A HISTORICAL ANALYSIS OF HARMONIC PROGRESSIONS USING CHORD EMBEDDINGS

Elia ANZUONI (elia.anzuoni@epfl.ch) [1*], **Sinan AYHAN** (sinan.ayhan@epfl.ch) [1*],
Federico DUTTO (federico.dutto@epfl.ch) [1*],
Andrew MCLEOD (andrew.mcleod@epfl.ch) (0000-0003-2700-2076) [1],
Fabian C. MOSS (fabian.moss@epfl.ch) (0000-0001-9377-2066) [1], and
Martin ROHRMEIER (martin.rohrmeier@epfl.ch) [1]

[1] *Digital and Cognitive Musicology Lab*, *Digital Humanities Institute*, **École Polytechnique Fédérale de Lausanne**, Switzerland
[*] Authors 1, 2, and 3 contributed equally to this work.

ABSTRACT

This study focuses on the exploration of the possibilities arising from the application of an NLP word-embedding method (Word2Vec) to a large corpus of musical chord sequences, spanning multiple musical periods. First, we analyse the clustering of the embedded vectors produced by Word2Vec in order to probe its ability to learn common musical patterns. We then implement an LSTM-based neural network which takes these vectors as input with the goal of predicting a chord given its surrounding context in a chord sequence. We use the variability in prediction accuracy to quantify the stylistic differences among various composers in order to detect idiomatic uses of some chords by some composers. The historical breadth of the corpus used allows us to draw some conclusions about broader patterns of changing chord usage across musical periods from Renaissance to Modernity.

1. INTRODUCTION

Algorithmic approaches to music usually come in two flavors: music information retrieval (MIR) aims at extracting relevant patterns from musical signals (e.g. audio recordings, MIDI files, or images of scores) and improve the performance on certain specific tasks, such as genre or composer classification, automatic playlist generation, optical music recognition and more. Computational music analysis, on the other hand, aims at using data-driven methods to study the domain of music in order to develop a deeper understanding of its cultural and historical diversity, or implications for its perception and cognition.

This study bridges the two approaches by applying the machine-learning (ML) methods often employed for the task of chord prediction in MIR to a large corpus of symbolic chord sequences. However, our goal is not to globally optimize chord prediction in this dataset. Rather, we use the chord-prediction task as a benchmark measure for investigating stylistic characteristics of different composers in the dataset. We suppose that the historical dimension in particular affects stylistic differences which, in turn, should be reflected in the performance of a (globally constant) chord predictor. In other words, assuming a fixed model for chord prediction, how does its performance change given historically varying input? What conclusions can we draw from this perspective?

In the remainder of the paper, we first summarize recent related work (Section 2). We then describe the dataset used in our study (Section 3), as well as our specific application of the three ML approaches in more detail (Section 4). We report two important results (Section 5): that clustering in an embedding space reveals functional relations between chords, and that changes in performance of our chord-prediction model (dependent on composer and historical time) indicate fundamental changes in the usage of harmony.

2. RELATED WORK

Our study draws on a dataset of symbolic musical chord sequences and uses three fundamental machine learning building blocks: word embeddings, clustering, and Recurrent Neural Networks (RNNs).

Word embedding is a popular technique in Natural Language Processing (NLP) which learns a mapping of words to vectors in a low-dimensional *embedding space* from a *corpus* of texts, which is supposed to contain sufficient information on the semantic relationships between words. The mapping is such that the relative positions of the vectors (hopefully) reflect these semantic relationships. The precise learning of this mapping is dependent on the specific method used. We use Word2Vec [1]. In Word2Vec, words often appearing in similar contexts are mapped to close points in the embedding space, according to their cosine distance.

Previous work has used Word2Vec successfully for modeling aspects of the musical language. In [2], the authors show that a simple approach of splitting musical scores into short slices containing note presence information is able to capture some simple features such as tonal proximity. Later, in [3], a similar slicing procedure is used on a larger corpus that re-affirms Word2Vec's ability to model musical concepts such as tonal relationships between musical keys. In [4], the authors learn an embedding space in a similar way, but use as input multi-hot vectors of note presence, rather than one-hot encodings of unique symbols (as in the standard Word2Vec). In contrast to these efforts, our work takes annotated chord symbols as input, thus enabling us to model information at a much higher level of abstraction by eliminating information spurious to the harmonic structure such as short passing tones and ornamentation.

