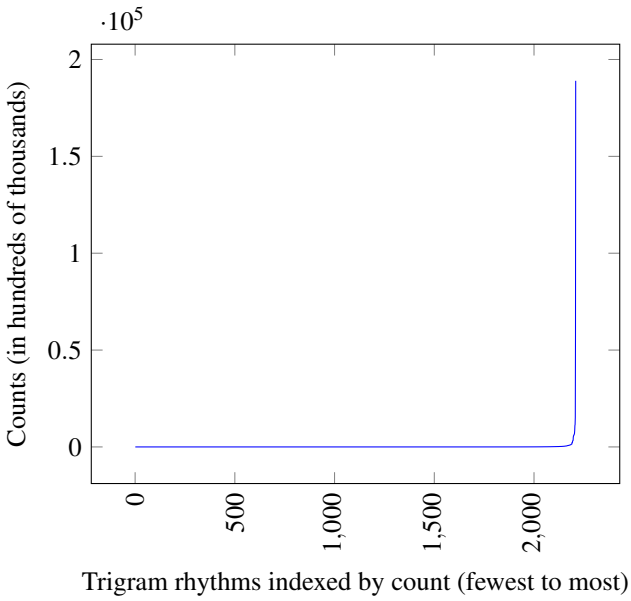

Figure 3: Exponential distribution of trigram rhythmic durations demonstrating well-defined structure of rhythm in jazz improvisation. Out of the 2,209 possible strings of trigram rhythmic durations, four of the sequences account for 70.0% of the rhythmic sequences played. Trigram sequences of pitch follow a similar exponential distribution.

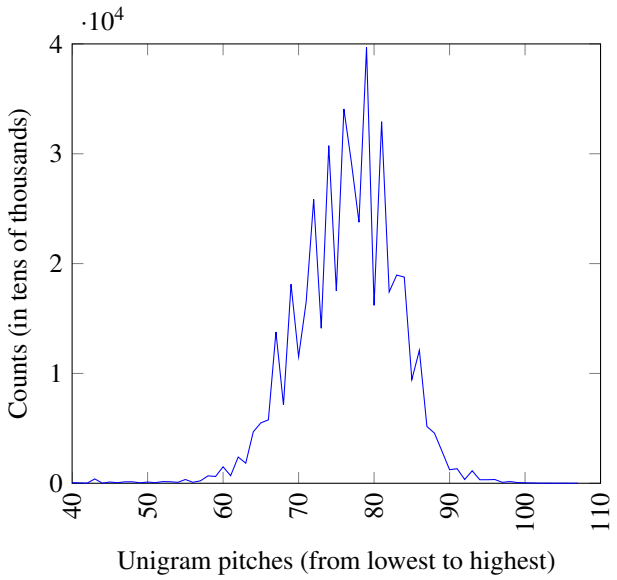

Figure 4: Roughly normal distribution of unigram pitches demonstrating well-defined structure of pitch in jazz improvisation. Spikes between neighbors are likely due to some keys being more common than others. For example, the most commonly played pitch, G5, is within the key of Eb, Ab, Bb, G, and C (the most common jazz keys); however, its neighbor, Ab5, is much less common because it is within the scale of Eb and Ab, but not Bb, G, or F.

Model of combined rhythm and pitch

This model combines rhythmic duration and pitch after learning n-gram probabilities, making the assumption that rhythmic duration and pitch are essentially independent (Figure 1). The effects of this assumption should be negligible (e.g., an F is not more likely to be an eighth note by virtue of being an F and not an F#). However, there may be intermittent cases where this assumption is incorrect. For example, if one phrase is commonly played by many musicians with the exact same rhythmic and melodic sequence, then the model will not capture this correspondence.

The computation performed by the model is simple: Use learned n-gram probabilities to (1) produce a string of rhythmic durations and (2) populate these rhythmic durations with pitches. A seed rhythmic duration and pitch are sampled from respective base rate probabilities.

Perplexity

We propose that models of jazz improvisation can be quantitatively evaluated based on how closely the model-produced melodies match with melodies produced by expert musicians. The same principle is used to evaluate language models, where a successful model is one that generates phrases closely resembling phrases produced by a native speaker. A basic calculation of the match between model and expert is perplexity (Equation 5).

In the present model, lowest (i.e., best) evaluated perplexity was 13.29 for pitch and 2.91 for rhythmic duration (Table 2). In comparison, Brown et al. (1992) learned a trigram language model that obtained a perplexity of 244 for a representative sample of English, containing 44,177 distinct tokens; Bengio et al. (2006) learned a 5-gram model that obtained a perplexity of 252 on the same sample. Shannon (1951) demonstrated a trigram model obtains a perplexity of 9.5 on predicting English letters (26 distinct tokens).

Discussion

This model provides a strong domain-general basis for modeling jazz improvisation. More domain-specific structure will be needed to improve the model to the point of human-level improvisation. Crucially, the model must account for both harmonic structure (i.e., key and chord changes) and rhythmic structure (i.e., tempo and rhythmic style).

