First published in the Proceedings of the 26th European Signal Processing Conference (EUSIPCO-2018) in 2018, published by EURASIP.

Automatic Chord Recognition with Higher-Order Harmonic Language Modelling

Filip Korzeniowski and Gerhard Widmer
Institute of Computational Perception,
Johannes Kepler University, Linz, Austria
Email: filip.korzeniowski@jku.at

***Abstract*—Common temporal models for automatic chord recognition model chord changes on a frame-wise basis. Due to this fact, they are unable to capture musical knowledge about chord progressions. In this paper, we propose a temporal model that enables explicit modelling of chord changes and durations. We then apply N-gram models and a neural-network-based acoustic model within this framework, and evaluate the effect of model overconfidence. Our results show that model overconfidence plays only a minor role (but target smoothing still improves the acoustic model), and that stronger chord language models do improve recognition results, however their effects are small compared to other domains.**

***Index Terms*—Chord Recognition, Language Modelling, N-Grams, Neural Networks**

Research on automatic chord recognition has recently focused on improving frame-wise predictions of acoustic models [1]–[3]. This trend roots in the fact that existing temporal models just smooth the predictions of an acoustic model, and do not incorporate musical knowledge [4]. As we argue in [5], the reason is that such temporal models are usually applied on the audio-frame level, where even non-Markovian models fail to capture musical properties.

We know the importance of language models in domains such as speech recognition, where hierarchical grammar, pronunciation and context models reduce word error rates by a large margin. However, the degree to which higher-order language models improve chord recognition results yet remains unexplored. In this paper, we want to shed light on this question. Motivated by the preliminary results from [5], we show how to integrate chord-level harmonic language models into a chord recognition system, and evaluate its properties.

Our contributions in this paper are as follows. We present a probabilistic model that allows for combining an acoustic model with explicit modelling of chord transitions and chord durations. This allows us to deploy language models on the *chord level*, not the frame level. Within this framework, we then apply N-gram chord language models on top of an neural network based acoustic model. Finally, we evaluate to which degree this combination suffers from acoustic model over-confidence, a typical problem with neural acoustic models [6].

This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 Framework Programme (ERC Grant Agreement number 670035, project "Con Espressione").

I. Problem Definition

Chord recognition is a sequence labelling problem similar to speech recognition. In contrast to the latter, we are also interested in the start and end points of the segments. Formally, assume $\mathbf{x}_{1:T}$[1] is a time-frequency representation of the input signal; the goal is then to find $y_{1:T}$, where $y_t \in \mathcal{Y}$ is a chord symbol from a chord vocabulary $\mathcal{Y}$, such that y_t is the correct harmonic interpretation of the audio content represented by $\mathbf{x}_t$. Formulated probabilistically, we want to infer

$$\hat{y}_{1:T} = \underset{y_{1:T}}{\operatorname{argmax}} P(y_{1:T} \mid \mathbf{x}_{1:T}). \tag{1}$$

Assuming a generative structure where $y_{1:T}$ is a left-to-right process, and each $\mathbf{x}_t$ only depends on y_t,

$$P(y_{1:T} \mid \mathbf{x}_{1:T}) \propto \prod_t \frac{1}{P(y_t)} P_A\left(y_t \mid \mathbf{x}_t\right) P_T\left(y_t \mid y_{1:t-1}\right),$$

where the $1/P(y_t)$ is a label prior that we assume uniform for simplicity [7], $P_A(y_t \mid \mathbf{x}_t)$ is the *acoustic model*, and $P_T\left(y_t \mid y_{1:t-1}\right)$ the *temporal model*.

Common choices for P_T (e.g. Markov processes or recurrent neural networks) are unable to model the underlying musical language of harmony meaningfully. As shown in [5], this is because modelling the symbolic chord sequence on a frame-wise basis is dominated by self-transitions. This prevents the models from learning higher-level knowledge about chord changes. To avoid this, we disentangle P_T into a chord *language model* P_L, and a chord *duration model* P_D.

