IMPROVED CHORD RECOGNITION BY COMBINING DURATION AND HARMONIC LANGUAGE MODELS

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

ABSTRACT

Chord recognition systems typically comprise an acoustic model that predicts chords for each audio frame, and a temporal model that casts these predictions into labelled chord segments. However, temporal models have been shown to only smooth predictions, without being able to incorporate musical information about chord progressions. Recent research discovered that it might be the low hierarchical level such models have been applied to (directly on audio frames) which prevents learning musical relationships, even for expressive models such as recurrent neural networks (RNNs). However, if applied on the level of chord sequences, RNNs indeed can become powerful chord predictors. In this paper, we disentangle temporal models into a harmonic language model—to be applied on chord sequences—and a chord duration model that connects the chord-level predictions of the language model to the frame-level predictions of the acoustic model. In our experiments, we explore the impact of each model on the chord recognition score, and show that using harmonic language and duration models improves the results.

1. INTRODUCTION

Chord recognition methods recognise and transcribe musical chords from audio recordings. Chords are highly descriptive harmonic features that form the basis of many kinds of applications: theoretical, such as computational harmonic analysis of music; practical, such as automatic lead-sheet creation for musicians [1] or music tutoring systems [2]; and finally, as basis for higher-level tasks such as cover song identification or key classification. Chord recognition systems face the two key problems of extracting meaningful information from noisy audio, and casting this information into sensible output. These translate to *acoustic modelling* (how to predict a chord label for each position or frame in the audio), and *temporal modelling* (how to create meaningful segments of chords from these possibly volatile frame-wise predictions).

Acoustic models extract frame-wise chord predictions, typically in the form of a distribution over chord labels. Originally, these models were hand-crafted and split into feature extraction and pattern matching, where the former computed some form of pitch-class profiles (e.g. [26, 29, 33]), and the latter used template matching or Gaussian mixtures [6, 14] to model these features. Recently, however, neural networks became predominant for acoustic modelling [18, 22, 23, 27]. These models usually compute a distribution over chord labels directly from spectral representations and thus fuse both feature extraction and pattern matching. Due to the discriminative power of deep neural networks, these models achieve superior results.

Temporal models process the predictions of an acoustic model and cast them into coherent chord segments. Such models are either task-specific, such as hand-designed Bayesian networks [26], or general models learned from data. Here, it is common to use hidden Markov models [8] (HMMs), conditional random fields [23] (CRFs), or recurrent neural networks (RNNs) [2, 32]. However, existing models have shown only limited capabilities to improve chord recognition results. First-order models are not capable of learning meaningful musical relations, and only smooth the predictions [4, 7]. More powerful models, such as RNNs, do not perform better than their first-order counterparts [24]. In addition to the fundamental flaw of first-order models (chord patterns comprise more than two chords) both approaches are limited by the low hierarchical level they are applied on: the temporal model is required to predict the next symbol for each audio frame. This makes the model focus on short-term smoothing, and neglect longer-term musical relations between chords, because, most of the time, the chord in the next audio frame is the same as in the current one. However, exploiting these longer-term relations is crucial to improve the prediction of chords. RNNs, if applied on *chord sequences*, are capable of learning these relations, and become powerful chord predictors [21].

Our contributions in this paper are as follows: i) we describe a probabilistic model that allows for the integration of chord-level language models with frame-level acoustic models, by connecting the two using chord duration models; ii) we develop and apply chord language models and chord duration models based on RNNs within this framework;

---

[1] https://chordify.net/
[2] https://yousician.com

and iii) we explore how these models affect chord recognition results, and show that the proposed integrated model out-performs existing temporal models.
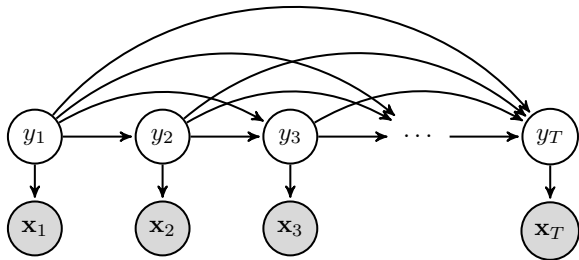
2. CHORD SEQUENCE MODELLING

Chord recognition is a sequence labelling task, i.e. we need to assign a categorical label $y_t \in \mathcal{Y}$ (a chord from a chord alphabet) to each member of the observed sequence $\mathbf{x}_t$ (an audio frame), such that y_t is the harmonic interpretation of the music represented by $\mathbf{x}_t$. Formally,

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

Assuming a generative structure as shown in Fig. 1, the probability distribution factorises as

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

where P_A is the acoustic model, P_T the temporal model, and $P(y_t)$ the label prior which we assume to be uniform as in [31].


Figure 1. Generative chord sequence model. Each chord label y_t depends on all previous labels $y_{1:t-1}$.

The temporal model P_T predicts the chord symbol of each audio frame. As discussed earlier, this prevents both finite-context models (such as HMMs or CRFs) and unrestricted models (such as RNNs) to learn meaningful harmonic relations. To enable this, we disentangle P_T into a *harmonic language model* P_L and a *duration model* P_D, where the former models the harmonic progression of a piece, and the latter models the duration of chords.

