

# Modeling Musical Context Using Word2vec

D. Herremans\*<sup>1</sup> and C.-H. Chuan†<sup>2</sup>

<sup>1</sup>Queen Mary University of London, London, UK

<sup>2</sup>University of North Florida, Jacksonville, USA

We present a semantic vector space model for capturing complex polyphonic musical context. A word2vec model based on a skip-gram representation with negative sampling was used to model slices of music from a dataset of Beethoven’s piano sonatas. A visualization of the reduced vector space using t-distributed stochastic neighbor embedding shows that the resulting embedded vector space captures tonal relationships, even without any explicit information about the musical contents of the slices. Secondly, an excerpt of the Moonlight Sonata from Beethoven was altered by replacing slices based on context similarity. The resulting music shows that the selected slice based on similar word2vec context also has a relatively short tonal distance from the original slice.

**Keywords:** music context, word2vec, music, neural networks, semantic vector space

## 1 Introduction

In this paper, we explore the semantic similarity that can be derived by looking solely at the context in which a musical slice appears. In past research, music has often been modeled through Recursive Neural Networks (RNNs) combined with Restricted Boltzmann Machines [Boulanger-Lewandowski et al., 2012], Long-Short Term RNN models [Eck and Schmidhuber, 2002, Sak et al., 2014], Markov models [Conklin and Witten, 1995] and other statistical models, using a representation that incorporates musical information (i.e., pitch, pitch class, duration, intervals, etc.). In this research, we focus on modeling the *context*, over the content.

Vector space models [Rumelhart et al., 1988] are typically used in natural language processing (NLP) to represent (or embed) words in a continuous vector space [Turney and Pantel, 2010, McGregor et al., 2015, Agres et al., 2016, Liddy et al., 1999]. Within this space, semantically similar words are represented geographically close to each other [Turney and Pantel, 2010]. A recent very efficient approach to creating these vector spaces for natural language processing is word2vec [Mikolov et al., 2013c].

---

\*d.herremans@qmul.ac.uk

†c.chuan@unf.edu

Although music is not the same as language, it possesses many of the same types of characteristics. Besson and Schön [2001] discuss the similarity of music and language in terms of, among others, structural aspects and the expectancy generated by both a word and a note. We can therefore use a model from NLP: word2vec. More specifically a skip-gram model with negative sampling is used to create and train a model that captures musical context.

There have been only few attempts at modeling *musical* context with semantic vector space models. For example, Huang et al. [2016] use word2vec to model chord sequences in order to recommend chords other than the ‘ordinary’ to novice composers. In this paper, we aim to use word2vec for modeling musical context in a more generic way as opposed to a reduced representation as chord sequences. We represent complex polyphonic music as a sequence of equal-length slices without any additional processing for musical concepts such as beat, time signature, chord tones and etc.

In the next sections we will first discuss the implemented word2vec model, followed by a discussion of how music was represented. Finally, the resulting model is evaluated.

## 2 Word2vec

Word2vec refers to a group of models developed by Mikolov et al. [2013c]. They are used to create and train semantic vector spaces, often consisting of several hundred dimensions, based on a corpus of text [Mikolov et al., 2013a]. In this vector space, each word from the corpus is represented as a vector. Words that share a context are geographically close to each other in this space. The word2vec architecture can be based on two approaches: a continuous bag-of-words, or a continuous skip-gram model (CBOW). The former uses the context to predict the current word, whereas the latter uses the current word to predict surrounding words [Mikolov et al., 2013b]. Both models have a low computational complexity, so they can easily handle a corpus with a size ranging in the billions of words in a matter of hours. While CBOW models are faster, it has been observed that skip-gram performs better on small datasets [Mikolov et al., 2013a]. We therefore opted to work with the latter model.

**Skip-gram with negative sampling** The architecture of a skip-gram model is represented in Figure 1. For each word  $w_t$  in a corpus of size  $T$  at position  $t$ , the network tries to predict the surrounding words in a window  $c$  ( $c = 2$  in the figure). The training objective is thus defined as:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq i \leq c, i \neq 0} \log p(w_{t+i}|w_t), \quad (1)$$

whereby the term  $p(w_{t+i}|w_t)$  is calculated by a softmax function. Calculating the gradient of this term is, however, computationally very expensive. Alternatives to circumvent this problem include hierarchical softmax [Morin and Bengio, 2005] and noise contrastive estimation [Gutmann and Hyvärinen, 2012]. The word2vec model used in this research implements a variant of the latter, namely negative sampling.