The chord language model is defined as $P_L\left(\bar{y}_i \mid \bar{y}_{1:i-1}\right)$, where $\bar{y}_{1:i} = \mathcal{C}\left(y_{1:t}\right)$, and $\mathcal{C}\left(\cdot\right)$ is a sequence compression mapping that removes all consecutive duplicates of a symbol (e.g. $\mathcal{C}\left((a, a, b, b, a)\right) = (a, b, a)$). P_L thus only considers chord *changes*. The duration model is defined as $P_D\left(s_t \mid y_{1:t-1}\right)$, where $s_t \in \{\mathrm{s}, \mathrm{c}\}$ indicates whether the chord changes (c) or stays the same (s) at time t. P_D thus only considers chord *durations*. The temporal model is then formulated as:

$$P_T\left(y_t \mid y_{1:t-1}\right) = \begin{cases} P_L\left(\bar{y}_i \mid \bar{y}_{1:i-1}\right) P_D\left(\mathrm{c} \mid y_{1:t-1}\right) & \text{if } y_t \neq y_{t-1} \\ P_D\left(\mathrm{s} \mid y_{1:t-1}\right) & \text{else} \end{cases}. \tag{2}$$

To fully specify the system, we need to define the acoustic model P_A, the language model P_L, and the duration model P_D.

[1] We use the notation $\mathbf{v}_{i:j}$ to indicate $(\mathbf{v}_i, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_j)$.

II. Models

A. Acoustic Model

The acoustic model used in this paper is a minor variation of the one introduced in [8]. It is a VGG-style [9] fully convolutional neural network with 3 convolutional blocks: the first consists of 4 layers of 32 3×3 filters, followed by 2×1 max-pooling in frequency; the second comprises 2 layers of 64 such filters followed by the same pooling scheme; the third is a single layer of 128 12×9 filters. Each of the blocks is followed by feature-map-wise dropout with probability 0.2, and each layer is followed by batch normalisation [10] and an exponential linear activation function [11]. Finally, a linear convolution with 25 1×1 filters followed by global average pooling and a softmax produces the chord class probabilities $P_A(y_k \mid \mathbf{x}_k)$. The input to the network is a log-magnitude log-frequency spectrogram patch of 1.5 seconds. See [8] for a detailed description of the input processing and training schemes.

Neural networks tend to produce overconfident predictions, which leads to probability distributions with high peaks. This causes a weaker training signal because the loss function saturates, and makes the acoustic model dominate the language model at test time [6]. Here, we investigate two approaches to mitigate these effects: using a temperature softmax in the classification layer of the network, and training using smoothed labels.

The temperature softmax replaces the regular softmax activation function at test time with

$$\sigma(\mathbf{z})_j = \frac{e^{z_j/T}}{\sum_{k=1}^{K} e^{z_k/T}},$$

where $\mathbf{z}$ is a real vector. High values for T make the resulting distribution smoother. With $T = 1$, the function corresponds to the standard softmax. The advantage of this method is that the network does not need to be retrained.

Target smoothing, on the other hand, trains the network with with a smoothed version of the target labels. In this paper, we explore three ways of smoothing: *uniform smoothing*, where a proportion of $1 - \beta$ of the correct probability is assigned uniformly to the other classes; *unigram smoothing*, where the smoothed probability is assigned according to the class distribution in the training set [12]; and *target smearing*, where the target is smeared in time using a running mean filter. The latter is inspired by a similar approach in [13] to counteract inaccurate segment boundary annotations.

B. Language Model

We designed the temporal model in Eq. 2 in a way that enables chord changes to be modelled explicitly via $P_L(\bar{y}_k \mid \mathcal{C}(\bar{y}_{1:k-1}))$. This formulation allows to use all past chords to predict the next. While this is a powerful and general notion, it prohibits efficient exact decoding of the sequence. We would have to rely on approximate methods to find $\hat{y}_{1:T}$ (Eq. 1). However, we can restrict the number of past chords the language model can consider, and use higher-order Markov models for exact decoding. To achieve that, we use N-grams for language modelling in this work.

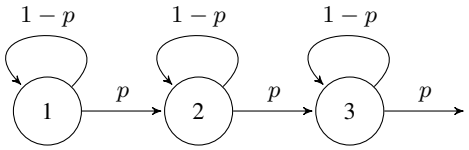


Fig. 1. Markov chain modelling the duration of a chord segment ($K = 3$). The probability of staying in one of the states follows a negative binomial distribution.

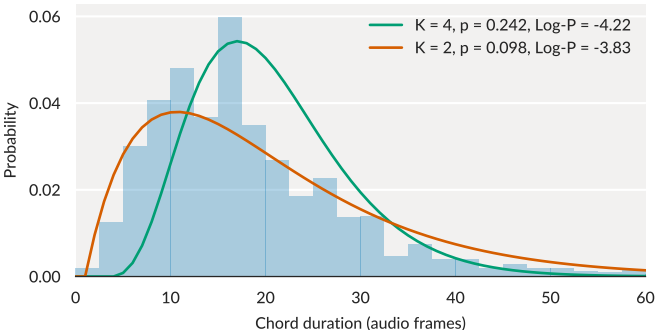


Fig. 2. Histogram of chord durations with two configurations of the negative binomial distribution. The log-probability is computed on a validation fold.

