

Chord Embeddings: Analyzing What They Capture and Their Role for Next Chord Prediction and Artist Attribute Prediction

Allison Lahnala¹, Gauri Kambhatla¹, Jiajun Peng¹, Matthew Whitehead², Gillian Minnehan¹, Eric Guldán¹, Jonathan K. Kummerfeld¹, Anıl Çamcı³, and Rada Mihalcea¹

¹ Department of Computer Science & Engineering

² School of Information

³ Department of Performing Arts Technology

University of Michigan, Ann Arbor MI 48109, USA

{alclahn,gkambhat,pjiajun,mwwhite,gminn,
eguldán,jkummerf,acamci,mihalcea}@umich.edu

Abstract. Natural language processing methods have been applied in a variety of music studies, drawing the connection between music and language. In this paper, we expand those approaches by investigating *chord embeddings*, which we apply in two case studies to address two key questions: (1) what musical information do chord embeddings capture?; and (2) how might musical applications benefit from them? In our analysis, we show that they capture similarities between chords that adhere to important relationships described in music theory. In the first case study, we demonstrate that using chord embeddings in a next chord prediction task yields predictions that more closely match those by experienced musicians. In the second case study, we show the potential benefits of using the representations in tasks related to musical stylometrics.

Keywords: Chord Embeddings · Representation Learning · Musical Artificial Intelligence.

1 Introduction

Natural language processing (NLP) methods such as classification, parsing, or generation models have been used in many studies on music, drawing the connection that music is often argued to be a form of language. However, while word embeddings are an important piece of almost all modern NLP applications, embeddings over musical notations have not been extensively explored. In this paper, we explore the use of *chord embeddings* and argue that it is yet another NLP methodology that can benefit the analysis of music as a form of language. Our objectives are (1) to probe embeddings to understand what musical information they capture, and (2) to demonstrate the value of embeddings in two example applications.

Using `word2vec` [18] to create embeddings over chord progressions, we first perform qualitative analyses of chord similarities captured by the embeddings using Principal Component Analysis (PCA). We then present two case studies on chord embedding applications, first in next-chord prediction, and second in artist attribute prediction. To show their value, we compare models that use chord embeddings with ones using other forms of chord representations.

In the next chord prediction study, we provide a short chord sequence and ask what the next chord should be. We collected human annotations for the task as a point of reference. By comparing model predictions with the human annotations, we observe that models using chord embeddings yield chords that are more similar to the predictions of more experienced musicians. We also measure the system’s performance on a larger set drawn from real songs. This task demonstrates a use case for chord embeddings that involves human perception, interaction, and composition. For the artist attribute prediction study, we perform binary classification tasks on artist type (solo performer or group), artist gender (when applicable), and primary country of the artist. Results on these tasks demonstrate that chord embeddings could be used in studies of musical style variations, including numerous studies in musicology.

This paper contributes analyses of the musical semantics captured in *chord2vec* embeddings and of their benefits to two different computational music applications. We find that the embeddings encode musical relationships that are important in music theory such as the *circle-of-fifths* and relative major and minor chords. The case studies provide insight into how musical applications may benefit from using chord embeddings in addition to NLP methods that have previously been employed.

2 Related Work

Methods for learning word embeddings [18, 22, 23, 4] have been useful for domains outside of language (e.g., network analysis [7]). Recent work has explored embeddings for chords, including an adaptation of *word2vec* [15], their use in a chord progression prediction module of a music generation application [3], and for aiding analysis and visualization of musical concepts in Bach chorales [24]. However, understanding the musical information captured as latent features in the embeddings has been limited by the decision to ground evaluation in language modeling metrics (e.g., perplexity) rather than analyses of their behavior in downstream tasks. In this work, our first case study shows that language models with no remarkable differences in performance by perplexity exhibit remarkable relationships in their predictions to the experience of musicians, and furthermore we provide insights into what is captured by the embeddings.

NLP methods have benefited computational musicology topics such as authorship attribution [30], lyric analysis [6], and music classification tasks using audio and lyrics [16]. In the task of composer identification, many approaches draw inspiration from NLP, applying musical stylometry features and melodic n-grams [2, 31, 8]. One study used language modeling methods on musical n-

grams to perform composer recognition [10] and another used a neural network over encodings of the pitches of musical pieces [12]. Similar methods have been used to study stylistic characteristics of eight jazz composers [19, 1] over chord sequences similar to ours. While these studies operated on small datasets (on the order of hundreds of samples) to identify and analyze music of a small set of musicians, we use a large dataset (on the order of tens of thousands) and predict attributes of artists based on the music.

