Preprint accepted for publication in Neural Computing and Applications, Springer

From Context to Concept: Exploring Semantic Relationships in Music with Word2Vec

Ching-Hua Chuan · Kat Agres · Dorien Herremans

Received: 22/06/2018 / Accepted: 26/11/2018

Abstract We explore the potential of a popular distributional semantics vector space model, *word2vec*, for capturing meaningful relationships in ecological (complex polyphonic) music. More precisely, the skip-gram version of word2vec is used to model slices of music from a large corpus spanning eight musical genres. In this newly learned vector space, a metric based on cosine distance is able to distinguish between functional chord relationships, as well as harmonic associations in the music. Evidence, based on cosine distance between chord-pair vectors, suggests that an implicit circle-of-fifths exists in the vector space. In addition, a comparison between pieces in different keys reveals that key relationships are represented in word2vec space. These results suggest that the newly learned embedded vector representation does in fact capture tonal and harmonic characteristics of music, without receiving explicit information about the musical content of the constituent slices. In order to investigate whether proximity in the discovered space of embeddings is indicative of 'semantically-related' slices, we explore a music generation task, by automatically replacing existing slices from a given piece of music with new slices. We propose an algorithm to find substitute slices based on spatial proximity and the pitch class distribution inferred in the chosen subspace. The results indicate that the size of the subspace used has a significant effect on whether slices belonging to the same key are

Ching-Hua Chuan
Department of Cinema & Interactive Media, School of Communication, University of Miami, USA
Tel.: +1-305-2844388
Fax: +1-305-2845226
E-mail: c.chuan@miami.edu

Kat Agres
Social and Cognitive Computing Department, Institute for High Performance Computing, A*STAR, Singapore

Dorien Herremans
Information Systems, Technology, and Design Department, Singapore University of Technology and Design, Singapore
Social and Cognitive Computing Department, Institute for High Performance Computing, A*STAR, Singapore

selected. In sum, the proposed word2vec model is able to learn music-vector embeddings that capture meaningful tonal and harmonic relationships in music, thereby providing a useful tool for exploring musical properties and comparisons across pieces, as a potential input representation for deep learning models, and as a music generation device.

Keywords word2vec · music · semantic vector space model

1 Introduction

Distributional semantic vector space models have become important modeling tools in computational linguistics and natural language processing (NLP) [38, 61]. These models are typically used to represent (or embed) words as vectors in a continuous, multi-dimensional vector space, in which geometrical relationships between word vectors are significant [3, 38, 40, 61]. For example, semantically similar words tend to occur in close proximity within the space, and analogical relationships may be discovered in some cases as vectors with similar angle and distance properties [61]. A popular approach to creating vector spaces for NLP is a neural network-based approach called *word2vec* [44]. In this paper, we explore whether word2vec can be used to model a related form of auditory expression: music. We build upon the previous work of the authors [24], in which a preliminary model was built by training word2vec on a small music dataset. This research takes a more comprehensive approach to exploring the extent to which word2vec is capable of discovering different types of meaningful features of music, such as tonal and harmonic relationships.

In the field of music cognition, there is a long tradition of investigating how the statistical properties of music influence listeners' perceptual and emotional responses. For example, the frequency of occurrence of tones in a key helps shape the perception of the tonal hierarchy (e.g., the relative stability of different notes in the key) [33], and the likelihood that a particular tone or chord will follow a previous tone(s) or chord(s) helps drive tonal learning and expectation mechanisms [54, 49, 2], as well as emotional responses to music [28, 41]. Researchers have employed various methods to capture the statistical properties of music using machine learning techniques, including Markov models [16], Recursive Neural Networks (RNNs) combined with Restricted Bolzmann Machines [7], and Long-Short Term RNN models [8, 13, 18, 55]. Notably, these models all use input representations that contain explicit information about musical structure (e.g. pitch intervals, chords, keys, etc). In the present research, rather than relying upon this sort of explicit musical content (e.g., defining the meaning inherent in particular notes and chords, such as their relative tonal stability values), we use distributional semantics techniques to examine how *contextual* information (e.g., the surrounding set of musical events) may be used to discover musical meaning. The premise of this approach is that examining the surrounding context and co-occurrence statistics of musical events can yield insight into the semantics (in this case, musical relationships) of the domain.

In NLP research employing deep learning, vector space models such as word2vec play an important role, because they provide a dense vector representation of words.

In this research, we explore the use of word2vec in the context of music. Computational models of music often use convolutional neural networks (CNNs)[13, 32], which are renowned for their ability to ‘learn’ features automatically [30, 52, 53]. These CNNs are typically combined with other models, such as memory-based neural networks like LSTMs [13] and RNNs [7], thus providing an end-to-end solution without the need for hand-crafted features. Therefore, the word2vec representation proposed in this research may provide an alternative input for RNNs and LSTMs in the domain of music.

Few researchers have attempted to model musical context using semantic vector space models. Huang et al [27] used word2vec to model chord progressions as a method of discovering harmonic relationships and recommending chords to novice composers. Madjiheurem et al [39] also trained NLP-inspired models for chord sequences. In their preliminary work, only training results were compared, not the actual ability of the models to capture musical properties. Using chords as input, however, gives the model explicit musical content, and vastly reduces the extensive space of all musical possibilities to a very small vocabulary. In this paper, complex polyphonic music is represented as a sequence of ‘slices’ that have undergone no preprocessing to extract musical features, such as key signature or time signature. Our goal is to explore the extent to which word2vec is able to discover semantic similarity and musical relationships by looking only at the *context* in which every musical slice appears. We might adapt the famous saying by linguist J. R. Firth [20] from, “You shall know a word by the company it keeps!” to “You shall know a slice of music by the company it keeps”, where a slice of music is a note or chord within a particular time interval.

The structure of the paper will be as follows: the next sections describe the implementation of the word2vec model, followed by the way in which the music is represented in this study. We then report on the model’s ability to capture tonal relationships, and then empirically test the model’s ability to capture key and harmonic relationships (including analogical relationships across keys). The results section discusses the use of the model for music generation, and is followed by the conclusion.

2 Word2vec

The SMART document information retrieval system by Salton [56] and Salton et al [57] was the first to utilize a vector space model. In this initial work, each document in a collection was represented as a dot in a multidimensional vector space. A user query was also represented as a dot in this same space. The query results included the closest documents to the query word in the vector space, as they were assumed to have similar meaning. This initial model was developed through building a frequency matrix. Traditional methods for creating vector space models typically take a bag-of-words approach [58] and create a vector representation for each word w_i based on the co-occurrence frequency with each other word w_j. This initial representation can then be post-processed using methods such as Positive Pointwise Mutual Information [19, 40], normalization, and dimensionality reduction [17, 35]. Another popular model is called GloVe [50], short for Global Vectors. This model also starts from a co-occurrence matrix and then trains a log-bilinear regression model to generate embeddings on the non-zero elements. Subsequently, in the early 2000s, an interest emerged in using neural network techniques such as word2vec for building vector space models [5, 14, 15, 45, 46, 47].

In this research, we work with word2vec [44], which provides an efficient way to create semantic vector spaces. A simple neural network model is trained on a corpus of text, in order to quickly create a vector space that can easily consist of several hundred dimensions [42]. In this multi-dimensional space, each word that occurs in the corpus can be represented as a vector. This semantic vector space reflects the distributional semantics hypothesis, which states that words that appear in the same contexts tend to have similar meanings [23]. In terms of vector spaces, this means that words that occur in the same contexts appear geometrically close to one another. This allows us to assess semantic similarity of words based on geometric distance metrics such as cosine distance.