Clustering is a well-known unsupervised learning primitive, which works by grouping together close points in a space, and is used to extract information about the points that might be contained in their coordinates. We use hierarchical clustering [5] with cosine distance to analyze the structural properties of our resulting chord embedding space. This hierarchical approach (as opposed to a more naive clustering approach like K-means [6]) has the benefit of allowing us to investigate clusters at different levels of granularity without needing to fine-tune any hyperparameters. Previous work has also investigated the clustering of musical embeddings, using explicitly trained chordal embeddings (e.g., [2,3]), chord clusters induced through training for a different task (e.g., [7, 8]), or clustering of larger groups of chords (e.g., [9]).

RNNs are widespread tools in NLP, particularly in the field of word prediction with their Long Short-Term Memory (LSTM) [10] variant. LSTMs are particularly suited to this task because of their structure, involving a *forget gate*, which solves the *short-term memory* problem, typical of traditional RNNs. Similar work shows how they can be successfully employed in musical contexts, for "next-slice" modeling [4, 11], as well as for chord prediction [7, 12], and cadence identification [13]. While the cited works try to maximize prediction accuracy as much as possible, our goal is slightly different. Of course, we do want the models to perform as well as possible, but our main focus in the current work is instead to investigate the change in prediction performance across historical time (enabled by our expansive corpus), and to try draw musicological conclusions from this.

3. DATA

The dataset at our disposal, used for embedding, clustering, and chord prediction, consists of 4045 chord progressions in pieces by 24 Western classical composers, spanning the wide historical range from the Renaissance to 20th-century Modernism. The data has been derived from harmonic annotations using the syntax presented in [14–17]. For this study, the labels have been simplified in order to decrease the size of the chord vocabulary and to remove sparsity in our data. The pieces have been partitioned into local key segments that are either in the major or the minor *mode* (i.e., they contain no modulations), and chords are expressed relative to the tonic of that mode. Specifically, chords are represented by their *root* (expressed as a Roman numeral referring to the scale degree of the mode) and their *quality* (major, minor, diminished, or augmented; 7th chords are reduced to their corresponding triad). Because of this representation, the chord vocabulary is *potentially* infinite because the seven scale degrees of the two modes can be preceded by arbitrarily many accidentals. In particular, this allows us to distinguish enharmonically equivalent triads, such as `#II:MAJ` and `bIII:MAJ` that may entail different harmonic functions. Applied chords have been reduced to be directly related to the tonic of the mode, e.g. "vii°/V" is translated to "♯iv°" and represented as `#IV:DIM`. Thus, the chord sequences in our dataset are of the form

- `MAJOR;I:MAJ,II:MIN,V:MAJ,...`, or
- `MINOR;I:MIN,II:DIM,III:MAJ,...`,

where mode and chord labels are separated by a semicolon and chords within a progression are separated by commas. The average length of a chord sequence is 31 chords for major sequences and 28 chords for minor sequences. Since the roots of chords are expressed in relative notation, i.e. as the distance to the tonic, an F major chord is represented as `IV:MAJ` if the chord sequence is in C major, but as `III:MAJ` if it is in D minor. Following these reductions, there are 81 distinct chords in major sequences, and 77 different chords in minor sequences in our data.

As one can observe in Figure 1, the amount of data at our disposal varies greatly across composers and historical periods. Note, for example, that no chord sequences in the major mode are available for Sweelinck. Great care has thus to be taken when generalizing our results to the entire œuvre of these composers or the historical periods they represent. The data is available at `https://github.com/DCMLab/chordembeddings-smc2021`.

4. METHODOLOGY

4.1 Chord embedding

Our first processing step, serving as a basis for the two downstream tasks of clustering and chord prediction, is the application of Word2Vec [1]—specifically its implementation in the Gensim library [18]—which takes as input "sentences" (in our case, major or minor sequences) of "words" (in our case, chord labels). We treat major and minor chord sequences as independent and never include chord sequences from both modes in conjunction. Thus, in the following, when we say "train/test on all sentences/sections of a composer" or "train/test on a composer", we implicitly mean that those sections are all in the same mode.