Our attribute prediction tasks are related to NLP work in authorship attribution and are motivated by studies on the connection between language and music in psychology [21, 11], and the intersection of society, music, and language, or sociomusicology [5, 28]. For instance, Sergeant & Himonides studied the perception of gender in musical composition, and found no significant match between the listener’s guess of a composer’s gender and their actual gender [27].

3 Dataset

We compile a dataset of 92,000 crowdsourced chord charts with lyrics from the Ultimate Guitar website.⁴ We identify and remove duplicate songs⁵ in our data using Jaccard similarity, then extract the chord progressions from our data representation to learn chord embeddings.

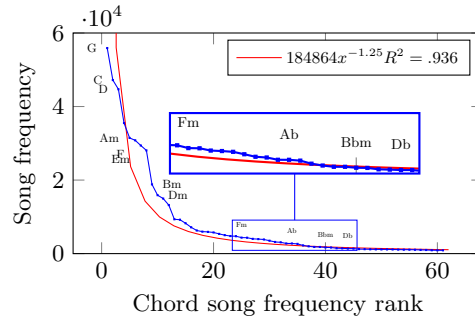


Fig. 1. Number of chords in the dataset and a fitted trendline with the parameters given in the figure for the 61 most common chords, showing a power law distribution. A sample of chords are labeled that also appear in the embedding visualization in Figure 4.

We remove songs with fewer than six chords, leaving us with a final set of 88,874 songs and 4,913 unique chords. The chords’ song frequencies are distributed according to a power law distribution, much like Zipf’s word frequency law exhibited in natural language. Figure 1 shows the song frequency and the power law trend line fitted to the top 61 chords for demonstration, though the

⁴ <https://www.ultimate-guitar.com/>

⁵ Sometimes multiple users submit chord charts for a song.

trend is even stronger over all chords. Beyond there existing many possible chord notations, we observe a relation between chord frequency and physical difficulty of playing the chord on the guitar or other instruments (e.g., G, C, and D), and variations in notation.

4 Chord Embeddings

Word embeddings have been used extensively to represent the meaning of words based on their use in a large collection of documents. We consider a similar process for chords, forming representations based on their use in a large collection of songs.

We create chord embeddings for chords that appear in at least 0.1% of our songs (237 chords) using the continuous bag of words model (\mathbf{CE}_{cbow}) and the *skip-gram* language model (\mathbf{CE}_{sglm}) from `word2vec` [18], a widely used method for creating word embeddings. For \mathbf{CE}_{cbow} , a target chord is predicted based on context chords, while \mathbf{CE}_{sglm} is the reverse: context chords are predicted given a target chord. In both cases, this has the effect of learning representations that are more similar for chords that appear in similar contexts. We tested context sizes of two, five, and seven, and varied the vector dimensions between 50, 100, 200, and 300, but observed only minor differences across different models, and chose to use a context window of five and a vector dimension of 200.

4.1 Qualitative Analyses

To better understand the information encoded in chord embeddings, we perform a qualitative analysis using PCA and present a 2D projection for the \mathbf{CE}_{sglm} model in Figure 4 (our main observations are consistent for the case of \mathbf{CE}_{cbow}).

We observe that chords that form a fifth interval are closer together, which suggests that the embeddings capture an important concept known as the *circle of fifths*. Fifth-interval relationships serve broad purposes in tonal music (music that adheres to a model of relationships to a central tone). Saker [26] encapsulates their structural importance by stating that “circle of fifths relationships dominate all structural levels of tonal compositions” and that “the strongest, most copious harmonic progressions to be found in tonal music are fifth related.” The *circle of fifths* relationship is observed in our chord embeddings over different chord *qualities*,⁶ specifically over major chords (highlighted in Figure 4b), minor chords (highlighted in Figure 4c), major-minor 7 chords, and minor 7 chords. For both chord qualities, the layout captured by the chord embeddings is similar to the ideal/theoretical circle of fifths, illustrated in Figure 4a. This pattern is particularly interesting as it does not follow the style of word analogy patterns

⁶ *Qualities* refers to sound properties that are consistent across chords with different roots but equidistant constituent pitches. The interaction of intervals between pitches determines the quality.