The idea behind negative sampling is that a well trained model should be able to distinguish between data and noise [Goldberg and Levy, 2014]. The original training objective is thus approximated by a new, more efficient, formulation that implements a binary logistic regression to classify between data and noise samples. When the model is able to assign high probabilities

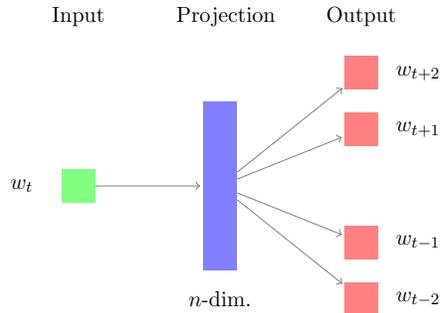


Figure 1: A skip-gram model with  $n$ -dimensions for word  $w_t$  at position  $t$ .

to real words and low probabilities to noise samples, the objective is optimized Mikolov et al. [2013c].

**Cosine similarity** was used as a similarity metric between two musical-slice vectors in our vector space. For two non-zero vectors A and B in  $n$  dimensional space, with an angle  $\theta$ , it is defined as [Tan et al., 2005]:

$$\text{Similarity}(A, B) = \cos(\theta) = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad (2)$$

In this research, we port the above discussed model and techniques to the field of music. We do this by replacing ‘words’ with ‘slices of polyphonic music’. The manner in which this is done is discussed in the next section.

### 3 Musical slices as words

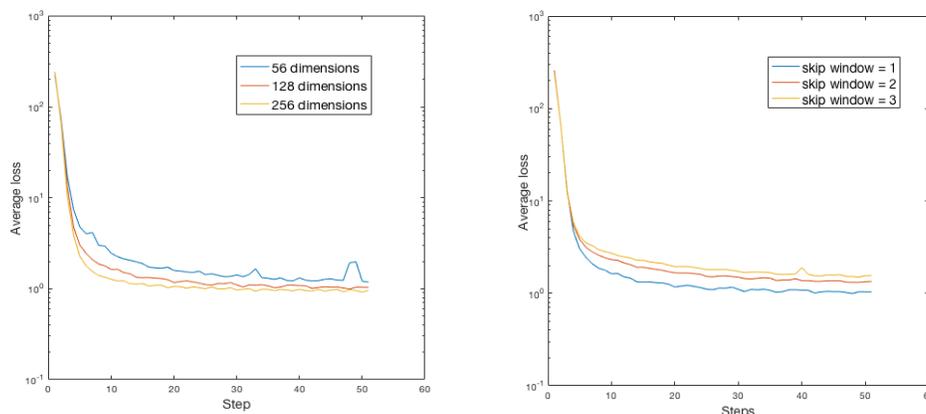
In order to study the extend to which word2vec can model *musical* context, polyphonic musical pieces are represented with as little injected musical knowledge as possible. Each piece is simply segmented into equal-length, non-overlapping slices. The duration of these slices is calculated for each piece based on the distribution of time between note onsets. The smallest amount of time between consecutive onsets that occurs in more than 5% of all cases is selected as the slice-size. The slices capture all pitches that sound in a slice: those that have their onset in the slice, and those that are played and held over the slice. The slicing process does not depend on musical concepts such as beat or time signature; instead, it is completely data-driven. Our vocabulary of words, will thus consist of a collection of musical slices.

In addition, we do not label pitches as chords. All sounding pitches, including chord tones, non-chord tones, and ornaments, are all recorded in the slice. We do not reduce pitches into pitch classes either, i.e., pitches  $C_4$  and  $C_5$  are considered different pitches. The only musical knowledge we use is the global key, as we transpose all pieces to either C major or A minor before segmentation. This enables the functional role of pitches in tonality to stay the same across compositions, which in turn causes there to be more repeated slices over the dataset and allows the model to be better trained on less data. In the next section, the performance of the resulting model is discussed.

## 4 Results

In order to evaluate how well the proposed model captures musical context, a few experiments were performed on a dataset consisting of Beethoven’s piano sonatas. The resulting dataset consists of 70,305 words, with a total of 14,315 unique occurrences. As discussed above, word2vec models are very efficient to train. Within minutes, the model was trained on the CPU of a MacBook Pro.