N-gram language models are Markovian probabilistic models that assume only a fixed-length history (of length $N - 1$) to be relevant for predicting the next symbol. This fixed-length history allows the probabilities to be stored in a table, with its entries computed using maximum-likelihood estimation (MLE)—i.e., by counting occurrences in the training set.

With larger N, the sparsity of the probability table increases exponentially, because we only have a finite number of N-grams in our training set. We tackle this problem using Lidstone smoothing, and add a pseudo-count α to each possible N-gram. We determine the best value for α for each model using the validation set.

C. Duration Model

The focus of this paper is on how to meaningfully incorporate chord language models beyond simple first-order transitions. We thus use only a simple duration model based on the negative binomial distribution, with the probability mass function

$$P(k) = \binom{k + K - 1}{K - 1} p^K (1 - p)^k,$$

where K is the number of failures, p the failure probability, and k the number of successes given K failures. For our purposes, $k + K$ is the length of a chord in audio frames.

The main advantage of this choice is that a negative binomial distribution is easily represented using only few states in a HMM (see Fig. 1), while still reasonably modelling the length of chord segments (see Fig. 2). For simplicity, we use the same duration model for all chords. The parameters (K, the number of states used for modelling the duration, and p, the probability of moving to the next state) are estimated using MLE.

D. Model Integration

If we combine an N-gram language model with a negative binomial duration model, the temporal model P_T becomes a Hierarchical Hidden Markov Model [14] with a higher-order Markov model on the top level (the language model) and a first-order HMM at the second level (see Fig. 3a). We can translate the hierarchical HMM into a first-order HMM; this will allow us to use many existing and optimised HMM implementations.

To this end, we first transform the higher-order HMM on the top level into a first-order one as shown e.g. in [15]: we factor the dependencies beyond first-order into the HMM state, considering that self-transitions are impossible as

$$\mathcal{Y}_N = \{(y_1, \ldots, y_N) : y_i \in \mathcal{Y}, y_i \neq y_{i+1}\},$$

where N is the order of the N-gram model. Semantically, $(y_1, \ldots, y_N)$ represents chord y_1, having seen $y_2, \ldots, y_N$ in the immediate past. This increases the number of states from $|\mathcal{Y}|$ to $|\mathcal{Y}| \cdot (|\mathcal{Y}| - 1)^{N-1}$.

We then flatten out the hierarchical HMM by combining the state spaces of both levels as $\mathcal{Y}_N \times [1..K]$, and connecting all incoming transitions of a chord state to the corresponding first duration state, and all outgoing transitions from the last duration state (where the outgoing probabilities are multiplied by p). Formally,

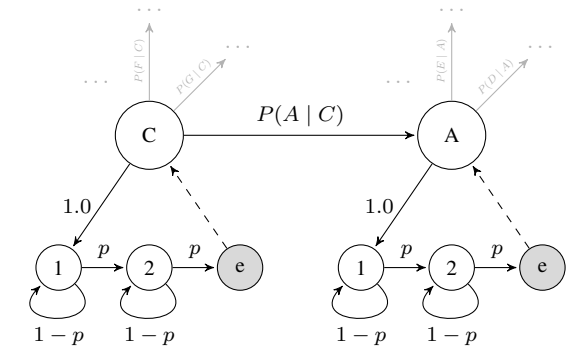
$$\mathcal{Y}_N^{(K)} = \{(\mathbf{y}, k) : \mathbf{y} \in \mathcal{Y}_N, k \in [1..K]\},$$

with the transition probabilities defined as

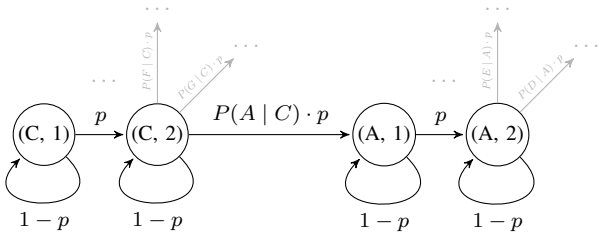
$$\begin{aligned} P((\mathbf{y}, k) \mid (\mathbf{y}, k)) &= 1 - p, \\ P((\mathbf{y}, k+1) \mid (\mathbf{y}, k)) &= p, \\ P((\mathbf{y}, 1) \mid (\mathbf{y}', K)) &= P_L(y_1 \mid y_{2:N}) \cdot p, \end{aligned}$$

where $y_{2:N} = y'_{1:N-1}$. All other transitions have zero probability. Fig. 3b shows the HMM from Fig. 3a after the transformation.