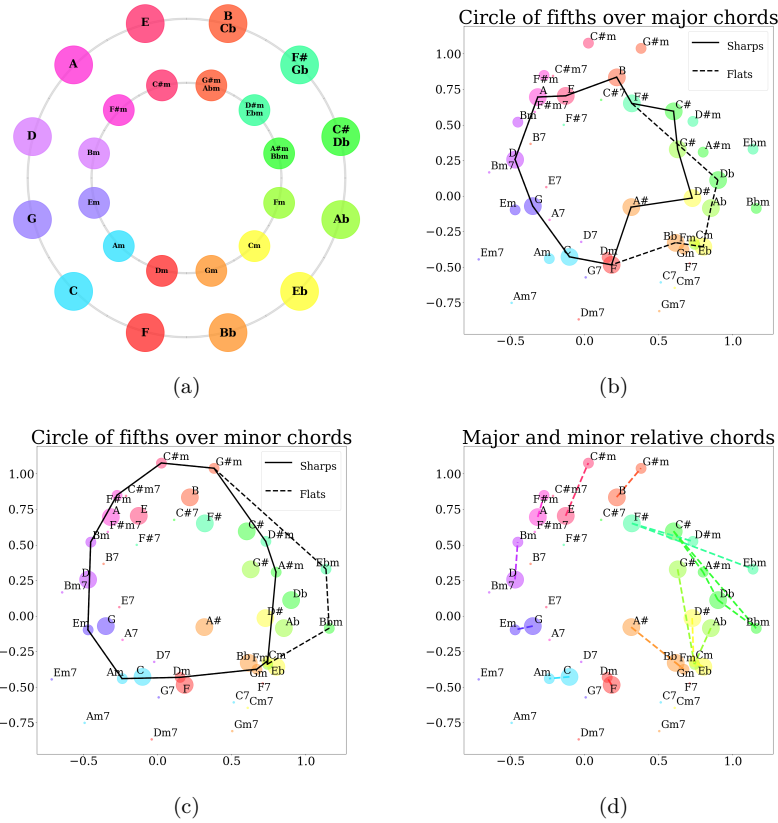


Fig. 2. In (a), we show the circle of fifths with the same colors as in (b), (c) and (d), which show the same 2-dimensional PCA projection of the chord embedding space with lines denoting the circle of fifths over major chords (b) and minor chords (c), and lines denoting major-minor relatives (d).

observed in language. This makes sense, as the “is-a-fifth” relation forms a circle in chords, whereas word analogies connect pairs of words without forming a circle.

Additionally, we observe that relative major and minor chords⁷ appear relatively close together in the embedding space, as shown by their proximity in the PCA plots (highlighted in Figure 4d). We also observe that enharmonics, notes with different names but the same pitch, are often close together. Not only that, but there is a consistent pattern in the positioning of enharmonics, with sharps to the left and flats to the right.

⁷ *Relative* refers to the relation between the chords’ roots, in which the scale beginning on the minor chord’s root shares the same notes as the scale beginning on the major chord’s root, but the ordering of the notes give different qualities to the scales.

These observations suggest that chord embeddings are capable of representing musical relationships that are important to music theory. Transitions between the tonic (I), dominant (V), and subdominant (IV) chords of a scale are prescriptive components in musical cadences [25]. Since these chords frequently appear in the same context, their embeddings are more similar. A common *deceptive cadence* is a transition between the fifth and sixth root chords of a scale [20]. An example progression with a deceptive cadence in C major is C-F-G-Am; these chords appear in a similar neighborhood in the PCA plots. Because these chords frequently co-occur in music, the embeddings capture a relationship between them.

We also note that relationships for chords that are used more frequently are more strongly represented. The major and minor relative pairs (G, Em), (C, Am), and (D, Bm), are among the top ten chords ranked by song frequency (Figure 1) and have clear relations in Figure 4. In contrast, the pairs (Ab, Fm) and (Db, Bbm) are ranked lower and their minor-major relative relationship appears weaker by their distance.

4.2 Alternative Representations

In addition to chord embeddings, we also explore two other chord representations: Pitch Representations (PR) and Bag-of-Chords (BOC). For a fair comparison, we use the same vocabulary of 237 chords for these representations.

Pitch Representations. A chord’s identity is determined by its pitches, so we test if the individual pitches provide a better representation of a chord as a whole than our chord embeddings. This method represents each chord by encoding each of its pitches by their index in the chromatic scale $\{C = 1, C\# = 2, \dots, B = 12\}$. The pitches are in order of the triad, followed by an additional pitch if marked, and by one extra dimension for special cases.⁸ Additional pitches are indicated by the interval relative to the root of the pitch that is being added. We also represent chord inversions, e.g., the chord G/B which is an inversion of G such that the bottom pitch is B.

Bag-of-Chords. In this representation, each chord is represented as a “one-hot” vector, where the vectors have length equal to the vocabulary size. We consider two ways of determining the value for a chord. For BOC_{count} , we use the frequency of a chord in a song, divided by the number of chords in the song. For BOC_{tfidf} , we use the TF-IDF of each chord (term-frequency, inverse document frequency).