Skip-gram with negative sampling Two types of approaches are typically used when building word2vec models: continuous bag-of-words (CBOW) and skip-gram. With a CBOW approach, the surrounding words are used to predict the current word [43], while the skip gram-model takes a current word and tries to predict the words in a surrounding window of size c (see Figure 1).

Both models are computationally efficient and have low complexity, which means that they can both handle a corpus of billions of words without much effort. Mikolov et al [42] states that while CBOW models are often slightly faster, skip-gram models tend to perform better on smaller datasets. In this research, we used the latter approach for the dataset described in Section 4.1.

In the current implementation, the neural network takes one word w_t as input and tries to predict one of the surrounding words w_{t+i}. The input-output pairs are randomly sampled k-times by selecting a random index value i from the full skip-gram window c. All of the sampled values for i for one input word w_t form the set J. We can thus define the training objective as:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{\forall i \in J}\log p(w_{t+i}|w_t), \qquad (1)$$

A softmax function is used to calculate term $p(w_{t+i}|w_t)$, however, it is computationally expensive to calculate the gradient of this term. Techniques that allow this problem to be avoided include noise contrastive estimation [22] and negative sampling [44]. In this research we implement the latter.

Negative sampling builds upon the idea that a trained model should be able to differentiate data from noise [21]. Based on this assumption, the training objective is approximated by formulating a more efficient implementation in which a binary logistic regression is used to classify real data versus noise samples. The objective is to maximize the assigned probability of real words and minimize that of noise samples [44].

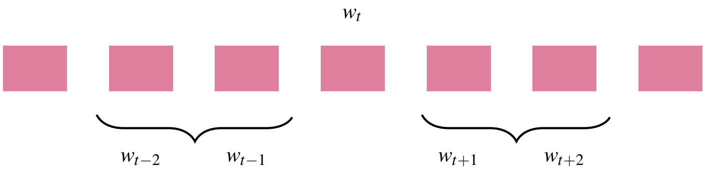


Fig. 1: Example of how a skip gram window of size $c = 4$ is determined on word w_t given a sequence of words.

Cosine distance D_s(A, B) is used to determine the distance between two words, represented by non-zero vectors A and B, in an n-dimensional vector space. The angle between A and B is defined as θ. The cosine similarity metric can be calculated as follows:

$$\mathrm{D}_c(\mathrm{A, B}) = 1 - cos(\theta) = 1 - \mathrm{D}_s(\mathrm{A, B}) \tag{2}$$

where D_s(A, B) refers to *cosine similarity*, a term often used to indicate the complement of cosine distance [59]:

$$\mathrm{D}_s(\mathrm{A, B}) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3}$$

The following section describes the representation that we used to port the word2vec model to the domain of music.

3 Musical slices as words

Although music lacks the precise referential semantics of language, there are clear similarities between these two expressive domains. Like language, music contains sequential events that are combined or constrained according to what can be thought of as a set of grammatical rules (although these rules are, of course, often more flexible for music than language). In addition, both domains are structured in such a way that they generate *expectations* about forthcoming events (words or chords, etc). Unexpected words or notes have both been shown to elicit “unexpectedness” responses in the brain, such as the N400 or early right anterior negativity (ERAN) event-related potentials (ERPs) [6, 31]. In language, grammatical rules place constraints upon *word* order. In music, the style and music-theoretic rules influence the order of *musical events*, such as tones or chords. Unlike language, of course, multiple musical elements can be sounded simultaneously, and may even convey similar semantics (for example, instead of a C major chord containing the pitches of C, E and G, a high C one octave above the lower C may be added, without altering the chord’s classification as C major). Because of this feature of music, in which multiple events may be presented simultaneously in addition to sequentially, we propose for the purpose of this research that the smallest unit of naturalistic music is the ‘*musical slice*’ - a brief duration (contingent on the time signature, musical density, etc) in which one or more tones may occur.

In order to investigate the extent to which word2vec is able to model musical context, and how much *semantic meaning in music* can be learned from the context, polyphonic music is represented with as little added musical knowledge as possible. We segment each piece in the corpus into equal-length slices, each consisting of a list of pitch classes contained within that slice. More specifically, each piece is segmented into beat-long, non-overlapping slices (note that different pieces can have different beat durations). The duration of these slices is based on the beat of the piece, as estimated by the MIDI toolbox [60]. In addition, pitches are not labeled as chords. Instead, all pitches, including chord tones, non-chord tones, and ornaments, are recorded within the slice. Also, pieces are not transposed to the same key because one of the goals of this research is to explore if word2vec can learn the concept of musical key. The only musical knowledge used is octave equivalence: the pitches in each slice are converted into pitch classes. For example, both the pitch C_4 and C_5 are converted to the pitch class C.

Figure 2 shows an example of how slices are encoded to represent a polyphonic music piece. The figure includes the first six bars of Chopin's Mazurka Op. 67 No. 4 and the first three slices for the encoding. Since a slice is a beat long, the first slice contains E, the pitch class for pitch E_5 in the quarter note. The second slice contains E and A because of pitches E_5 and A_3 in the second beat. Note that we include E in the second slice even though pitch E_5 is a tie from the first beat (not an onset) but it is still sounded in the second beat. Similarly, since the third beat contains pitches E_3, A_3, E_4, E_5 (from the dotted tie), and F_5, the third slice includes pitch classes E, A, and F.

The example in Figure 2 can also be used to explain the choice of beat as the slice duration. If the slice is longer than a beat, we may lose nuances in pitch and chord changes. In contrast, if the slice is shorter than a beat, we may have too many repetitive slices (where the content between slices is the same). Finding the optimal setting for the duration of the slice is out of the scope of this paper, however more research is warranted on this topic.

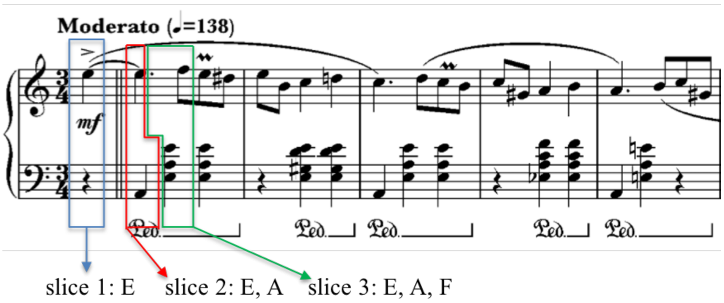


Fig. 2: Representing polyphonic music as a sequence of slices.

In the next section, we describe a number of experiments that were conducted in order to examine whether the word2vec model can capture meaningful musical relationships.

4 Experimental validation

A number of experiments were set up in order to determine how well the geometry of a vector space model, built on a mixed-genre dataset of music, reflects the tonal musical characteristics of this dataset.

4.1 Dataset and model parameters

Machine learning research that involves training models on musical corpora [7] often makes use of very specialized mono-genre datasets such as Musedata from CCARH[1], JSB chorales [4], classical piano archives[2] [51], and Nottingham folk tune collection[3]. In this work, we aim to capture a large semantic vector space across genres. We therefore use a MIDI dataset that contains a mix of popular and classical pieces[4]. This midi collection contains a total of around 130,000 pieces, from a total of eight different genres (classical, metal, folk, etc) and is referred to as "the largest MIDI dataset on the Internet"[5]. From this dataset, we used only the pieces with a genre label, in order to avoid lesser quality files. This resulted in a final dataset consisting of 23,178 pieces.