The resulting model is similar to a higher-order duration-explicit HMM (DHMM). The main difference is that we use a compact duration model that can assign duration probabilities using few states, while standard DHMMs do not scale well if longer durations need to be modelled (their computation increases by a factor of $D^2/2$, where D is the longest duration to be modelled [17]). For example, [16] uses first-order DHMMs to decode beat-synchronised chord sequences, with $D = 20$. In our case, we would need a much higher D, since our model operates on the frame level, which would result in a prohibitively large state space. In comparison, our duration models use only $K = 2$ (as determined by MLE) states to model the duration, which significantly reduces the computational burden.



(a) First-Order Hierarchical HMM.



(b) Flattened version of the First-Order Hierarchical HMM.

Fig. 3. Exemplary Hierarchical HMM and its flattened version. We left out incoming and outgoing transitions of the chord states for clarity (except C → A and the ones indicated in gray). The model uses 2 states for duration modelling, with "e" referring to the final state on the duration level (see [14] for details). Although we depict a first-order language model here, the same transformation works for higher-order models.

III. Experiments

Our experiments aim at uncovering (i) if acoustic model overconfidence is a problem in this scenario, (ii) whether smoothing techniques can mitigate it, and (iii) whether and to which degree chord language modelling improves chord recognition results. To this end, we investigated the effect of various parameters: softmax temperature $T \in \{0.5, 1.0, 1.3, 2.0\}$, smoothing type (uniform, unigram, and smear), smoothing intensity $\beta \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ and smearing width $w \in \{3, 5, 10, 15\}$, and the language model order $N \in \{2, 3, 4\}$.

The experiments were carried out using 4-fold cross-validation on a compound dataset consisting of the following sub-sets: **Isophonics[2]:** 180 songs by the Beatles, 19 songs by Queen, and 18 songs by Zweieck, 10:21 hours of audio; **RWC Popular [18]:** 100 songs in the style of American and Japanese pop music, 6:46 hours of audio; **Robbie Williams [19]:** 65 songs by Robbie Williams, 4:30 of audio; and **McGill Billboard [20]:** 742 songs sampled from the American billboard charts between 1958 and 1991, 44:42 hours of audio. The compound dataset thus comprises 1125 unique songs, and a total of 66:21 hours of audio.

We focus on the major/minor chord vocabulary (i.e. major and minor chords for each of the 12 semitones, plus a "no-chord" class, totalling 25 classes). The evaluation measure we are interested in is thus the weighted chord symbol recall of major and minor chords, WCSR $= t_c/t_a$, where t_c is the total time the our system recognises the correct chord, and t_a is the total duration of annotations of the chord types of interest.