5 Case Study One: Next Chord Prediction

In this section, we present our first case study, which investigates if there is a relationship between chord embedding representations and the ways humans

⁸ Special cases include: the “*” marking on a chord, which is a special marker specific to the ultimate-guitar.com site; “UNK” which we use to replace chords that do not meet the 0.1% document frequency threshold; and “H” and “Hm” which indicates “hammer-ons” in the notation on ultimate-guitar.com

perceive and interact with chords. We test the use of chord representations for predicting the most likely next chord following a given sequence of chords, and then compare these to human-annotated responses.

We train a next chord prediction model using a *long short-term memory* model [9] (LSTM).⁹ We follow standard practice and do not freeze the embeddings, meaning the chord representations undergo updates throughout training, adjusting to capture the musical features most important for the task. Our main model uses the pre-trained chord embeddings to initialize the chord prediction architecture. We test both CE_{cbow} and CE_{sglm} embeddings, and will refer to these models by these acronyms. We also define a baseline model, where the encoder is randomly initialized (denoted NI, for no initialization). Finally, we also evaluate a model where we initialize the encoder with the pitch representations introduced in Section 4.2 (denoted PR).

We divide our data into three sets, with 69,985 songs (80%) for training, 8,748 songs (10%) for validation, and 8,748 (10%) for testing. We train using a single GPU with parameters: epochs = 40, sequence length = 35, batch size = 20, dropout rate = 0.2, hidden layers = 2, learning rate = 20, and gradient clipping = 0.25.

5.1 Human Annotations

To evaluate the next chord prediction models, we collect data with a human annotation task in which annotators are asked to add a new chord at the end of a chord progression. For example, given the progression “A, D, E,” they must pick a chord that would immediately follow E. They are also asked to pick one or two alternatives to this selection. Continuing the example, if an annotator provides E7 and A, then the chord progressions they have specified are “A, D, E, E7,” and “A, D, E, A.” The annotators are given a tool to play 48 different chords (all major, minor, major-minor 7, and minor 7 chords) so they can hear how different options would sound as part of the progression.¹⁰ They were given a total of 39 samples shown in the same order to all annotators. The samples were chosen randomly from our entire dataset of songs, permitting only one sequence to come from a single song, and requiring they contain the same 48 chords. We presented sequences of length three and six as we expect that patterns in the given sequence affect the responses.

Participants. The annotators were first asked to estimate their expertise in music theory on a scale from 0 - 100, where 0 indicates no knowledge of music theory, 25 - 75 indicates some level of knowledge from pre-university training though self-teaching, private lessons/tutoring, or classroom settings, and 75 - 100 indicates substantial expertise gained by formal university studies, performing and/or composing experience. They were given the option to provide comments about how they estimated their expertise. We collected this information because

⁹ We use an open-source repository of neural language models https://github.com/pytorch/examples/blob/master/word_language_model/model.py

¹⁰ We did not limit our next chord prediction models to these 48 chords.

we expected that the annotations provided by a participant may vary depending on their background education in music theory. It also allows us to perform comparisons of our system with sub-groups defined by self-reported knowledge. Nine participants provided complete responses, with expertise ratings of 0, 0, 10, 10, 19, 25, 25, 50 and 73. For the following analyses, we define a *beginner* set containing annotators who provided 0 for their self-rating, an *intermediate* set containing annotators with ratings > 0 and < 50 , and an *expert* set containing annotators whose ratings are at least 50.

Inter-annotator Agreement. For pairwise agreement, we compute the proportion of chord progressions in which a pair of annotators provided the same chord, averaged over all pairs of annotators. The pairwise agreement across all annotators is 22.51, it is 23.08 for the beginner set, 25.38 for the intermediate set, and 17.95 for the expert set.

To account for responses of similar but not identical chords (discussed in Section 5.2), we measure pairwise agreement on response pitches. We compute the fraction of matching pitches for a pair of annotators’ responses for a given chord progression, averaged over all pairs of annotators. The pairwise pitch agreement score for all annotators is 38.00, it is 37.01 for the beginner set, 40.90 for the intermediate set, and 33.59 for the expert set. The average number of unique chords used by each annotator is 30.2, it is 35.5 for the beginner set, 27.4 for intermediate set, and 32.0 for the expert set.

5.2 Evaluation Metrics

The main objective of this case study is to investigate whether chord similarities captured by our embeddings reflect human-perceived similarities. We use the chord-prediction systems to perform the same task given to the annotators. Each model provides a probability distribution over the full set of chords, therefore we treat the chords with highest probabilities as each model’s selection.

We evaluate the predictions with the following metrics, which are inspired by the metrics employed by the Lexical Substitution task [17] but modified for our setup, which weight more frequent responses higher:

Match_{best}: For each example we calculate the fraction of people who included the model’s top prediction in their answer. These values are then averaged over all examples.

Match_{oo4}: This adds together values for the previous metric across the model’s top four predictions.

Mode_{best}: The fraction of cases in which the top model prediction is the same as the most common annotator response, when there is a single most common response.