Inspired by the existing literature on NLP models [40, 50], in which only the most frequently occurring word types in the text are included in the model's vocabulary, we trained our model using only the 500 most occurring musical words (slices) out of a total of 4,076 unique slices. Infrequently occurring words were replaced with a dummy word ('UNK'). This cutoff ensured that the most frequently occurring words were included, as depicted in Figure 3. By reducing the number of words in our vocabulary and thus removing rare words, we were able to augment the accuracy of the model, as the included words occur multiple times in the training dataset. Figure 4 shows that lower training losses can be achieved when reducing the vocabulary size.

A number of parameters were set using trial-and-error, including learning rate (0.1), skip window size (4), number of training steps (1,000,000), and number of dimensions of the model (256). More details on the selection of parameters can be found in [24].

[1] http://musedata.org

[2] http://piano-midi.de

[3] http://ifdo.ca/seymour/nottingham/nottingham.html

[4] https://www.reddit.com/r/datasets/comments/3akhxy/the_largest_midi_collection_on_the_internet

[5] http://stoneyroads.com/2015/06/behold-the-worlds-biggest-midi-collection-on-the-internet/

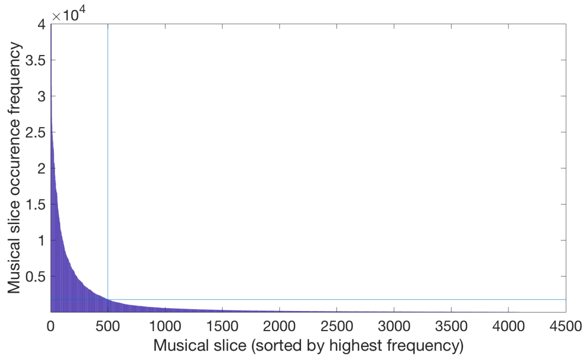


Fig. 3: Frequencies of each musical slice in the corpus. The blue grid lines mark the vocabulary size cutoff of 500 slices.

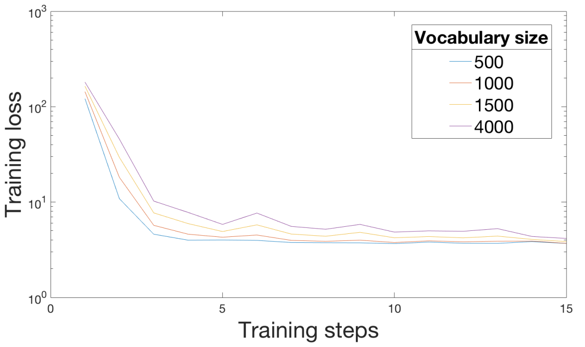


Fig. 4: Evolution of the average loss during training with varying vocabulary size. A step represents 2,000 training batches.

4.2 Context to concept: Chords

The first musical concept that we examine is related to chords. Chords are a set of pitches that sound together and form the building blocks of harmonization. We investigate if word2vec is capable of learning the concept of harmonic relationships by examining the distance between slices that represent different chords. The goal is to see if the distance between chords in word2vec space reflects the functional roles of chords in music theory. It should be noted that, while we are studying triads in this particular experimental context, the encoding can easily handle more complex slices (from a single note to many simultaneous notes), as shown in Section 3.

In tonal music, chords are often labeled with Roman numerals in order to indicate their functional role and scale degree in a given key. For example, the C major triad (tonic) is labeled as I in the key of C major, as it is the tonic triad of the key. Triads such as G major (V, dominant), F major (IV, subdominant), and A minor (vi, relative minor) are considered common chords and closely related to the tonic triad in the key of C major. Others such as E^{b} major (III^{b}), D^{b} major, (II^{b}), and G minor (v) are less associated with I in the key of C major.

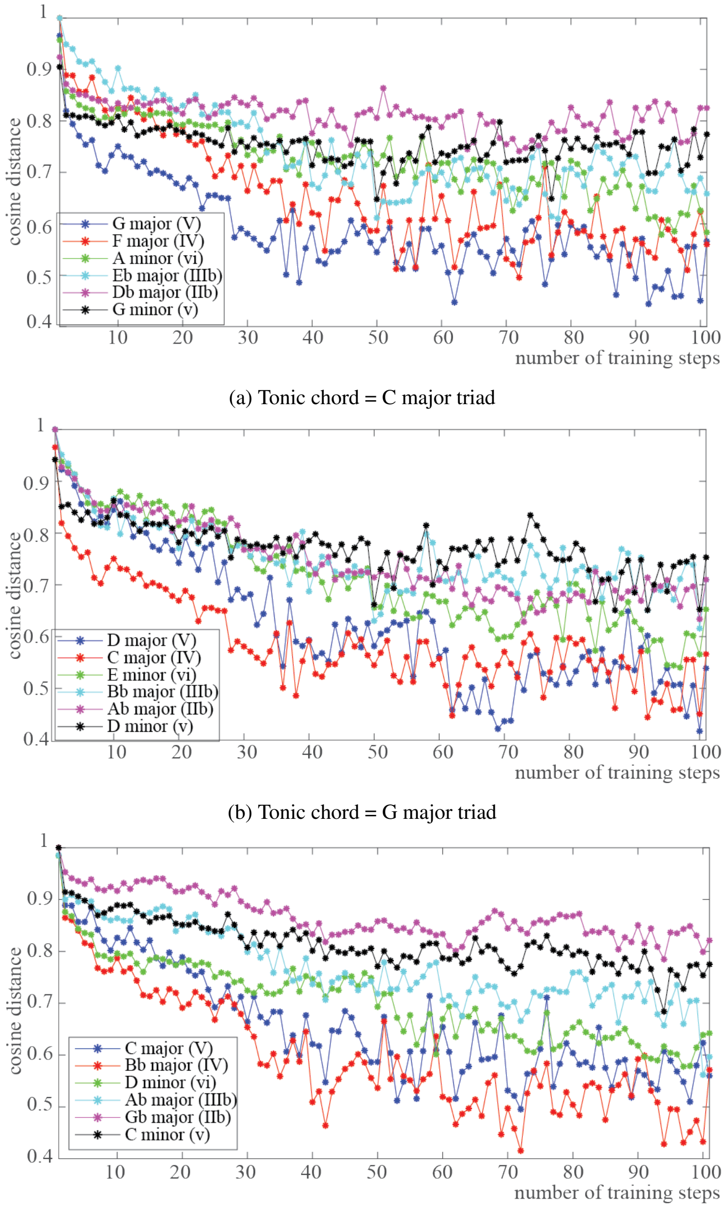
In this study, we examined the distance between the tonic triad (I) and other triads including V, IV, vi, III^{b}, II^{b}, and v in three different keys. To calculate the distance between chords, we first mapped the chord to its pitch classes to find the corresponding musical slice. The geometrical position of this slice was then retrieved in the learned word2vec vector space. We then calculated the cosine distance between the word2vec vectors of the pair of chords.

Figure 5 shows the distance between (a) C major, (b) G major, and (c) F major triads. Generally, one can observe that the distances from a I triad to V, IV, and vi are smaller than those to III^{b}, II^{b}, and v. This finding is promising because it confirms the chordal relationships in tonal music theory [33, 36], and thus shows that word2vec embeddings are able to capture meaningful relationships between chords.