Word2Vec has four hyperparameters to tune: `size`, `window`, `sg` (skip-gram), and `min_count`. `size` determines the dimension of the embedding space. To avoid overfitting, it should be less than the size of the vocabulary, i.e. the number of distinct chords in the corpus. In our case, the vocabulary size varies considerably, between 20 and 100 chord types per composer within either of the two modes. `window` defines the "width" of the context, i.e. how many chords, to the left and to the right, constitute the context of the current chord. The binary parameter `sg`

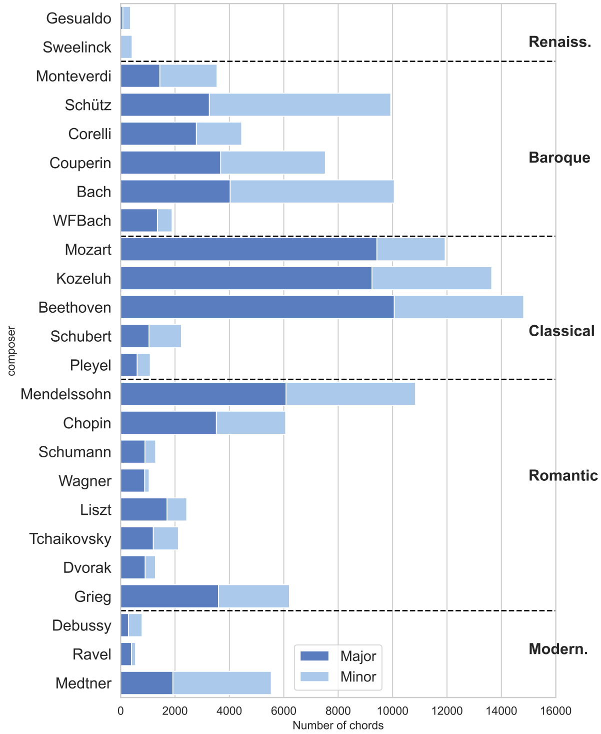

Figure 1. Total number of non-unique chord labels used by each composer, split between major and minor sequences. Composers are ordered by year of death (from oldest at the top to more recent at the bottom).

is short for "skip-gram" and selects the training algorithm: it can be either "continuous bag of words" (CBOW, i.e. guessing the target word from its context), or "skip-gram" (guessing the context given the target word). `min_count` sets a minimum absolute frequency a chord must have in order to be kept in the corpus. Since our corpus contains a Zipf-like distribution, this allows us to remove from the result the numerous irrelevant mappings of rare chords.

For all of our experiments, we exclude rare chords, as the model is unable to learn a stable embedding for such chords, making any relevant conclusion impossible. We therefore set `min_count` $= 50$ (since the most common chords have absolute frequencies of hundreds, if not thousands), which led to a vocabulary size of 32. The `size` of the embedding space was then chosen to be 5 (alternatives were essentially equivalent). We set `window` $= 2$ (again, other values led to similar results), and finally, we chose to use skip-gram rather than CBOW embeddings, because this led to more interpretable results.

4.2 Clustering

A first application of the mapping learned by Word2Vec is clustering, which is used to detect musical patterns. As is understandable from the properties of the mapping, chords appearing in the same cluster are likely to often appear in similar contexts. For this task, it is very difficult to carry out an objective, quantitative model evaluation. Therefore, we choose hyperparameters based on how much the outcome corresponds to music-theoretical intuitions. For example, we expect, when only training on major sections, that tonics and dominants are embedded close to each other, since they constitute the most basic musical pattern imaginable, as discussed in [15], and therefore often occur in very similar contexts.

Hierarchical clustering works by recursively merging the pair of clusters C_i and C_j (starting from singletons) that are the closest to each other according to some distance metric. We use *cosine distance*, commonly used for vector embedding spaces. The recursion stops when the minimum distance between clusters is above a given `distance_threshold`, or when only a single cluster remains.

The fact that this algorithm can work with cosine distance is ideal to detect similarities in a Word2Vec embedding space. Moreover, it is able to capture clusters of any shape. One might argue that a choice of `distance_threshold` can be quite arbitrary. However, this can be avoided by setting the `distance_threshold` to some large value (thus merging all clusters into one), and then plotting a dendrogram of all possible mergers. A dendrogram (e.g., Figure 3) is a depiction of the nested clusters produced by this method: it clearly shows all the mergers $C_i - C_j$ that happened, and the distance associated to them.

4.3 Chord prediction

Another use of the mapping provided by Word2Vec is the chord prediction task. LSTMs are an improvement over the classic RNN design that solve its *short-term memory* problem (caused by the well-known *vanishing gradient* problem): this allows them to effectively track long-term dependencies in sequential data. They are commonly used in NLP to predict the next word in a sentence.