[2]http://isophonics.net/datasets

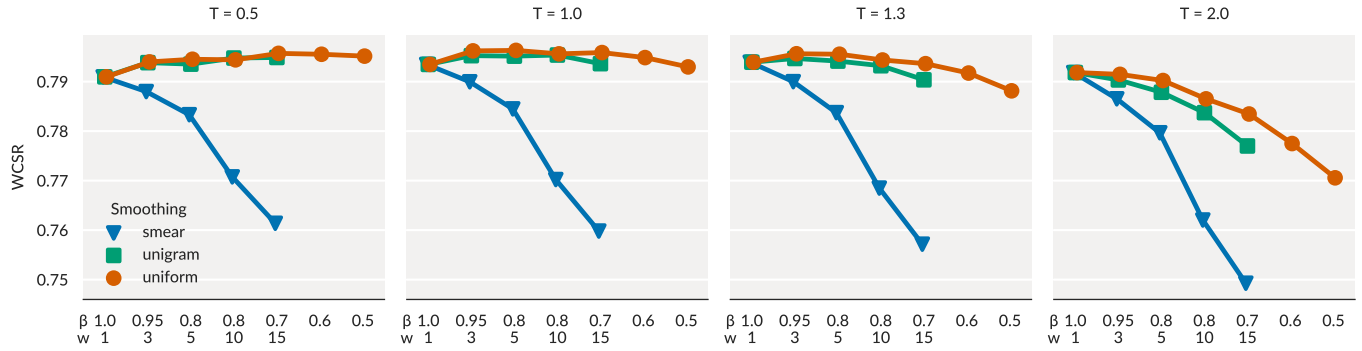


Fig. 4. The effect of temperature T, smoothing type, and smoothing intensity on the WCSR. The x-axis shows the smoothing intensity: for uniform and unigram smoothing, β indicates how much probability mass was kept at the true label during training; for target smearing, w is the width of the running mean filter used for smearing the targets in time. For these results, a 2-gram language model was used, but the outcomes are similar for other language models. The key observations are the following: (i) target smearing is always detrimental; (ii) uniform smoothing works slightly better than unigram smoothing (in other domains, authors report the contrary [6]); and (iii) smoothing improves the results, however, excessive smoothing is harmful in combination with higher softmax temperatures (a relation we explore in greater detail in Fig. 5).

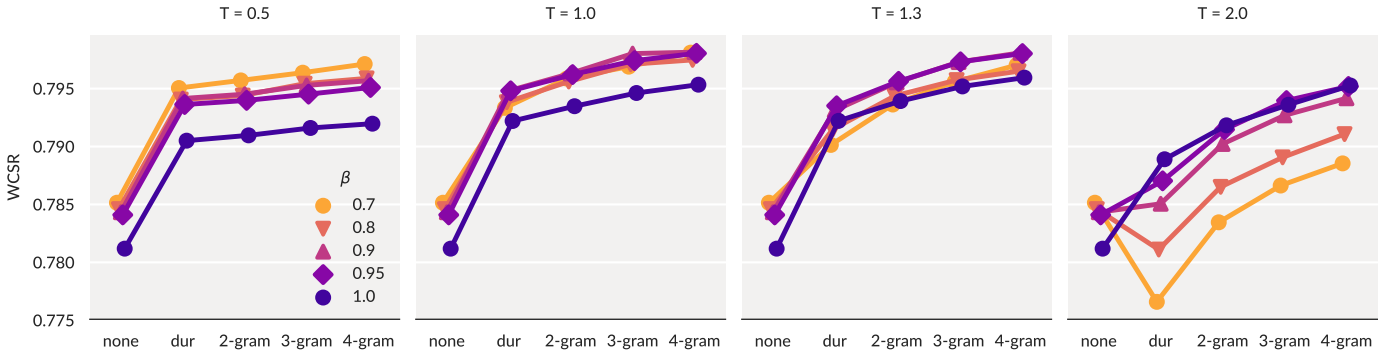


Fig. 5. Interaction of temperature T, smoothing intensity β and language model with respect to the WCSR. We show four language model configurations: *none* means using the predictions of the acoustic model directly; *dur* means using the chord duration model, but no chord language model; and *N-gram* means using the duration model with the respective language model. Here, we only show results using uniform smoothing, which turned out to be the best smoothing technique we examined in this paper (see Fig. 4). We observe the following: (i) Even simple duration modelling accounts for the majority of the improvement (in accordance with [16]). (ii) Chord language models further improve the results—the stronger the language model, the bigger the improvement. (iii) Temperature and smoothing interact: at $T = 1$, the amount of smoothing plays only a minor role; if we lower T (and thus make the predictions more confident), we need stronger smoothing to compensate for that; if we increase both T and the smoothing intensity, the predictions of the acoustic model are over-ruled by the language model, which shows to be detrimental. (iv) Smoothing has an additional effect during the training of the acoustic model that cannot be achieved using post-hoc changes in softmax temperature. Unsmoothed models never achieve the best result, regardless of T.

A. Results and Discussion

We analyse the interactions between temperature, smoothing, and language modelling in Fig. 4 and Fig. 5. Uniform smoothing seems to perform best, while increasing the temperature in the softmax is unnecessary if smoothing is used. On the other hand, target smearing performs poorly; it is thus not a proper way to cope with uncertainty in the annotated chord boundaries.

The results indicate that in our scenario, acoustic model overconfidence is not a major issue. The reason might be that the temporal model we use in this work allows for exact decoding. If we were forced to perform approximate inference (e.g. by using a RNN-based language model), this overconfidence could cut off promising paths early. Target smoothing still exhibits a positive effect during the training of the acoustic model, and can be used to fine-balance the interaction between acoustic and temporal models.

TABLE I
WCSR FOR THE COMPOUND DATASET. FOR THESE RESULTS, WE USE A SOFTMAX TEMPERATURE OF $T = 1.0$ AND UNIFORM SMOOTHING WITH $\beta = 0.9$.