4.3 Context to concept: Keys

In this section we examine whether the concept of a musical key is learned in the music word2vec space. In music theory, the relationships between keys in tonal music are represented in the circle-of-fifths [37]. In this circle, every key is a fifth apart from its adjacent neighbor, and therefore shares all pitch classes but one with the adjacent key. For example, the key of G major is adjacent to (a fifth above) C major, and contains all of the pitch classes of C major's diatonic key signature, except for including an F$^{\#}$ instead of an F. When following the circle-of-fifths clockwise from a given key, the farther one is from the original key, the more different pitch classes are present. The concept of the circle-of-fifths is important in music, and has appeared in the representations or output of various machine learning models such as Restricted Boltzmann Machines [9], convolutional deep models [32], and to some extent, RNNs [12]. In this study, we examine whether the distance between keys in the learned word2vec space of musical slices reflects the geometrical key relationships of the circle-of-fifths.

We chose to experiment with Bach's Well-Tempered Clavier (WTC)'s 24 preludes, as it features a piece in each of the 12 major and 12 minor keys. The Well-Tempered Clavier is a popular music collection for computational study of tonality in music [1, 8, 10, 34, 48]. Each piece in the WTC was transposed to each of the other 11 major or minor keys, depending on whether the original piece is major or minor. Our augmented dataset therefore included 12 versions of the same piece, one for each key. For each piece in the resulting augmented dataset, the musical slices were mapped to the learned music word2vec space (trained on the full corpus as discussed in Section 4.1). The k-means algorithm was used to identify the centroid of the slices for each piece. We used this centroid as the reference point for each piece in the dataset, and calculated the cosine distance between centroids as a metric of the distance between keys in the word2vec space. By comparing the centroid of a piece to the centroid of its own transposed version, one can be sure that the distance between the centroids is only affected by one factor: musical key.



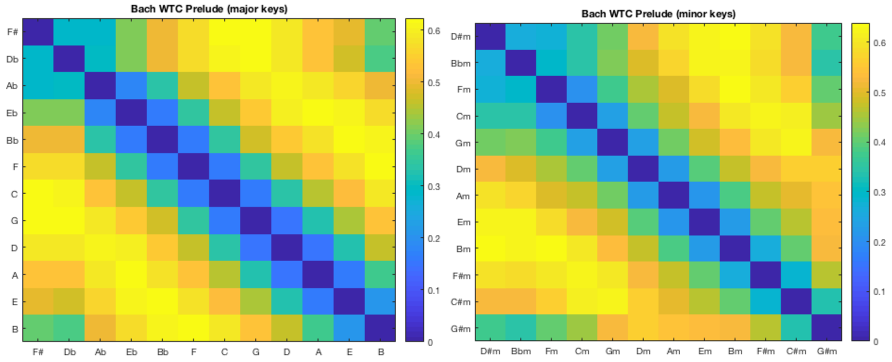
(a) Tonic chord = C major triad

(b) Tonic chord = G major triad

(c) Tonic chord = F major triad

Fig. 5: Cosine distance between slices that represent a tonic chord (C, G, or F major) and its related functional chords. A training step represents 10,000 training batch examples.

Figure 6 shows the average cosine distance between pairs of centroids of pieces in different keys in the learned word2vec space. Both axes list the keys in the same order as they occur in the circle-of-fifths. Based on music theory, we expect this similarity matrix to be blue/green near the diagonal (low distance) where we find identical chords and chords a fifth or two fifths apart) and red/orange (high distance) towards to the top-right and bottom-left corners, where we find pairs such as E-Db. The color should then become blue/green (low distance) again near the top-right and bottom-left corner because of the circular arrangement of keys in circle-of-fifths (e.g., F$^{\#}$ is the fifth of B). It can be observed that the color patterns confirm the expectations from music theory, both in (a) major and (b) minor keys, in Figure 6.



(a) major keys

(b) minor keys

Fig. 6: Similarity matrix for average cosine distance between pairs of each of the 24 preludes in Bach's Well-Tempered Clavier and their 11 transposed versions in word2vec embedding space.

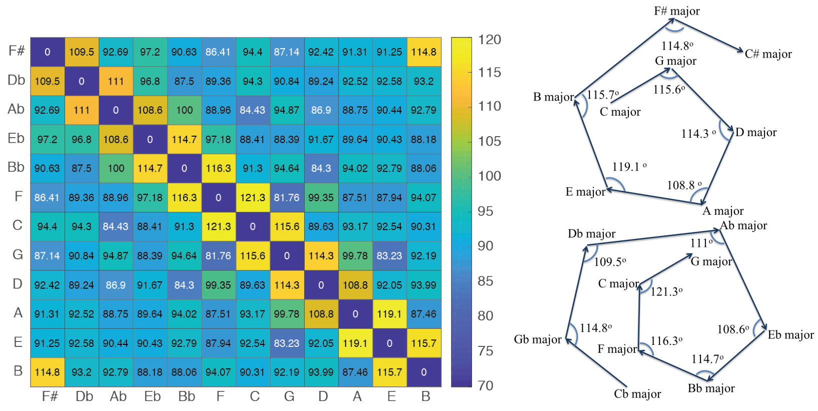
4.4 'Analogy' in music

One of the most striking results from word2vec in natural language processing is the emergence of analogies between words. Mikolov et al [45] studied linguistic regularity for examples of syntactic analogy such as 'man is to woman as king is to queen'. They proposed the analogy question to the learned word2vec space as $xa : xb = xc : xd$, where xa, xb, xc are given words. Their goal was to search for xd in the word2vec space where the vector xc-to-xd is parallel to the vector xa-to-xb. The authors found xd by searching for the word for which xc-to-xd has the highest cosine similarity to the vector xa-to-xb.

Although music lacks the explicit semantic content of language, researchers have argued that the construct of analogy exists in music as, for example, the "functional repetition" of musical content that is otherwise dissimilar [29]. Therefore, inspired by computational linguistics [45], we explore the meaning of analogy in the music word2vec space. Perhaps the most fundamental definition of analogy in tonal music would be the functional role of chords in a given key. For example, the relationship between the tonic chord and the dominant (I-V) should remain the same in all keys, i.e., C major triad is to G major triad in the key of C major as G major triad is to D major triad in the key of G major. Based on this definition, we investigate the analogy or relationship between a vector representing the transition from one chord to another (called a 'chord-pair vector'), and its transposed equivalent in every key.

For the present experiment, we focused on three chord-pair vectors: C major-to-G major (I-V chord-pair vector in major keys), A minor-to-E minor (i-v in minor keys), and C major-to-A minor (I-vi, the tonic and its relative minor triad, in major keys). In order to accommodate working with music instead of language, we had to slightly adapt the question of how analogy is represented geometrically: instead of searching for *xd* given *xa*, *xb*, and *xc*, we calculated the angle between the two chord-pair vectors *xa*-to-*xb* and *xc*-to-*xd* given all four variables (chords). This change was made because a vector in music word2vec space may not necessary be a chord, e.g., it could contain only one or two pitch classes. If the retrieved *xd* is not a chord that is commonly recognized by music theory, we cannot verify whether it is a meaningful analogy.

Figure 7a lists the angle in degrees between the I-V chord vectors in pairs of keys (where the x-and y-axis indicate every key).