We trained the model a number of times, with a different number of dimensions of the vector space (see Figure 2a). The more dimensions there are, the more accurate the model becomes, however, the time to train the model also becomes longer. In the rest of the experiments, we decided to use 128 dimensions. In a second experiment, we varied the size of the skip window, i.e., how many words to consider to the left and right of the current word in the skip-gram. The results are displayed in Figure 2b, and show that a skip window of 1 is most ideal for our dataset.



(a) Results for varying the number of dimensions of the vector space. (b) Results for varying the size of the skip window.

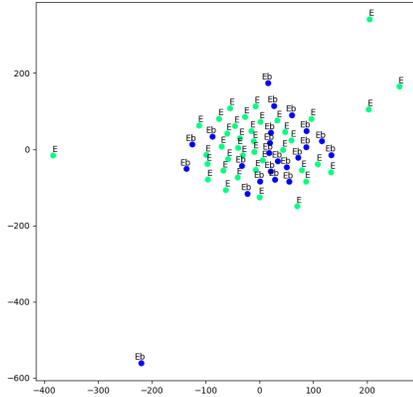
Figure 2: Evolution of the average loss during training. A step represents 2000 training windows.

### 4.1 Visualizing the semantic vector space

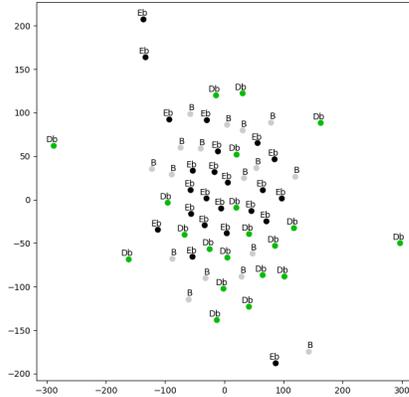
In order to better understand and evaluate the proposed model, we created visualizations of selected musical slices in a dimensionally reduced space. We use t-Distributed Stochastic Neighbor Embedding (t-SNE), a technique developed by Maaten and Hinton [2008] for visualizing high-dimensional data. t-SNE has previously been used in a music analysis context for visualizing clusters of musical genres based on musical features [Hamel and Eck, 2010].

In this case, we identified the ‘chord’ to which each slice of the dataset belongs based on a simple template-matching method. We expect that tonally close chords occur together in the semantic vector space. Figure 3 confirms this hypothesis. When examining slices that contain C and G chords (a perfect fifth apart), the space looks very dispersed, as they often co-occur (see Figure 3c). The same occurs for the chord pair  $E^b$  and  $B^b$  in Figure 3d. On the other

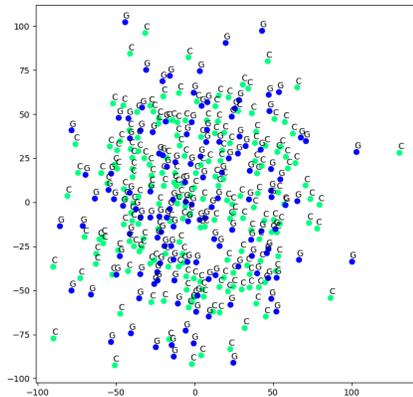
hand, when looking at the tonally distant chord pair E and E<sup>b</sup> (Figure 3a), we see that clusters appear in the reduced vector space. The same happens for the tonally distant chords E<sup>b</sup>, B<sup>b</sup> and B in Figure 3b.



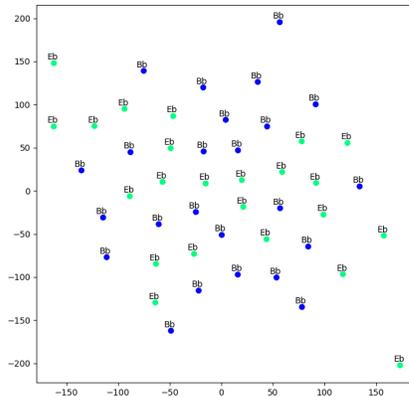
(a) E (green) and Eb (blue).



(b) Eb (black), Db (green) and B (gray).



(c) C (green) and G (blue).