We implemented an LSTM-based neural network for chord prediction, which trains on a *training corpus* (all sentences from a set of *training composers* for a given mode) and is tested on a *test corpus* (all sentences from a single *test composer* for that mode). For the LSTM experiments, the Word2Vec embedding is retrained using only the training corpus. Thus, we test how well-predictable chords in musical sequences by a composer are given the knowledge about chord sequences by all other composers. The metric used is the simple accuracy: the fraction of correctly-predicted chord occurrences, either overall or grouped by chord. We use the overall accuracy results for a single test composer to see how "predictable" they are, from what we learned from the training composers. We use the same results, split by chord, to investigate which chords are easier to predict and which are used more idiomatically (and are thus more difficult to predict).

The LSTM design is shown in Figure 2, and is structured as follows. Given a target chord (c_n in the figure), a first LSTM layer takes as input the concatenation of the embedded vectors of chords within some window of the target chord (shown as black circles in the figure with a window size of 2). A linear layer then maps the LSTM's output vector to a vector of length `n_vocab` (where `n_vocab` is the number of distinct chords), with a final softmax activa-

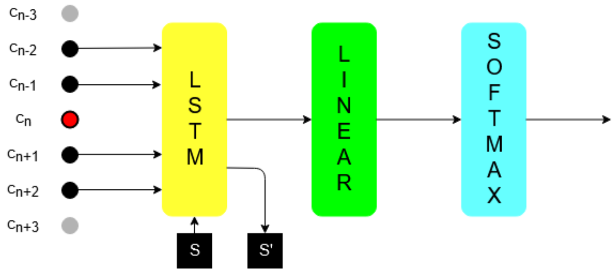

Figure 2. Diagram for the predictor network.

tion.

For the chord prediction experiments, we use the same Word2Vec parameters as above, although the embeddings are recalculated for each test composer. For both training and testing the LSTM, we take care to remove any data points which contain any chord (either as input or as the target) that falls below Word2Vec's `min_count` (in the training corpus). For training the LSTM, we use the Adam optimizer [19] with mean squared error (MSE) for the loss. We train all results for 2 epochs (this was enough for them to converge in all cases).

5. RESULTS

Our results imply two main findings: 1) clustering chords in the embedding space reveals meaningful functional relations between many of them; 2) chord prediction accuracy exhibits historical trends.

5.1 Clustering reveals functional chord relations

First, we report the results we obtained by applying hierarchical clustering on the embedded chords from the major and minor sections of all composers in the corpus. We visualize the hierarchical clustering in the embedding spaces for the major and the minor mode in dendrograms in Figures 3 and 4, respectively. As mentioned before, distances in embedding spaces are inherently difficult to interpret in general. However, many of the resulting clusters are quite well interpretable in various ways.

The resulting clusters for chord sequences for both modes reveal two fundamental tonal relations: functional *equivalence* and functional *difference* [20–22]. This extends earlier similar findings on functional categories restricted to J. S. Bach's chorales and based on chord bigrams [23]. Below we list a number of notable functional chord relations that can be found in our clusterings.

5.1.1 Functional equivalence

Chords that share common tones may be regarded as functionally equivalent. Functionally equivalent chords include *relative* and *parallel* chords, as well as other *common-tone* relations [24]. Two chords are each other's relative if they are the tonics of two keys that have the same key signature (e.g. `V:MAJ` and `III:MIN` in a major key). A major and minor chord are parallel if they have the same root (e.g. `II:MAJ` and `II:MIN`). Chords may also retain the same function, if they share a number of tones (e.g. `V:MAJ` and