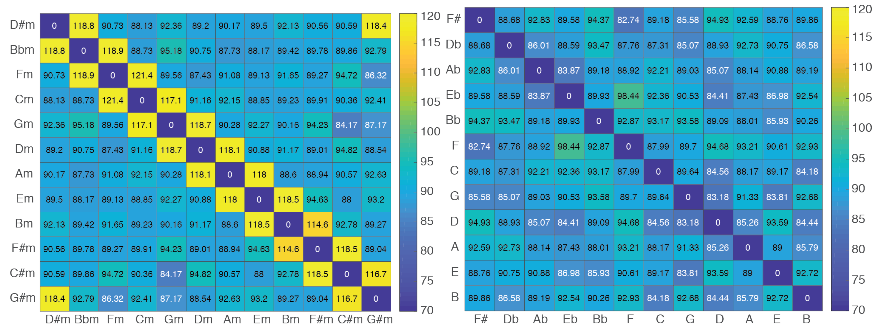


	F#	Db	Ab	Eb	Bb	F	C	G	D	A	E	B
F#	0	109.5	92.69	97.2	90.63	86.41	94.4	87.14	92.42	91.31	91.25	114.8
Db	109.5	0	111	96.8	87.5	89.36	94.3	90.84	89.24	92.52	92.58	93.2
Ab	92.69	111	0	108.6	100	88.96	84.43	94.87	86.9	88.75	90.44	92.79
Eb	97.2	96.8	108.6	0	114.7	97.18	88.41	88.39	91.67	89.64	90.43	88.18
Bb	90.63	87.5	100	114.7	0	116.3	91.3	94.64	84.3	94.02	92.79	88.06
F	86.41	89.36	88.96	97.18	116.3	0	121.3	81.76	99.35	87.51	87.94	94.07
C	94.4	94.3	84.43	88.41	91.3	121.3	0	115.6	89.63	93.17	92.54	90.31
G	87.14	90.84	94.87	88.39	94.64	81.76	115.6	0	114.3	99.78	83.23	92.19
D	92.42	89.24	86.9	91.67	84.3	99.35	89.63	114.3	0	108.8	92.05	93.99
A	91.31	92.52	88.75	89.64	94.02	87.51	93.17	99.78	108.8	0	119.1	87.46
E	91.25	92.58	90.44	90.43	92.79	87.94	92.54	83.23	92.05	119.1	0	115.7
B	114.8	93.2	92.79	88.18	88.06	94.07	90.31	92.19	93.99	87.46	115.7	0

(a) (b)

Fig. 7: (a) Similarity matrix for the angle (in degrees) between I-V vectors in pairs of keys (keys are indicated by the x-and y-axes) and (b) an illustration of the chord-pair vectors and angles between them. Note that the angles between chord-pair vectors are computed between adjacent chords; that is, the reader should not draw conclusions about the relationship between non-adjacent chords from this diagram. In addition, it should be noted that the lengths of the arrows were chosen to maximize the clarity of the figure, and as such do not convey semantic meaning in of themselves.

For example, the value 121.3 in row C and column F is the degree between the chord-pair vector C major-to-G major (I-V in C major) and the chord-pair vector F major-to-C major (I-V in F major). It is apparent that the numbers next to the diagonal line, which represent keys a fifth apart, are significantly different from other numbers in the matrix. To help the reader interpret the meaning of these highlighted numbers, we illustrate the I-V chord-pair vectors with their angle between different keys in Figure 7b. Note that Figure 7b almost mimics the circle-of-fifths, although it does not perfectly fit all 12 keys in the projected 2D circle due to the multidimensional nature of the underlying vectors. This again confirms that the relationships between keys in the circle-of-fifths is reflected in the learned vector space, as the angles between I-V chord-pair vectors all fall within a very specific range (108.8-121.3 degrees).



	D#m	Bbm	Fm	Cm	Gm	Dm	Am	Em	Bm	F#m	C#m	G#m
D#m	0	118.8	90.73	88.13	92.36	89.2	90.17	89.5	92.13	90.56	90.59	118.4
Bbm	118.8	0	118.9	88.73	95.18	90.75	87.73	88.17	89.42	89.78	89.86	92.79
Fm	90.73	118.9	0	121.4	89.56	87.43	91.08	89.13	91.65	89.27	94.72	86.32
Cm	88.13	88.73	121.4	0	117.1	91.16	92.15	88.85	89.23	89.91	90.36	92.41
Gm	92.36	95.18	89.56	117.1	0	118.7	90.28	92.27	90.16	94.23	84.17	87.17
Dm	89.2	90.75	87.43	91.16	118.7	0	118.1	90.88	91.17	89.01	94.82	88.54
Am	90.17	87.73	91.08	92.15	90.28	118.1	0	118	88.6	88.94	90.57	92.63
Em	89.5	88.17	89.13	88.85	92.27	90.88	118	0	118.5	94.63	88	93.2
Bm	92.13	89.42	91.65	89.23	90.16	91.17	88.6	118.5	0	114.6	92.78	89.27
F#m	90.56	89.78	89.27	89.91	94.23	89.01	88.94	94.63	114.6	0	118.5	89.04
C#m	90.59	89.86	94.72	90.36	84.17	94.82	90.57	88	92.78	118.5	0	116.7
G#m	118.4	92.79	86.32	92.41	87.17	88.54	92.63	93.2	89.27	89.04	116.7	0

(a) i-v

	F#	Db	Ab	Eb	Bb	F	C	G	D	A	E	B
F#	0	88.68	92.83	89.58	94.37	82.74	89.18	85.58	94.93	92.59	88.76	89.86
Db	88.68	0	86.01	88.59	93.47	87.76	87.31	85.07	88.93	92.73	90.75	86.58
Ab	92.83	86.01	0	83.87	89.18	88.92	92.21	89.03	85.07	88.14	90.88	89.19
Eb	89.58	88.59	83.87	0	89.93	98.44	92.36	90.53	84.41	87.43	86.98	92.54
Bb	94.37	93.47	89.18	89.93	0	92.87	93.17	93.58	89.09	88.01	85.93	90.26
F	82.74	87.76	88.92	98.44	92.87	0	87.99	89.7	94.68	93.21	90.61	92.93
C	89.18	87.31	92.21	92.36	93.17	87.99	0	89.64	84.56	88.17	89.17	84.18
G	85.58	85.07	89.03	90.53	93.58	89.7	89.64	0	83.18	91.33	83.81	92.68
D	94.93	88.93	85.07	84.41	89.09	94.68	84.56	83.18	0	85.26	93.59	84.44
A	92.59	92.73	88.14	87.43	88.01	93.21	88.17	91.33	85.26	0	89	85.79
E	88.76	90.75	90.88	86.98	85.93	90.61	89.17	83.81	93.59	89	0	92.72
B	89.86	86.58	89.19	92.54	90.26	92.93	84.18	92.68	84.44	85.79	92.72	0

(b) I-vi

Fig. 8: Similarity matrix for the angle (in degrees) between chord-pair vectors, (a) i-v and (b) I-vi, respectively, for pairs of minor keys (a) and for pairs of major keys (b). Keys are listed along the x-and y-axes.

Figure 8 shows the similarity matrices for the angle between chord-pair vectors i-v in pairs of minor keys and I-vi in pairs of major keys. One may observe that Figure 8a also shows a different pattern along the diagonal line, implying that the circle-of-fifths relationship is also learned in the music word2vec space for minor keys. In contrast, the relationship between the tonic and its relative minor (I-vi) is not maintained, as no significant patterns are observed in Figure 8b. It might be that the I-vi chord-pair vector does not occur as frequently as the I-V vector or that the context in which the I-vi chord-pair vector appears tends to be more diverse.