Harmonic structure is a crucial aspect of jazz improvisation. While a melody can still incorporate pitches that are not within the key being played, the likelihood of a given pitch depends significantly on the chords played by the accompaniment. An account of harmonic structure will require domain-specific knowledge about chords to incorporate statistical dependencies between pitch and harmony into the model. For example, the n-gram will need to account for the chord over which the current string of notes is being played. In practice, this should be feasible to incorporate into future models, but will require a transition from resolute domain-generality to

Table 2: Perplexity evaluation results. Ability to predict expert melodic phrases increases with n, with the most dramatic increase between unigram and bigram.

Attribute	N-gram	Set	Perplexity
pitch	1	test	31.88
pitch	1	train	34.15
pitch	2	test	14.61
pitch	2	train	13.38
pitch	3	test	13.92
pitch	3	train	12.18
pitch	4	test	**13.29**
pitch	4	train	10.49
rhythm	1	test	7.25
rhythm	1	train	8.34
rhythm	2	test	2.96
rhythm	2	train	2.86
rhythm	3	test	**2.91**
rhythm	3	train	2.78

incorporation of domain-specific knowledge.

It should be noted that though the model does not incorporate knowledge about harmony, it is still able to account for a significant amount of pitch structure in melodies. Just by incorporating a dependency on the previous pitch, the model can implicitly infer harmonic information. A model that incorporates knowledge about harmony will be able to account for even more melodic structure in jazz improvisation (i.e., result in a lower perplexity).

Rhythmic structure is another crucial aspect of jazz improvisation. Not only do musicians need to be playing at the same tempo, they also must agree on a rhythmic style. For instance, if a song is played with swung eighth notes, it will induce a different rhythmic paradigm than if a song is played with straight eighth notes. Musicians refer to this differentiation as rhythmic "feel". Incorporating rhythmic feel into the model will require an independent computational model of the way in which rhythms are played by jazz musicians.

A considerable issue in all existing models of jazz improvisation, including the present one, is the lack an account for the global scope of a solo, called compositional intentionality. Compositional intentionality captures the resemblance of a solo to a narrative story, which has specific characteristics of exposition, rising action, and resolution. Current models produce melodies without respect to placement in the greater context a solo. Future models should incorporate context-dependent facets such as idiomatic repetition of melodic phrases and melody complexity as a function of solo context (e.g., simpler phrases initially, increasing in complexity as solo progresses).

An idealized model, accounting for harmonic, rhythmic, and global structure, is outlined in Figure 5.

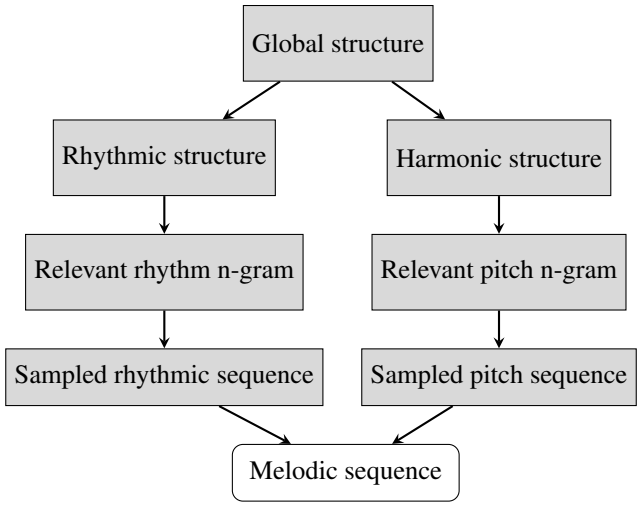

Figure 5: Basic structure of an idealized model. Global structure of the current solo informs local considerations of rhythmic structure and harmonic structure. In turn, these considerations allow for sampling respective rhythmic and pitch sequences from the appropriate n-grams to create a melodic sequence. A model with these contingencies would approximate a comprehensive account of jazz improvisation.

Future Methods of Evaluation

Future methods of evaluation will require comparing competency of predicting expert-level jazz improvisation between models. This is analogous to how language models are evaluated. The better model is the one that performs more competently on a given task. The difficulty in this method of evaluation for jazz modeling will be aligning each model in a common computational paradigm.

Another future method of evaluation could take the form of an adapted Turing Test (Turing, 1950): Given all non-improvisational variables equal, can a human judge distinguish between music from a model of jazz improvisation and music from an expert musician? If musical samples from models and humans were compared with equivalent timbre, rhythmic density, and melodic range, then only the variable of idiomatic improvisation would remain. Of course, this could be considered the ultimate goal of jazz modeling; no existing model would likely pass this test. However, an adapted Turing Test would be a sound method of evaluation in principle.

Domain-general Implications

While understanding jazz improvisation may be a worthwhile goal in itself, the further reaching considerations are that of domain-general communication and creativity.