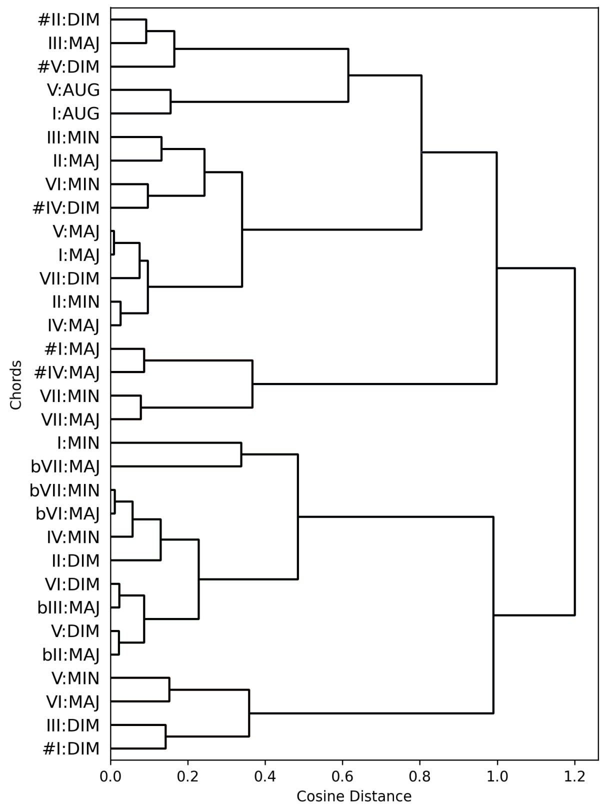

Figure 3. Dendrogram for chord embeddings in major.

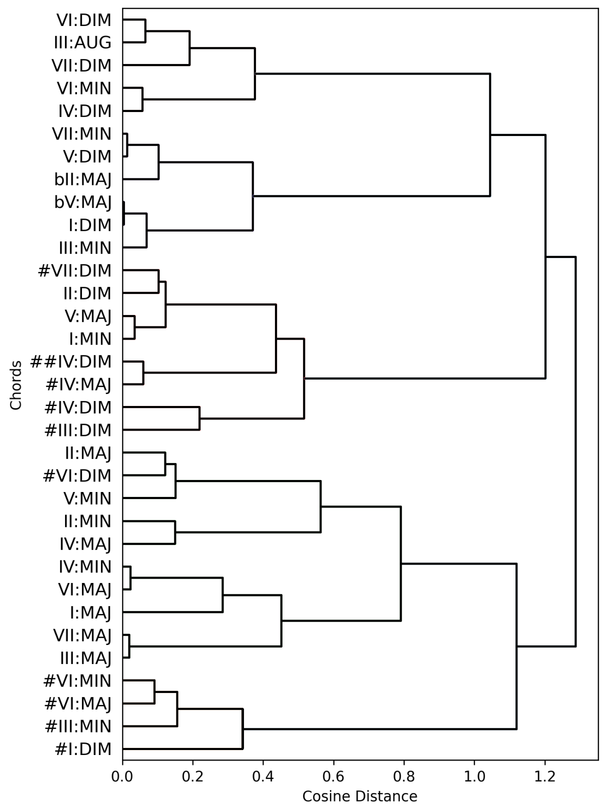

Figure 4. Dendrogram for chord embeddings in minor.

#VII:DIM jointly form a dominant seventh chord in any minor key).

In the major mode (Figure 3), the relative chords that are clustered together are II:MIN and IV:MAJ as well as IV:MIN and bVI:MAJ. The parallel chords are VII:MAJ and VII:MIN, and the chords involved in other common-tone relations are V:MAJ and VII:DIM; II:MAJ and #IV:DIM; II:DIM and IV:MIN; III:MAJ and #V:DIM; VI:MIN and #IV:DIM as well as III:DIM and #I:DIM.

In minor (Figure 4), the relative chords close to one another in the embedding space are VII:MIN and bII:MAJ; IV:MAJ and II:MIN; as well as IV:MIN and VI:MAJ. The parallel chords in minor are #VI:MAJ and #VI:MIN; and, finally, the chords with other common-tone relations are II:DIM and #VII:DIM; I:DIM and III:MIN; V:DIM and VII:MIN; #III:MIN and #I:DIM; as well as #IV:MAJ and ##IV:DIM.

Overall, in our chord embeddings, the relative and common-tone relations are much more frequent than parallel relations, which is to be expected, since the latter involves a change of mode and the sections from which the chords are drawn are precisely defined as staying within one mode (major or minor, notwithstanding potential singular exceptions).