4.5 Music generation with word2vec slices

Because we have shown that word2vec is capable of recognizing different relationships between chords in Section 4.2, we give an example of a potential application of word2vec for music: music generation by means of substitutions. Systems that generate music by recommending alternative chords or pitches can be useful for composers and amateurs to experiment with new ideas or that are not in their repertoire. Music generation systems are also becoming popular to create music for commer-

Algorithm 1: Finding the substitute slice for music generation

```
Input     : $s$: input slice, $W2V_s$: slices in the word2vec vocabulary with their embeddings
Output    : $\bar{s}$: the substitute slice for $s$
Parameter: $n$: top $n$ candidate slices
if $s$ is not in $W2V_s$ then
    $\bar{s} \leftarrow s$;
end
else
    // initializing arrays for the slice, cosine distance, pitch class score, and pitch class count of
      the top n candidates (with 1 representing the same number of pitch classes compared to
      original piece and 0 representing a different amount) ;
    for $i$ = 1...n do
        $slice_n[i] \leftarrow$Inf;
        $distance_n[i] \leftarrow$Inf;
        $score_n[i] \leftarrow$0;
        $count_n[i] \leftarrow$0;
    end
    // finding the top/closest n slices to s;
    foreach slice $t$ in $W2V_s$ do
        if cosdis(s,t)<max($distance_n$) then
            $i \leftarrow argmax(distance_n)$;
            $slice_n[i] \leftarrow t$;
            $distance_n[i] \leftarrow$ cosdis($s$, $t$);
        end
    end
    // calculating weights for the 12 pitch classes in array pitchclasses;
    foreach pc in pitchclasses do
        $pc \leftarrow$0;
    end
    foreach slice $t$ in $slice_n$ do
        foreach note in $t$ do
            $pitchclasses[note] \leftarrow pitchclasses[note]$+1
        end
    end
    foreach pc in pitchclasses do
        $pc \leftarrow pc$/sum($pc$)
    end
    // selecting $\bar{s}$;
    for $i$=1...n do
        foreach note in $slice_n[i]$ do
            $score_n[i]$ = $score_n[i]$+$pitchclasses[note]$
        end
        $score_n[i]$ = $score_n[i]$/(total number of pitch classes in $slice_n[i]$;
        if number of pitch classes in $slice_n[i]$ == number of pitch classes in $s$ then
            $count_n[i]$ = 1;
        end
    end
    // when no slices with same number of pitch classes are in top-n list;
    if sum($count_n$)==0 then
        $i \leftarrow$ argmax($score_n$);
        $\bar{s} \leftarrow slice_n[i]$;
    end
    // when there are slices with same number of pitch classes in the top-n list;
    else
        $m \leftarrow$0;
        $j \leftarrow$0;
        for $i$=1...n do
            if $count_n[i]$==1 and $score_n[i] > m$ then
                $m \leftarrow score_n[i]$;
                $j \leftarrow i$;
            end
        end
        $\bar{s} \leftarrow slice_n[j]$;
    end
end
```

cial applications (for example, see Jukedeck[6]). For a complete overview of current state-of-the-art of music generation systems, the reader is referred to Herremans et al [26].

Algorithm 1 describes the strategy that we propose for finding a substitute slice for a given input slice using the trained word2vec model. Note that the pitches in a slice may or may not represent a chord. If the input slice is not in the word2vec vocabulary, the algorithm returns the original input slice as its own substitute (lines 1-3). Otherwise, it calculates the cosine distance between all slices in the vocabulary and the input slice (lines 6-11). It then collects a list of n slices that are the closest to the input, where n is an input parameter of the algorithm (lines 13-19). Because the music considered in this study is tonal, the algorithm uses the top-n slices to calculate a score for the 12 pitch classes, which can be used to infer the key. The score of a particular pitch class is its normalized frequency of occurrence in the top-n slices (lines 21-41).

As a design rule, we chose to preferentially substitute a slice with one containing the same number of pitch classes. Therefore, we added a preference rule in the algorithm: From the top-n slices, the algorithm selects the slice with the highest pitch class scores among those with the same number of pitch classes. If none of the slices in the top-n list have the same number of pitch classes as in the input slice, the algorithm returns the slice with the highest pitch class score regardless of the number of pitch classes (lines 38-40 and 42-57).

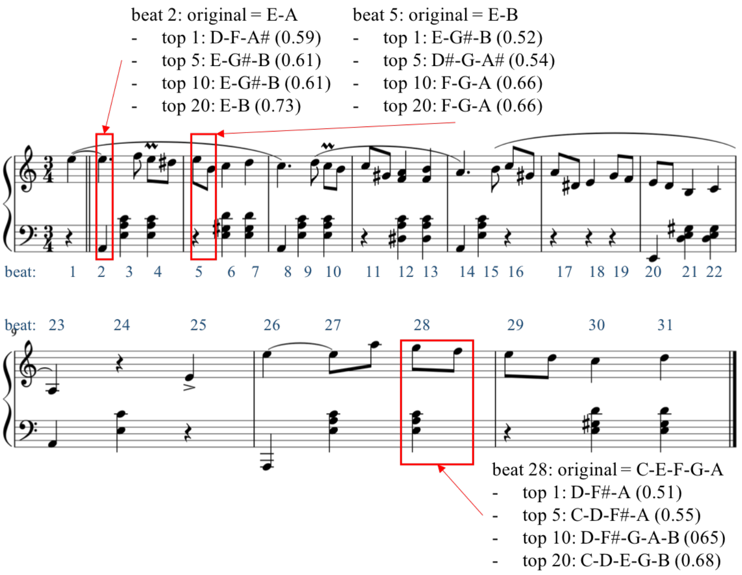
To show the effectiveness of the algorithm, we applied the algorithm to Chopin's Mazurka Op. 67 No. 4 and generated substitute slices for the first 30 beats. For the input parameter n, we experimented with four settings for the top-n list: 1, 5, 10 and 20 slices. The details of the generated slices and their cosine distance to their respective input slice can be found in Figure 10 in Appendix.

To illustrate the effect of using different values for n, we selected the beats for which the substitute slice has a relatively short distance to the original, and then depict them on the score, as shown in Figure 9. Three beats were selected: beats 2, 5, and 28. As shown in Figure 10, the substitute slice for these three beats has a smaller distance to the original compared to the other beats. In addition, the slice that the model selects for substitution in the selected beats changes as n increases. For each of the three highlighted beats in 9, we annotated the pitch classes in the original slice (as shown in the score) and of the substitute slice, and calculated the cosine distance between them (in parentheses). The most notable trend in Figure 9 is that the number of out-of-key pitch classes decreases when the value of n increases. Because this piece is in the key of A minor, pitch classes with sharps (#) or flats (b) are considered to be outside of the key. When $n = 20$, none of the selected slices contain any sharps or flats, which is surprising, as the cosine distance also increases when the value of n increases.

Readers may listen to examples of the generated pieces online[7]. It should be noted that our aim was not to create a full-fledged music generation system, but rather, to illustrate how word2vec might be useful in a music generation context. Although

[6] www.jukedeck.com/

[7] http://sites.google.com/view/chinghuachuan/



beat 2: original = E-A
- top 1: D-F-A# (0.59)
- top 5: E-G#-B (0.61)
- top 10: E-G#-B (0.61)
- top 20: E-B (0.73)

beat 5: original = E-B
- top 1: E-G#-B (0.52)
- top 5: D#-G-A# (0.54)
- top 10: F-G-A (0.66)
- top 20: F-G-A (0.66)

beat 28: original = C-E-F-G-A
- top 1: D-F#-A (0.51)
- top 5: C-D-F#-A (0.55)
- top 10: D-F#-G-A-B (065)
- top 20: C-D-E-G-B (0.68)

beat: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
beat: 23 24 25 26 27 28 29 30 31
9

Fig. 9: Chopin Mazurka Op. 67 No. 4, with newly generated slices for three selected beats. Each of these slices is annotated with the cosine distance of the best slice (according to Algorithm 1) in a set of top-n lists.

we have only described one method for slice replacement above, there are numerous ways in which one might appropriate word2vec for slice/chord replacement in music. In future research, it would be interesting to explore how word embeddings, which, as we have demonstrated, capture meaningful musical features, may be used as input for other music generation models.