(d) Eb (green) and Bb (blue).

Figure 3: Reduced vector space with t-SNE for different slices (labeled by the most close chord)

## 4.2 Content versus context

In order to further examine if word2vec captures semantic meaning in music via the modeling of context, we modify a piece by replacing some of its original slices with the most similar one as captured by the cosine similarity in the vector space model. If word2vec is really able to capture this, the modified piece should sound similar to the original. This allows us to evaluate the effectiveness of using word2vec for modeling music.

Figure 4 shows the first 17 measures of Beethoven’s piano sonata Op. 27 No. 2 (Moonlight),

2nd movement in (a) and the measures with modified pitch slices in the dashed box in (b). An audio version of this score is available online<sup>1</sup>. The modified slices in (b) are produced by replacing the original with the slice that has the highest cosine similarity based on the word2vec embeddings. The tonal distance between the original and modified slices is presented below each slice pair. This is calculated as the average of the number of steps between each pair of pitches in the two slices in a tonnetz representation [Cohn, 1997], extended with pitch register. It can be observed that even though the cosine similarity is around 0.5, the tonal distance of the selected slice remains relatively low in most of the cases. For example, the tonal distance in the third dashed box between the modified slice ( $D^b$  major triad with pitches  $D^b_4$ ,  $F_4$ , and  $A^b_4$ ) and the original slice of a single pitch  $A^b_4$  is 1.25. However, we notice that word2vec does not necessarily model musical context for voice leading. For example, better voice leading can be achieved if the pitch  $D_4$  in the last dashed box is replaced with pitch  $D_5$ .



Figure 4: (a) An excerpt of Beethoven’s piano sonata Op. 27 No. 2, 2nd movement with (b) modified measures by replacing with slices that report the highest word2vec cosine similarity.

In Figure 4b, a number of notes are marked in a different color (orange). These are the held notes, i.e., their onsets are played previously and the notes remain being played over the current slice. These notes create a unique situation in music generation using word2vec. For example, the orange note (pitch  $D^b_5$ ) in the first dashed box is a held note, which indicates that the pitch should have been played in the previous slice. However, word2vec does not capture this relation; it only considers the similarity between the original and modified slices.

<sup>1</sup><http://dorienherremans.com/word2vec>

## 5 Conclusions

A skip-gram model with negative sampling was used to build a semantic vector space model for complex polyphonic music. By representing the resulting vector space in a reduced two-dimensional graph with t-SNE, we show that musical features such as a notion of tonal proximity are captured by the model. Music generated by replacing slices based on word2vec context similarity also presents close tonal distance compared to the original.

In the future, an embedded model that combines both word2vec with, for instance, a long-short term memory recurrent neural network based on musical features, would offer a more complete way to more completely model music. The TensorFlow code used in this research is available online<sup>2</sup>.

## Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 658914.

## References

- Kat R Agres, Stephen McGregor, Karolina Rataj, Matthew Purver, and Geraint A Wiggins. 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*, pages 1–14, 2016.
- Mireille Besson and Daniele Schön. Comparison between language and music. *Annals of the New York Academy of Sciences*, 930(1):232–258, 2001.
- Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.
- Richard Cohn. Neo-riemannian operations, parsimonious trichords, and their” tonnetz” representations. *Journal of Music Theory*, 41(1):1–66, 1997.
- Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- Douglas Eck and Juergen Schmidhuber. 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*, pages 747–756. IEEE, 2002.
- Yoav Goldberg and Omer Levy. word2vec explained: Deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
- Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361, 2012.

---

<sup>2</sup><http://dorienherremans.com/word2vec>

- Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344. Utrecht, The Netherlands, 2010.
- Cheng-Zhi Anna Huang, David Duvenaud, and Krzysztof Z Gajos. Chordripple: Recommending chords to help novice composers go beyond the ordinary. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 241–250. ACM, 2016.
- Elizabeth D Liddy, Woojin Paik, S Yu Edmund, and Ming Li. Multilingual document retrieval system and method using semantic vector matching, December 21 1999. US Patent 6,006,221.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- Stephen McGregor, Kat Agres, Matthew Purver, and Geraint A Wiggins. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *Journal of Artificial General Intelligence*, 6(1):55–86, 2015.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.
- Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005.
- David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.
- Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pages 338–342, 2014.
- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.
- Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.