None	Dur.	2-gram	3-gram	4-gram	5-gram
78.51	79.33	79.59	79.69	79.81	79.88

Further, we see consistent improvement the stronger the language model is (i.e., the higher N is). Although we were not able to evaluate models beyond $N = 4$ for all configurations, we ran a 5-gram model on the best configuration for $N = 4$. The results are shown in Table I.

Although consistent, the improvement is marginal compared to the effect language models show in other domains such as speech recognition. There are two possible interpretations of this result: (i) even if modelled explicitly, chord language

models contribute little to the final results, and the most important part is indeed modelling the chord duration; and (ii) the language models used in this paper are simply not good enough to make a major difference. While the true reason yet remains unclear, the structure of the temporal model we propose enables us to research both possibilities in future work, because it makes their contributions explicit.

Finally, our results confirm the importance of duration modelling [16]. Although the duration model we use here is simplistic, it improves results considerably. However, in further informal experiments, we found that it underestimates the probability of long chord segments, which impairs results. This indicates that there is still potential for improvement in this part of our model.

IV. Conclusion

We proposed a probabilistic structure for the temporal model of chord recognition systems. This structure disentangles a chord language model from a chord duration model. We then applied N-gram chord language models within this structure and evaluated various properties of the system. The key outcomes are that (i) acoustic model overconfidence plays only a minor role (but target smoothing still improves the acoustic model), (ii) chord duration modelling (or, sequence smoothing) improves results considerably, which confirms prior studies [4], [16], and (iii) while employing N-gram models also improves the results, their effect is marginal compared to other domains such as speech recognition.

Why is this the case? Static N-gram models might only capture global statistics of chord progressions, and these could be too general to guide and correct predictions of the acoustic model. More powerful models may be required. As shown in [21], RNN-based chord language models are able to adapt to the currently processed song, and thus might be more suited for the task at hand.

The proposed probabilistic structure thus opens various possibilities for future work. We could explore better language models, e.g. by using more sophisticated smoothing techniques, RNN-based models, or probabilistic models that take into account the key of a song (the probability of chord transitions varies depending on the key). More intelligent duration models could take into account the tempo and harmonic rhythm of a song (the rhythm in which chords change). Using the model presented in this paper, we could then link the improvements of each individual model to improvements in the final chord recognition score.

References

[1] F. Korzeniowski and G. Widmer, "Feature Learning for Chord Recognition: The Deep Chroma Extractor," in *17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, Aug. 2016.

[2] B. McFee and J. P. Bello, "Structured Training for Large-Vocabulary Chord Recognition," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, Oct. 2017.

[3] 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)*, Kyoto, Japan, 2012.

[4] T. Cho and J. P. Bello, "On the Relative Importance of Individual Components of Chord Recognition Systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 477–492, Feb. 2014.

[5] F. Korzeniowski and G. Widmer, "On the Futility of Learning Complex Frame-Level Language Models for Chord Recognition," in *Proceedings of the AES International Conference on Semantic Audio*, Erlangen, Germany, Jun. 2017.

[6] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv:1612.02695*, Dec. 2016.

[7] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist Probability Estimators in HMM Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, Jan. 1994.

[8] F. Korzeniowski and G. Widmer, "A Fully Convolutional Deep Auditory Model for Musical Chord Recognition," in *26th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Salerno, Italy, Sep. 2016.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556*, Sep. 2014.

[10] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167*, Mar. 2015.

[11] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations (ICLR), arXiv:1511.07289*, San Juan, Puerto Rico, Feb. 2016.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv:1512.00567*, Dec. 2015.

[13] K. Ullrich, J. Schlüter, and T. Grill, "Boundary Detection in Music Structure Analysis Using Convolutional Neural Networks," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, Oct. 2014.

[14] S. Fine, Y. Singer, and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, Jul. 1998.

[15] U. Hadar and H. Messer, "High-order Hidden Markov Models - Estimation and Implementation," in *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, Aug. 2009, pp. 249–252.

[16] R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia, "Chord Recognition Using Duration-Explicit Hidden Markov Models," in *13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, Oct. 2012.

[17] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[18] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases." in *3rd International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[19] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *Proceedings of the 8th International Workshop on Multidimensional Systems*, Erlangen, Germany, Sep. 2013.

[20] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis." in *12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, Oct. 2011.

[21] F. Korzeniowski, D. R. W. Sears, and G. Widmer, "A Large-Scale Study of Language Models for Chord Prediction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018.