Mode_{oo4}: The fraction of cases in which one of the top four model predictions is the same as the most common annotator response.

Note that only 25 out of the 39 examples had a unique most common response. Of these, 20 had a chord chosen by three or four annotators, and five had a chord chosen by five to seven annotators. The rest of the examples are not considered in the *Mode* metrics.

Pitch Matches: The metrics above penalise all differences in predictions equally, even though some chord pairs are more different than others, e.g., A7 and A differ only in the addition of a pitch, whereas B and A share no pitches. To address this, we use a metric that is the total number of pitches that match for each question between the model’s top response and the annotator’s first response. We calculate this separately per-annotator, and then average across annotators (PM_{ave}).

Loss and Perplexity(PPL): These are two measures from the language modeling literature that we apply to see how well the models do on the true content of songs. Note that this evaluation is over a different set: 8,748 randomly chosen songs that are not included in model training.

	Match		Mode		PM		Match		Mode		PM
	best	oo4	best	oo4	ave		best	oo4	best	oo4	ave
All						Beginner					
NI	7.52	30.10	32.00	64.00	48.33	NI	5.61	22.44	0.00	44.44	40.50
PR	7.49	29.64	24.00	72.00	51.11	PR	5.61	22.44	11.11	33.33	43.00
CE _{cbow}	7.45	29.82	16.00	64.00	47.78	CE _{cbow}	4.97	19.87	0.00	22.22	39.50
CE _{sglm}	7.60	30.22	12.00	68.00	49.00	CE _{sglm}	5.29	21.15	0.00	33.33	42.00
Intermediate						Expert					
NI	8.17	32.68	33.33	71.43	48.80	NI	7.92	31.67	28.57	85.71	55.00
PR	8.19	32.42	23.81	71.43	53.60	PR	7.72	30.26	28.57	85.71	53.00
CE _{cbow}	8.04	32.14	14.29	66.67	47.20	CE _{cbow}	8.56	34.23	42.86	85.71	57.50
CE _{sglm}	8.34	33.19	14.29	71.43	50.00	CE _{sglm}	8.15	32.18	14.29	57.14	53.50

Table 1. Results for all models when compared with different sets of annotators based on their expertise.

5.3 Results

Match and Mode. Table 1 reports metrics for each system when compared with the beginner set, intermediate set, expert set, and full set of annotators. When evaluating against all annotators, CE_{sglm} is best in $Match_{best}$ and $Match_{oo4}$, NI is best in $Mode_{best}$, and PR is best in $Mode_{oo4}$.

In the results for subsets of annotators, all systems tend to match experts better than the beginner or intermediate groups. In particular, CE_{cbow} obtains the lowest scores when evaluated against the beginner group, but the highest when evaluated against the expert group. To investigate this pattern, we compute $Match_{best}$ and $Match_{oo4}$ for each individual annotator and then the Spearman Correlation Coefficient r_s between these metrics and their expertise. In Table 2, we observe strong significant correlation between the CE_{cbow} and expertise, and no significant correlation for the other models. This can be explained by the fact

that the models are trained on a large collection of songs composed by experts, and the chord embeddings seem to capture the chord use style of the experts.

Pitch Matches. We report PM_{ave} for the models and expertise groups in Table 1. Similarly to the *Match* and *Mode* metrics, we observe that all models perform better when compared to annotators with higher expertise, and the differences between groups is most extreme with CE_{cbow} . Table 2 shows the results of correlation analysis for pitch matches (Pearson Correlation Coefficient r_p), with a significant linear correlation for only CE_{cbow} . This trend is visually shown in Figure 3.

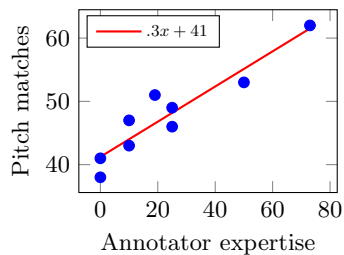


Fig. 3. Total pitch matches between annotator and output of CE_{cbow} model, plotted by the annotator’s expertise. The red line is the line of best fit computed by linear regression.

	<i>best</i> r_s	p-val	<i>oo4</i> r_s	p-val	<i>PM</i> r_p	p-val
NI	0.44	0.24	0.44	0.24	0.71	0.03
PR	0.64	0.06	0.57	0.11	0.43	0.25
CE_{cbow}	0.72	0.03	0.72	0.03	0.94	2e-4
CE_{sglm}	0.41	0.28	0.41	0.28	0.42	0.27

Table 2. Correlation coefficients for expertise and Match and Pitch Match metrics.