Communication Just as the present techniques for modeling jazz improvisation are inspired by techniques for modeling language, perhaps future advances developed for modeling jazz may impact language modeling in return. In many respects, jazz improvisation seems to be a more tractable prob-

lem than language (e.g., smaller lexicon of possible notes than possible words). Thus, if significant advances are made in modeling jazz improvisation (a similar, but substantially easier problem), then these advances may lend translational insight to language modeling.

Creativity Jazz improvisation is a paradigmatic example of creative expression under well-defined constraints. Further developments in modeling jazz improvisation may yield understanding about the interaction between expression and constraint in creativity. Most notably, intriguing implications could result from the interaction between compositional intentionality (accounting for the global structure of a solo) and development of novel musical strings: how does an artist produce creative entities that exist in the context of a whole work?

Computationally modeling jazz improvisation is perhaps most salient because it represents a culminating goal of cognitive science and artificial intelligence: Creating something that is itself creative.

Acknowledgments

This research was funded by the Undergraduate Research Scholars Program at University of California, Los Angeles.

References

Beaty, R. E. (2015). The neuroscience of musical improvisation. *Neuroscience & Biobehavioral Reviews (in press)*.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2006). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137–1155.

Bengtsson, S., Csikszentmihalyi, M., & Ullen, F. (2007). Cortical regions involved in the generation of musical structures during improvisation in pianists. *Journal of Cognitive Neuroscience*, *19*(5), 830–842.

Berkowitz, A., & Ansari, D. (2008). Generation of novel motor sequences: The neural correlates of musical improvisation. *NeuroImage*, *41*, 535–543.

Berliner, P. F. (2009). *Thinking in jazz: The infinite art of improvisation*. University of Chicago Press.

Bickerman, G., Swire, P., Bosley, S., & Keller, R. (2010). Learning to create jazz melodies using deep belief nets. In *Proceedings of the international conference on computational creativity* (pp. 228–237). Lisbon, Portugal.

Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1994 international computer music conference* (pp. 131–137). Aarhus, Denmark.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Comput. Linguist.*, *18*(4), 467–479.

Brown, S., Martinez, M., & Parsons, L. (2006). Music and language side by side in the brain: a pet study of the generation of melodies and sentences. *European Journal of Neuroscience*, *23*, 2791–2803.

Cohen, J. (1962). Information theory and music. *Behavioral Science*, *7*(2), 137–163.

Conklin, D. (2003). Music generation from statistical models. In *Proceedings of the artificial intelligence and simulated behavior 2003 symposium on artificial intelligence and creativity in the arts and sciences* (p. 30-35). Aberystwyth, Wales.

Culicover, P. (2005). Linguistics, cognitive science, and all that jazz. *The Linguistic Review*, *22*, 227–248.

Demski, A., Ustun, V., Rosenbloom, P., & Kommers, C. (2015). Outperforming word2vec on analogy tasks with random projections. In *Proceedings of the 2015 international conference on learning representations (submitted)*. San Diego, California.

Donnay, G., Rankin, S., Lopez-Gonzalez, M., Jiradejvong, P., & Limb, C. (2014). Neural substrates of interactive musical improvisation: An fmri study of trading fours in jazz. *PLOS ONE*, *9*(2).

Eerola, T., & Toiviainen, P. (2004). *Midi toolbox: Matlab tools for music research*. Jyväskylä, Finland: University of Jyväskylä.

Gratchen, M. (2001). Jig: jazz improvisation generator. In *Workshop on curr. research dir. in comp. music* (pp. 1–6).

Johnson-Laird, P. (2002). How jazz musicians improvise. *Music Perception: An Interdisciplinary Journal*, *19*, 415-442.

Johnson-Laird, P. N. (1991). Jazz improvisation: A theory at the computational level. *Representing musical structure, London*, 291–325.

Keller, R., & Morrison, D. (2007). A grammatical approach to automatic improvisation. In *Proceedings of the 4th sound and music computing conference* (pp. 330–337). Lefkada, Greece.

Limb, C., & Braun, A. (2008). Neural substrates of spontaneous musical performance: An fmri study of jazz improvisation. *PLOS ONE*, *3*(2).

Liu, S., Chow, H., Xu, Y., Erkkinen, M., Swett, K., Eagle, M., ... Braun, A. (2012). Neural correlates of lyrical improvisation: An fmri study of freestyle rap. *Nature, Scientific Reports*, *2*(834).

Manning, C. (1999). *Foundations of statistical natural language processing* (H. Schutze, Ed.). MIT Press.

Pinkerton, R. (1956). Information theory and melody. *Scientific American*, *194*(2), 77–86.

Shannon, C. (1951). Prediction and entropy of printed english. *Bell system technical journal*, *20*(1), 50–64.

Toiviainen, P. (1995). Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. *Music Perception: An Interdisciplinary Journal*, *12*(4), 399–413.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), pp. 433-460.