5.1.2 *Functional difference*

Chords are functionally different if they, or their equivalents, are separated by a perfect fifth, as for example in tonic-dominant or tonic-subdominant pairs, e.g. in authentic or plagal progressions. Note, however, that pairs of chords in the embedding space are undirected. In the major mode (Figure 3), we find fifth-based relations between chords in the embedding space for I:MAJ and V:MAJ; I:AUG and V:AUG; III:MAJ and #II:DIM;[1] as well as #IV:MAJ and #I:MAJ. In the minor mode (Figure 4), we find I:MIN and V:MAJ; II:MAJ and V:MIN; #VI:MAJ/MIN and #III:MIN; I:MAJ and IV:MIN; as well as III:MAJ and VII:MAJ

It is notable that the main cadential chords in both modes (i.e. triads on scale degrees I, V, IV, II, and VII in major, and I, V, and II in minor) occur in relatively close proximity. Despite the fact that distances in embedding spaces are generally hard to interpret, we take the ubiquity of relative, parallel, subset, and fifths-based relations to be an indicator for their pervasiveness in the harmonic progressions in our corpus.

5.2 Chord prediction indicates historical differences in harmonic styles

Here, we summarise the results obtained in chord prediction. Since a composer's prediction accuracy may change for each run of our algorithm due to random initialization of the Word2Vec and LSTM models, we run each experiment ten times, and report mean and standard deviation values for each composer. These are plotted in Figure 5, per composer and mode, where each point represents the mean accuracy for all chords combined, and the shaded bands show the standard deviation across the ten runs. The composers are ordered by their year of death in order to investigate historical trends.

The first thing to notice is that the standard deviations are all quite small (< 0.04 in all cases), showing that our results are consistent across runs and are not affected by random noise in the modeling process. Furthermore, the approximately "inverted U-shape" of the mean values implies that Classical composers are the most predictable from our data, followed by Baroque and Romantic composers, with Modernist and Renaissance composers being the least predictable. This is not to say that Classical composers are more predictable *in general* than composers from other eras. Indeed, remembering that for each composer we train on the data from all other composers in the corpus, this trend is roughly implied by the distribution of data shown in Figure 1. However, the very fact that such an effect exists suggests that composers of the different eras do use chords in fundamentally different ways. Since each model is trained on a very similar set of data (differing by only one composer), the learned model is necessarily similar across composers. Therefore, if two composers used chords similarly, their results would likewise be extremely similar. So, the fact that we see a historical trend *at all* suggests that composers of the different eras do indeed use chords in fundamentally different ways (although we make no claim here about what those differences are).

Furthermore, since the majority of our data comes from Classical composers, we can hypothesize that the mean accuracy of a composer should be positively correlated with the similarity of that composer's chord usage to that of an average Classical composer. From this perspective, the overall shape of the curve makes a lot of sense.

An analysis of the detailed per-chord accuracy results (data available with the code), gives even more insight about the idioms common to a specific composer or period. The strongest result, in a major context, is the very low prediction accuracy for I:MAJ and V:MAJ (the easiest chords to predict overall) when testing on Ravel and Debussy. Indeed, they are two Impressionist composers, who are generally known for their "distinct" harmonies, which rarely (if ever) use authentic cadences. Moreover, we find IV:MAJ and II:MIN to be two "polarising" chords: for most composers, we either predict them very well or very poorly compared to the average. In particular, IV:MAJ is only well predictable for Baroque composers, while others (with the exception of Beethoven, Chopin, and Dvořak) seem to use it in a more peculiar way. II:MIN, on the other hand, only becomes hard to predict from the late Romantic period. This latter result, albeit neat and striking, is not as easily interpretable as the previous one. In minor sections, a low accuracy on I:MIN (the most common chord together with V:MAJ) for Renaissance composers (Gesualdo, Sweelinck, Monteverdi, Schütz) and for Modernists, again signals that this chord has played diverse roles across the centuries. We achieve a relatively low accuracy on many of the most common minor chords for both Romantic and Modernist composers, with the exception

[1] We interpret #II:DIM as a shortened VII:DOM7.