	Test Loss	Test PPL
NI	1.42	4.16
PR	1.44	4.20
CE_{cbow}	1.44	4.22
CE_{sglm}	1.42	4.15

Table 3. Loss and perplexity metrics for the chord prediction models on a held-out test set.

Loss and Perplexity. Table 3 shows the results on our 8,748 song test set. All models perform similarly in this setting.

5.4 Discussion

We observe that automatic models produce predictions that resemble human responses. Comparing against annotators grouped by expertise, CE_{cbow} compares best to the high expertise group for all metrics, while the best model varies

among the other groups. CE_{cbow} 's predictions also correlate significantly with pitch matches and annotator expertise. NI and PR achieve the highest Mode_{best} and Mode_{oo4} scores, however, fewer samples are considered because only twenty-five had a unique *mode* chord across the annotators and only five of these samples had more than half the annotators agree. Additionally, the chord symbol based metrics are strict, requiring an exact match on chords, and had lower inter-annotator agreement than the pitch-based metrics.

While CE_{cbow} 's predictions exhibit a strong pitch-match correlation, CE_{sglm} 's predictions exhibit no significant correlate at all. However, differences between the CE_{cbow} and CE_{sglm} embeddings may not be as apparent in other downstream applications; in fact, by the perplexity and test loss metrics shown in Table 3, there is barely a difference between these two, or any, models. Investigating key differences between these embedding models in musical contexts is a direction for future work.

6 Case Study Two: Artist Attribute Prediction

Our second case study introduces the task of performing artist attribute prediction, demonstrating that these chord representations could be used more broadly in tasks involving musical stylometry and musical author profiling. With binary classifiers using our chord representations, we predict three attributes as separate tasks: gender (male or female), performing country (U.S. or U.K.), and type of artist (group or solo artist).

Data. For these experiments, we augment the dataset with information obtained with the MusicBrainz API,¹¹ which includes the song artist's location, gender, lifespan, tags that convey musical genres, and other available information for 35,351 English songs (identified using Google's Language Detection project [29]). From this extracted information, we choose artist type, performing country, and gender because of the sufficient quantity of data available with these attributes enabling the tasks; we note that tasks dedicated to genre or time period are of interest for future investigations, and our preliminary experiments using the artists' lifespan and tags as proxies for time period and genre indicated these tasks are promising use cases for chord embeddings.

We use the top two most frequent classes of each attribute, and balance the data to have the same number of examples for each class. For artist type, there are 20,000 songs per class (group and solo). For performing country, there are 8,000 songs per class (U.S. and U.K.). For gender, there are 6,000 songs per class (male and female). The number of samples varies because of differences in the raw class counts and because not all songs have a label for each property.

Experimental Setup. We build two binary classifiers and compare their performance with each chord representation. The first uses logistic regression (LR) over a single vector for each song by aggregating chord representations. We also experimented with an SVM classifier, but LR was more efficient with minimal

¹¹ <https://musicbrainz.org/>

performance trade-offs. The BOC methods are defined in Section 4.2. The PR method aggregates the chords with a many-hot encoding vector counting each chord pitch, normalized by the total number of pitches. The CE methods aggregate chord embeddings by max-pooling, taking the most extreme absolute value in each dimension across all chords.

The second classifier is a Convolutional Neural Network (CNN) that considers the chords in sequences. We experimented with an LSTM, and found that the CNN functions better for these tasks. We use the CNN model for sentence classification by Kim [13]¹² over the chord progressions for each song, using the same NI, PR, CE_{sglm} , and CE_{cbow} representations from the first case study.

For our model parameters, chosen in preliminary experiments on a subset of the data, we use L2 regularization for the LR classifier, and the CNN model uses filter window sizes 3, 4, 5 with 30 feature maps, drop-out rate 0.5 and Adam [14] optimization. The sequence limit is 60 chords, cutting off extra chords and padding when there are fewer.

Model	Gender	Country	Artist	Type	Model	Gender	Country	Artist	Type
<i>Logistic regression</i>					<i>CNN</i>				
BOC_{count}	*†† 57.53	†55.71	*† 57.04		NI	†58.93	†56.79		†59.20
BOC_{tfidf}	†55.94	†55.17	†56.06						
PR	52.32	53.13	53.75		PR	57.98	55.31		57.54
CE_{cbow}	*†56.90	†55.37	*†56.92		CE_{cbow}	58.67	†57.30		†58.92
CE_{sglm}	†56.37	† 55.85	*†56.88		CE_{sglm}	† 58.95	† 57.54		† 59.29

* $p < 0.05$ over BOC_{tfidf} , † $p < 0.05$ over PR, ‡ $p < 0.05$ over CE_{sglm} , § $p < 0.05$ over NI

Table 4. Accuracy scores from 10-fold cross validation in artist gender, country, and type prediction tasks. The significance tests are performed among the logistic regression models and CNN models separately.