5 Conclusion

In this paper, we explore whether a popular technique from computational linguistics, word2vec, is useful for modeling music. More specifically, we implement a skip-gram model with negative sampling to construct a semantic vector space for complex polyphonic musical slices. We expand upon preliminary research from the authors [24], and provide the first thorough examination of the kinds of semantic information that word2vec is capable of capturing in music.

In our experiment, we trained a model on a large MIDI dataset consisting of multiple genres of polyphonic music. This allowed our dataset to be much larger and more ecologically valid than what is traditionally used for music-based models. Although previous preliminary research [27] explored the use of word2vec by modeling 92 labeled chords, the present work builds upon initial work by the authors [24] by using a vocabulary of 4,076 complex polyphonic slices with no labels. In contrast to many traditional music modeling approaches, such as convolutional neural networks using a ‘piano roll’ representation, our representation does not encode any musical information (e.g., intervals) within the slices. We sought to explore whether word2vec is powerful enough to derive tonal and harmonic properties based solely on the co-occurrence statistics of musical slices.

First, we found that tonal distance between chords is indeed reflected in the learned vector space. For instance, as would be expected from music theory, the tonic and dominant chords of a key (I and V) have smaller cosine distance than the tonic and the second (I and II) chords. This shows that elements of tonal proximity are captured by the model, which is striking, because the model received no explicit musical information about the *semantic content* of chords. Second, we observed that the relationships between keys are reflected in our geometrical model: by transforming Bach’s Well Tempered Clavier to all keys, we were able to verify that the cosine distance between the centroids of tonally-similar keys is much smaller than for tonally-distant keys. Third, we observed that the angle between I-V chord vectors between pairs of keys (e.g., G major to D major, Eb major to Bb major, etc) in vector space is generally consistent with the circle-of-fifths. This suggests that the model was able to extract fundamental harmonic relationships from the slices of music. In sum, not only are tonal and harmonic relationships learned by our model over the course of training, but the associations between keys are learned as well.

By extracting information about chords and chord relationships, the model’s learned representations highlight the building blocks of musical *analogy*. In the present approach, musical *analogy* is defined as the semantic invariance of chord transitions, such as I-V, across different keys. That is, similar translations (e.g., the angle between chord-pair vectors) can be used to move between functional chords in different keys because the angle between chord pairs (such as I-V) is preserved regardless of the key. As we can see from the angle between chord-pair vectors across keys, the model successfully learns the relationship between functional chords in different keys: the angle between I-V and i-v chord-pair vectors is more consistent than for I-vi chord-pair vectors across keys (see Figures 7 and 8).

Given the potential of word2vec to capture musical characteristics, we performed an initial exploration of how a semantic vector space model may be used for music generation. The approach we tested involved a new strategy for replacing musical slices with tonally-similar slices based on word2vec cosine similarity. While it is not within the scope of the current paper to explore the quality of the generated music, our approach describes a new way in which word2vec could be a useful tool for future automatic composition tools.

This paper demonstrates the promising capabilities of word2vec for modeling basic musical concepts. In the future, we plan to further explore word2vec’s ability to model complex relationships by comparing it with mathematical models of tonality, such as the one described in [11]. We also plan to combine word2vec with sequential modeling techniques (e.g., recurrent neural networks) to model structural concepts like musical tension [25]. The results of such studies could provide a new way of capturing *semantic similarity* and *compositional style* in music. We will also continue to explore the use of word2vec in music generation by further investigating aspects of music similarity and style. In future work, it would also be interesting to investigate how word2vec may be integrated as an automatic feature extraction model into existing models, such as RNNs and LSTMs. This could be done, for instance, by using the learned word2vec word embeddings as input for these models. The reader is invited to further build upon the model of this project, which is available online.[8]

Acknowledgements This research was partly supported through SUTD Grant no. SRG ISTD 2017 129.

Conflict of interest The authors declare that they have no conflict of interest.

References

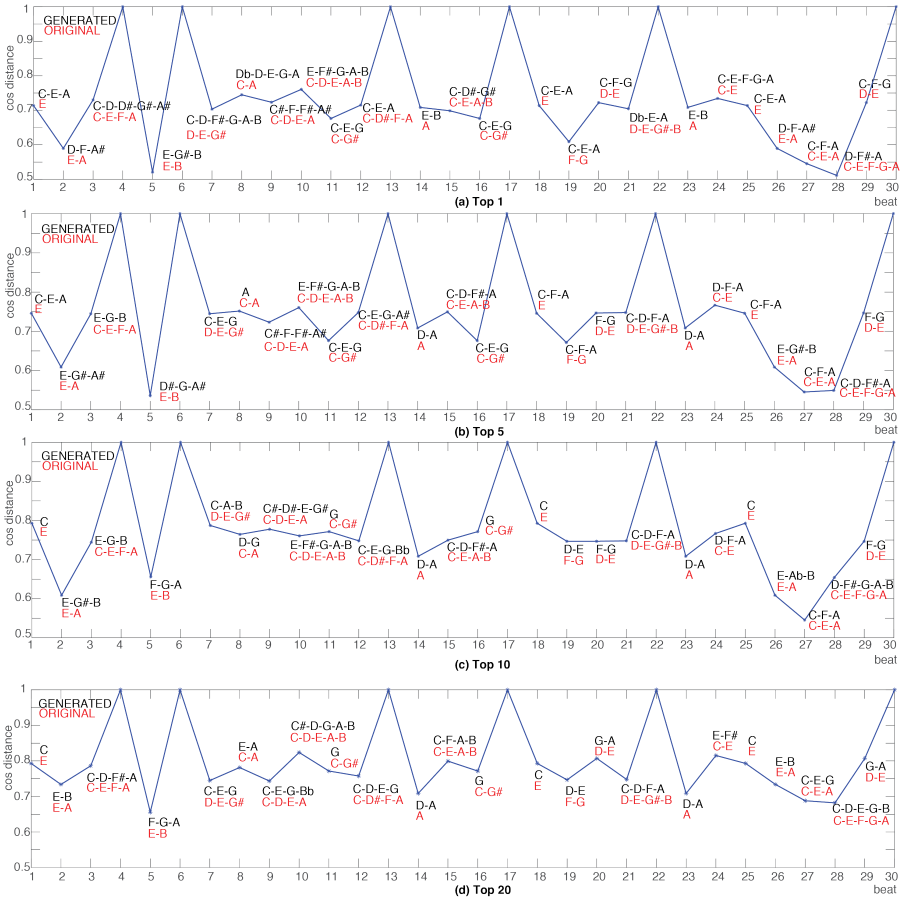
1. Agres K, Cancino C, Grachten M, Lattner S (2015) Harmonics co-occurrences bootstrap pitch and tonality perception in music: Evidence from a statistical unsupervised learning model. Proceedings of the Cognitive Science Society
2. Agres K, Abdallah S, Pearce M (2018) Information-theoretic properties of auditory sequences dynamically influence expectation and memory. Cognitive science 42(1):43–76
3. Agres KR, McGregor S, Rataj K, Purver M, Wiggins GA (2016) Modeling metaphor perception with distributional semantics vector space models. In: Workshop on Computational Creativity, Concept Invention, and General Intelligence. Proceedings of 5 th International Workshop, C3GI at ESSLI, pp 1–14
4. Allan M, Williams C (2005) Harmonising chorales by probabilistic inference. In: Advances in neural information processing systems, pp 25–32
5. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. Journal of machine learning research 3(Feb):1137–1155
6. Besson M, Schön D (2001) Comparison between language and music. Annals of the New York Academy of Sciences 930(1):232–258
7. Boulanger-Lewandowski N, Bengio Y, Vincent P (2012) Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. arXiv preprint arXiv:12066392
8. Cancino-Chacón C, Grachten M, Agres K (2017) From bach to the beatles: The simulation of human tonal expectation using ecologically-trained predictive models. In: ISMIR, Suzhou, China
9. Chacón CEC, Lattner S, Grachten M (2014) Developing tonal perception through unsupervised learning. In: ISMIR, pp 195–200
10. Chew E (2000) Towards a mathematical model of tonality. PhD thesis, Massachusetts Institute of Technology
11. Chew E, et al (2014) Mathematical and computational modeling of tonality. AMC 10:12
12. Choi K, Fazekas G, Sandler M (2016) Text-based lstm networks for automatic music composition. arXiv preprint arXiv:160405358
13. Chuan CH, Herremans D (2018) Modeling temporal tonal relations in polyphonic music through deep networks with a novel image-based representation. In: The Thirty-Second AAAI Conference on Artificial Intelligence, AAAI, AAAI, New Orleans, US
14. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, ACM, pp 160–167
15. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. Journal of Machine Learning Research 12(Aug):2493–2537
16. Conklin D, Witten IH (1995) Multiple viewpoint systems for music prediction. Journal of New Music Research 24(1):51–73
17. Dhillon P, Foster DP, Ungar LH (2011) Multi-view learning of word embeddings via CCA. In: Advances in neural information processing systems, pp 199–207
18. Eck D, Schmidhuber J (2002) Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In: Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on, IEEE, pp 747–756