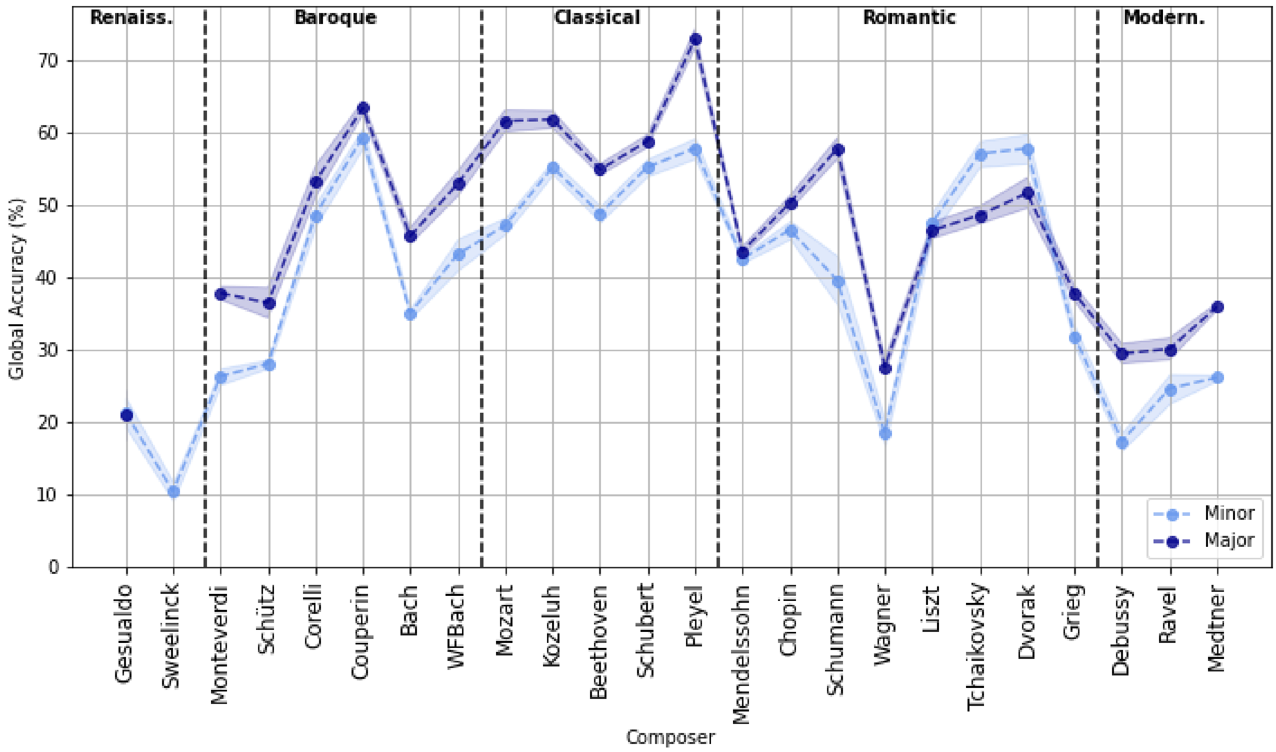

Figure 5. Global chord prediction accuracy for each composer, for major and minor sections. Standard deviation is given by the shaded region around each point. Composers are ordered chronologically by year of death.

of Tchaikovsky. This indicates that he is closer to Classical composers in his works in minor contexts (indeed, his only work in the dataset are the *Seasons*, a collection of rather traditional piano pieces overall). Changes in chord predictability related to stylistic differences are supported by historical studies focusing on the pitch-class content of musical pieces [25, 26].

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated progressions of chords in both the major and minor mode by a number of different composers. Our study explored two applications of deep learning methods to music theory: which inferences about tonal relations between chords could be drawn from embedding them in a lower-dimensional space, and whether attempting to predict chords based on the regularities in the data would reveal stylistic differences between composers across historical periods. All data and code are available at `https://github.com/DCMLab/chordembeddings-smc2021`.

Word2Vec was our first processing step, which provided useful grounds to base our subsequent analyses on. When applied to the output vectors of Word2Vec, clustering could capture some well-known tonal relationships between chords, including relative, parallel, and subset relations, as well as (possibly transposed) tonic-dominant pairs of chords. On the other hand, LSTM-based chord prediction yielded fairly high accuracy results in general (roughly 50% for most composers), but it also allowed us to use their high variability across chords and composers to draw some conclusions about chord usage across time which are supported by music theory. Globally, we found that Classical and Baroque composers use chords in a similar way, while Modernists and Renaissance composers seem to have a more distinctive style. The Romantic style seems to be complex, as there is a high variance in how composers from that era use chords.

Future work might also include a more refined use of clustering, for instance by applying it to a Word2Vec model trained only on a single composer—or on a group of composers which are known to be relatively similar to each other—in order to detect some special tonal relationship unique to that set of composers. Alternatively, chord prediction could be employed to investigate how rigidly a given composer belongs to a given artistic era: by restricting the training corpus to composers in the same era, we would prevent the model from learning totally unrelated idioms, thus achieving a higher accuracy on the test composer (to an extent depending on how similar he actually is to the others in that era).