6.1 Results and Analyses

Table 4 shows the models’ accuracy scores from experiments using 10-fold cross validation. CE_{sglm} CNN is the top performer for all tasks, significantly outperforming CNN PR and all LR models for all tasks, and NI for country.¹³ All models outperform a random baseline (50%) significantly in each task. In each task, CNN models significantly outperform their LR counterparts.

For insight into the models’ performance, we analyze the gender prediction task, the only attribute where the LR CE_{cbow} and LR BOC_{count} predictions differed

¹² CNN model is built on <https://github.com/Shawn1993/cnn-text-classification-pytorch>

¹³ By a paired t-test with $p < .05$.

significantly. First, we compare the rate of use of each chord between genders. To show the differences in Figure 4a, we divide the higher rate by the lower rate, subtract one to set equal use to zero, and flip the sign when female use is higher. We observe greater variations among chords with lower song frequency. For instance, *C/G*, *F#7*, *Bbm*, and *Ab* are twice as salient for one gender than the other. The highest variation among the top 20 chords reaches 1.5 times more salient, and for the top 10, 1.2 times more salient for one gender.

To investigate the impacts of the musical relationships captured in embeddings (Section 4) to the CE models, we also compared use of five *chord qualities*. Figure 4b shows higher relative frequency of suspended and diminished chords among the songs of male artists, augmented and minor chords among the female artist songs, and fairly similar use of major chords.

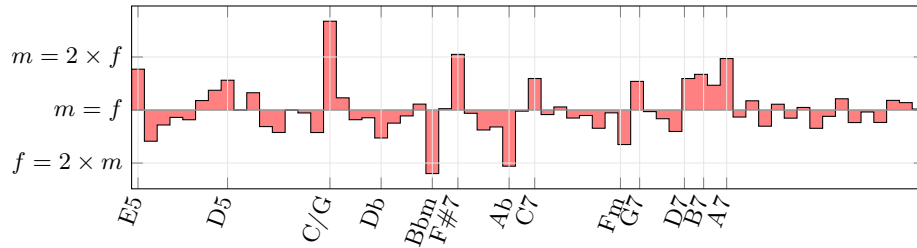


Fig. 4. Chord variations by gender, ordered from least to greatest by song frequency. The labeled chords are at least 1.5 more salient for one gender than the other.

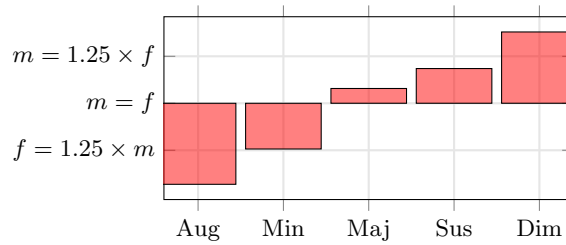


Fig. 5. Variation in chord quality usage by gender by ratio of use percentage.

6.2 Discussion

To our knowledge, this is the first time that an association between chord representations and author attributes has been explored. Each model for each attribute showed significant improvement over a random baseline of 50%, indicating there are quantifiable differences in our music data between the genders,

countries, and artist types. In addition to the tasks we presented, we also observed improvement when using the system to predict the life-period of the artist and their associated genres. As life-period is a proxy for the music’s time period, chord embeddings could benefit future work in musical historical analysis.

For gender, the variation of rare chords may contribute to BOC_{count} outperforming CE_{cbow} . However, CE_{cbow} significantly outperforms BOC_{tfidf} which gives more weight to rare chords by their inverse-document frequency. This suggests that chord rarity is not the only critical feature. The variations of chord quality use may contribute to the performance of the CE models as the embeddings capture musical relationships.

LR PR consistently underperforms all other models, which may indicate the importance chord structures. Different chords with the same pitches (e.g., G and G/B) have the same PR vector. Chords with overlapping pitches have similar PR vectors. However, CNN PR, which performs closer to the others, encodes pitch orderings (Section 4.2) and BOC methods encode chord symbols which indicate structure. CE representations are learned from chord symbols, likely capturing contextual functions of chord structures. These functions would matter for the CNN models which make predictions from chord sequences rather than a single aggregated vector. Since we observed the best performance by CNN CE_{sglm} , there suggests the importance of contextual semantics of chord structures. A deeper study into structural semantics captured by chord embeddings is a direction for future work.

7 Conclusion

In this paper, we presented an analysis of the information captured by chord embeddings and explored how they can be applied in two case studies. We found that chord embeddings capture chord similarities that are consistent with important musical relationships described in music theory. Our case studies showed that the embeddings are beneficial when integrated in models for downstream computational music tasks. Together, these results indicate that chord embeddings are another useful NLP tool for musical studies. The code to train chord embeddings and the resulting embeddings, as well as the next-chord annotations are publicly available from <https://lit.eecs.umich.edu/downloads.html>.