[8] `https://sites.google.com/view/chinghuachuan/`

19. Erk K (2012) Vector space models of word meaning and phrase meaning: A survey. Language and Linguistics Compass 6(10):635–653
20. Firth JR (1957) A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis
21. Goldberg Y, Levy O (2014) word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:14023722
22. Gutmann MU, Hyvärinen A (2012) Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. Journal of Machine Learning Research 13(Feb):307–361
23. Harris ZS (1954) Distributional structure. Word 10(2-3):146–162
24. Herremans D, Chuan CH (2017) Modeling musical context with word2vec. In: First International Workshop On Deep Learning and Music joint with IJCNN, Anchorage, US, vol 1, pp 11–18
25. Herremans D, Weisser S, Sörensen K, Conklin D (2015) Generating structured music for bagana using quality metrics based on markov models. Expert Systems with Applications 42(21):7424–7435
26. Herremans D, Chuan CH, Chew E (2017) A functional taxonomy of music generation systems. ACM Computing Surveys (CSUR) 50(5):69
27. Huang CZA, Duvenaud D, Gajos KZ (2016) Chordripple: Recommending chords to help novice composers go beyond the ordinary. In: Proceedings of the 21st International Conference on Intelligent User Interfaces, ACM, pp 241–250
28. Huron DB (2006) Sweet anticipation: Music and the psychology of expectation. MIT press
29. Kielian-Gilbert M (1990) Interpreting musical analogy: From rhetorical device to perceptual process. Music Perception: An Interdisciplinary Journal 8(1):63–94
30. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:14085882
31. Koelsch S, Schmidt Bh, Kansok J (2002) Effects of musical expertise on the early right anterior negativity: an event-related brain potential study. Psychophysiology 39(5):657–663
32. Korzeniowski F, Widmer G (2016) A fully convolutional deep auditory model for musical chord recognition. In: Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on, IEEE, pp 1–6
33. Krumhansl CL (1990) Cognitive foundations of musical pitch. Oxford University Press
34. Krumhansl CL, Schmuckler M (1990) A key-finding algorithm based on tonal hierarchies. Cognitive Foundations of Musical Pitch pp 77–110
35. Lebret R, Collobert R (2013) Word emdeddings through hellinger PCA. arXiv preprint arXiv:13125542
36. Lerdahl F, Jackendoff R (1977) Toward a formal theory of tonal music. Journal of music theory 21(1):111–171
37. Lewin D (1982) A formal theory of generalized tonal functions. Journal of Music Theory 26(1):23–60
38. Liddy ED, Paik W, Edmund SY, Li M (1999) Multilingual document retrieval system and method using semantic vector matching. US Patent 6,006,221
39. Madjiheurem S, Qu L, Walder C (2016) Chord2vec: Learning musical chord embeddings. In: Proceedings of the Constructive Machine Learning Workshop at 30th Conference on Neural Information Processing Systems (NIPS2016), Barcelona, Spain
40. McGregor S, Agres K, Purver M, Wiggins GA (2015) From distributional semantics to conceptual spaces: A novel computational method for concept creation. Journal of Artificial General Intelligence 6(1):55–86
41. Meyer LB (1956) Emotion and meaning in music. University of chicago Press
42. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781
43. Mikolov T, Le QV, Sutskever I (2013) Exploiting similarities among languages for machine translation. arXiv preprint arXiv:13094168
44. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
45. Mikolov T, Yih Wt, Zweig G (2013) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 746–751
46. Mnih A, Hinton GE (2009) A scalable hierarchical distributed language model. In: Advances in neural information processing systems, pp 1081–1088
47. Mnih A, Kavukcuoglu K (2013) Learning word embeddings efficiently with noise-contrastive estimation. In: Advances in neural information processing systems, pp 2265–2273

48. Noland K, Sandler M (2009) Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio. Computer Music Journal 33(1):42–56
49. Pearce MT, Wiggins GA (2012) Auditory expectation: the information dynamics of music perception and cognition. Topics in cognitive science 4(4):625–652
50. Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
51. Poliner GE, Ellis DP (2006) A discriminative model for polyphonic piano transcription. EURASIP Journal on Advances in Signal Processing 2007(1):048,317
52. Poria S, Cambria E, Gelbukh A (2015) Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 2539–2544
53. Poria S, Cambria E, Hazarika D, Vij P (2016) A deeper look into sarcastic tweets using deep convolutional neural networks. arXiv preprint arXiv:161008815
54. Saffran JR, Johnson EK, Aslin RN, Newport EL (1999) Statistical learning of tone sequences by human infants and adults. Cognition 70(1):27–52
55. Sak H, Senior AW, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Interspeech, pp 338–342
56. Salton G (1971) The SMART retrieval systemexperiments in automatic document processing. Prentice-Hall, Inc.
57. Salton G, Wong A, Yang C (1975) A vector space model for automatic indexing. Communications of the ACM 18
58. Schwartz R, Reichart R, Rappoport A (2015) Symmetric pattern based word embeddings for improved word similarity prediction. In: Proceedings of the Nineteenth Conference on Computational Natural Language Learning, pp 258–267
59. Tan PN, Steinbach M, Kumar V (2005) Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
60. Toiviainen P, Eerola T (2016) MIDI toolbox 1.1. https://github.com/miditoolbox/
61. Turney PD, Pantel P (2010) From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research 37:141–188

A Appendix



(a) Top 1

(b) Top 5

(c) Top 10

(d) Top 20

Fig. 10: Generated slices and their cosine distance to the original slices from Chopin's Mazurka Op. 67 No. 4, using (a) top 1, (b) top 5, (c) top 10, and (d) top 20 slices for the search in music word2vec space. Note that as the value of *n* increases (e.g., moving from figure (a) down to (d)), the number of pitches outside of the key (see generated pitches in black) decreases.