As mentioned, in the current work, we identified the existence of historical differences in chord usage. However, we did not identify what those differences were. Future work could look at the problem from a more causal perspective by limiting the training corpus for each composer to only those composers who preceded them.

Acknowledgments

Research supported through the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811).

7. REFERENCES

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013, pp. 3111–3119.

[2] D. Herremans and C.-H. Chuan, "Modeling musical context with word2vec," in *Proceedings of the First International Conference on Deep Learning and Music*, 2017, pp. 11–18.

[3] C.-H. Chuan, K. Agres, and D. Herremans, "From context to concept: exploring semantic relationships in music with word2vec," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1023–1036, 2020.

[4] S. Madjiheurem, L. Qu, and C. Walder, "Chord2vec: Learning musical chord embeddings," in *Proceedings of the constructive machine learning workshop at 30th conference on neural information processing systems (NIPS2016), Barcelona, Spain*, 2016.

[5] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[6] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available: http://www.jstor.org/stable/2346830

[7] F. Korzeniowski and G. Widmer, "Improved chord recognition by combining duration and harmonic language models," in *ISMIR*, 2018.

[8] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust tonnetz-space transform for automatic chord recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 453–456.

[9] B. Duane and J. Jakubowski, "Harmonic clusters and tonal cadences: Bayesian learning without chord identification," *Journal of New Music Research*, vol. 47, no. 2, pp. 143–165, 2018.

[10] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, pp. 2451–71, 10 2000.

[11] A. Ycart, A. McLeod, E. Benetos, and K. Yoshii, "Blending acoustic and language model predictions for automatic music transcription," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 454–461.

[12] K. Landsnes, L. Mehrabyan, V. Wiklund, R. Lieck, F. C. Moss, and M. Rohrmeier, "A model comparison for chord prediction on the Annotated Beethoven Corpus," in *Proceedings of the 16th Sound & Music Computing Conference. Málaga, Spain*, 2019.

[13] L. Feisthauer, L. Bigo, and M. Giraud, "Modeling and learning structural breaks in sonata forms," in *ISMIR*, 2019.

[14] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, "The Annotated Beethoven Corpus (ABC): A dataset of harmonic analyses of all Beethoven string quartets," *Frontiers in Digital Humanities*, vol. 5, p. 16, 2018.

[15] F. C. Moss, M. Neuwirth, D. Harasim, and M. Rohrmeier, "Statistical characteristics of tonal harmony: A corpus study of beethoven's string quartets," *PLOS ONE*, vol. 14, no. 6, pp. 1–16, 06 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0217242

[16] F. C. Moss, "Transitions of tonality: A model-based corpus study," Ph.D. dissertation, EPFL, 2019.

[17] J. Hentschel, M. Neuwirth, and M. Rohrmeier, "The Annotated Mozart Sonatas: Score, Harmony, and Cadence," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 1–14, 2021.

[18] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, 2017.

[20] Z. Gárdonyi and H. Nordhoff, *Harmonik*. Wolfenbüttel: Möseler Verlag, 2002.

[21] F. C. Moss, "Tonality and functional equivalence: A multi-level model for the cognition of triadic progressions in 19th century music," in *International Conference of Students of Systematic Musicology - Proceedings*, vol. 1, London, 2014, pp. 1–8.

[22] M. Rohrmeier, "The Syntax of Jazz Harmony: Diatonic Tonality, Phrase Structure, and Form," *Music Theory and Analysis (MTA)*, vol. 7, no. 1, pp. 1–63, Apr. 2020.

[23] M. Rohrmeier and I. Cross, "Statistical Properties of Tonal Harmony in Bach's Chorales," in *Proceedings of the 10th International Conference on Music Perception and Cognition*, 2008, pp. 619–627.

[24] H. Riemann, *Vereinfachte Harmonielehre oder die Lehre von den tonalen Funktionen der Akkorde*. London: Augener, 1893.

[25] C. Weiß, M. Mauch, S. Dixon, and M. Müller, "Investigating style evolution of Western classical music: A computational approach," *Musicae Scientiae*, vol. 23, no. 4, pp. 486–507, 2019.

[26] D. Harasim, F. C. Moss, M. Ramirez, and M. Rohrmeier, "Exploring the foundations of tonality: Statistical cognitive modeling of modes in the history of Western classical music," *Humanities and Social Sciences Communications*, vol. 8, no. 1, 2021.