Acknowledgements. We would like to thank the anonymous reviewers and the members of the Language and Information Technologies lab at Michigan for their helpful suggestions. We are grateful to MeiXing Dong and Charles Welch for helping with the design and interface of the next-chord annotation task. This material is based in part upon work supported by the Michigan Institute for Data Science, and by Girls Encoded and Google for sponsoring Jiajun Peng through the Explore Computer Science Research program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Michigan Institute for Data Science, Girls Encoded or Google.

References

1. Absolu, B., Li, T., Ogihara, M.: Analysis of chord progression data. In: *Advances in Music Information Retrieval*, pp. 165–184. Springer (2010)
2. Brinkman, A., Shanahan, D., Sapp, C.: Musical stylometry, machine learning and attribution studies: A semi-supervised approach to the works of josquin. In: *Proc. of the Biennial Int. Conf. on Music Perception and Cognition*. pp. 91–97 (2016)
3. Brunner, G., Wang, Y., Wattenhofer, R., Wiesendanger, J.: Jambot: Music theory aware chord based generation of polyphonic music with lstms. In: *Proc. of ICTAI* (2017)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proc. of NAACL* (2019)
5. Feld, S., Fox, A.A.: Music and language. *Annual review of anthropology* **23**(1), 25–53 (1994)
6. Fell, M., Sporleder, C.: Lyrics-based analysis and classification of music. In: *Proc. of COLING* (2014)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: *Proc. of KDD* (2016)
8. Hillewaere, R., Manderick, B., Conklin, D.: Melodic models for polyphonic music classification. In: *Second International Workshop on Machine Learning and Music* (2009)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
10. Hontanilla, M., Pérez-Sancho, C., Inesta, J.M.: Modeling musical style with language models for composer recognition. In: *Iberian Conference on Pattern Recognition and Image Analysis*. pp. 740–748. Springer (2013)
11. Jäncke, L.: The relationship between music and language. *Frontiers in psychology* **3**, 123 (2012)
12. Kaliakatsos-Papakostas, M.A., Epitropakis, M.G., Vrahatis, M.N.: Musical composer identification through probabilistic and feedforward neural networks. In: *European Conference on the Applications of Evolutionary Computation*. pp. 411–420. Springer (2010)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1746–1751 (2014)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *ICLR*, 2015. (2014)
15. Madjiheurem, S., Qu, L., Walder, C.: Chord2vec: Learning musical chord embeddings. In: *Proc. of the Constructive Machine Learning Workshop* (2016)
16. Mayer, R., Rauber, A.: Musical genre classification by ensembles of audio and lyrics features. In: *Proc. of ISMIR* (2011)
17. McCarthy, D., Navigli, R.: The english lexical substitution task. *Language resources and evaluation* **43**(2), 139–159 (2009)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *ICLR* (2013)
19. Ogihara, M., Li, T.: N-gram chord profiles for composer style representation. In: *ISMIR*. pp. 671–676 (2008)
20. Owen, H.: *Music theory resource book*. Oxford University Press, USA (2000)
21. Patel, A.D.: Language, music, syntax and the brain. *Nature neuroscience* **6**(7), 674 (2003)

22. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of EMNLP (2014)
23. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
24. Phon-Amnuaisuk, S.: Exploring music21 and gensim for music data analysis and visualization. In: International Conference on Data Mining and Big Data. pp. 3–12. Springer (2019)
25. Randel, D.M.: The Harvard concise dictionary of music and musicians. Harvard University Press (1999)
26. Saker, M.N.: A theory of circle of fifths progressions and their application in the four Ballades by Frederic Chopin. Ph.D. thesis, University of Wisconsin-Madison (1992)
27. Sergeant, D.C., Himonides, E.: Gender and music composition: A study of music, and the gendering of meanings. *Frontiers in Psychology* **7**, 411 (2016). <https://doi.org/10.3389/fpsyg.2016.00411>, <https://www.frontiersin.org/article/10.3389/fpsyg.2016.00411>
28. Shepherd, J.: A theoretical model for the sociomusicological analysis of popular musics. *Popular music* **2**, 145–177 (1982)
29. Shuyo, N.: Language detection library for java (2010), <http://code.google.com/p/language-detection/>
30. Stamatatos, E.: A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* **60**(3), 538–556 (2009). <https://doi.org/10.1002/asi.21001>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001>
31. Wołkiewicz, J., Kulka, Z., Kešelj, V.: N-gram-based approach to composer recognition. *Archives of Acoustics* **33**(1), 43–55 (2008)