The language model P_L is defined as $P_L\left(\bar{y}_k \mid \bar{y}_{1:k-1}\right)$, where $\bar{y}_{1:k} = \mathcal{C}\left(y_{1:t}\right)$, and $\mathcal{C}\left(\cdot\right)$ is a sequence compression mapping that removes all consecutive duplicates of a chord (e.g. $\mathcal{C}\left((C, C, F, F, G)\right) = (C, F, G)$). The frame-wise labels $y_{1:t}$ are thus reduced to chord changes, and P_L can focus on modelling these.

The duration model P_D is defined as $P_D\left(s_t \mid y_{1:t-1}\right)$, where $s_t \in \{c, s\}$ indicates whether the chord changes (c) or stays the same (s) at time t. P_D thus only predicts whether the chord will change or not, but not which chord will follow—this is left to the language model P_L. This definition allows P_D to consider the preceding *chord labels* $y_{1:t-1}$; in practice, we restrict the model to only depend on
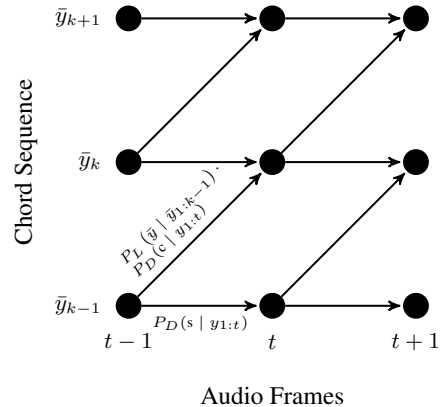
Figure 2. Chord-time lattice representing the temporal model P_T, split into a language model P_L and duration model P_D. Here, $\bar{y}_{1:K}$ represents a concrete chord sequence. For each audio frame, we move along the time-axis to the right. If the chord changes, we move diagonally to the upper right. This corresponds to the first case in Eq. 2. If the chord stays the same, we move only to the right. This corresponds to the second case of the equation.

the preceding chord *changes*, i.e. $P_D\left(s_t \mid s_{1:t-1}\right)$. Exploring more complex models of harmonic rhythm is left for future work.

Using these definitions, the temporal model P_T factorises as

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

The chord progression can then be interpreted as a path through a chord-time lattice as shown in Fig. 2.

This model cannot be decoded efficiently at test-time because each y_t depends on all predecessors. We will thus use either models that restrict these connections to a finite past (such as higher-order Markov models) or use approximate inference methods for other models (such as RNNs).

3. MODELS

The general model described above requires three sub-models: an acoustic model P_A that predicts a chord distribution from each audio frame, a duration model P_D that predicts when chords change, and a language model P_L that predicts the progression of chords in the piece.

3.1 Acoustic Model

The acoustic model we use is a VGG-style convolutional neural network, similar to the one presented in [23]. It uses three convolutional blocks: the first consists of 4 layers of 32 3×3 filters (with zero-padding in each layer), 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
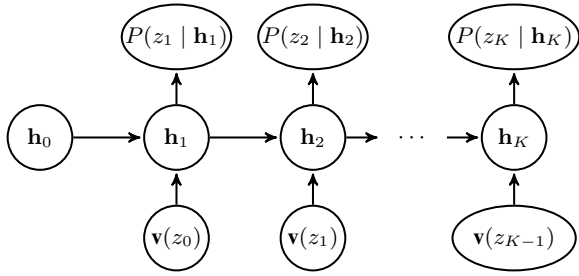
Figure 3. Sketch of a RNN used for next step prediction, where z_k refers to an arbitrary categorical input, $\mathbf{v}(\cdot)$ is a (learnable) input embedding vector, and $\mathbf{h}_k$ the hidden state at step k. Arrows denote matrix multiplications followed by a non-linear activation function. The input is padded with a dummy input z_0 in the beginning. The network then computes the probability distribution for the next symbol.

probability 0.2, and each layer is followed by batch normalisation [19] and an ELU activation function [10]. Finally, a linear convolution with 25 1×1 filters followed by global average pooling and a softmax produces the chord class probabilities $P_A(y_t \mid \mathbf{x}_t)$.

The input to the network is a 1.5 s patch of a quarter-tone spectrogram computed using a logarithmically spaced triangular filter bank. Concretely, we process the audio at a sample rate of 44 100 Hz using the STFT with a frame size of 8192 and a hop size of 4410. Then, we apply to the magnitude of the STFT a triangular filter bank with 24 filters per octave between 65 Hz and 2 100 Hz. Finally, we take the logarithm of the resulting magnitudes to compress the input range.

Neural networks tend to produce over-confident predictions, which in further consequence could over-rule the predictions of a temporal model [9]. To mitigate this, we use two techniques: first, we train the model using uniform smoothing (i.e. we assign a proportion of $1 - \beta$ to other classes during training); second, during inference, we apply the *temperature softmax* function $\sigma_\tau(\mathbf{z})_j = e^{z_j/\tau} / \sum_{k=1}^{K} e^{z_k/\tau}$ instead of the standard softmax in the final layer. Higher values of τ produce smoother probability distributions. In this paper, we use $\beta = 0.9$ and $\tau = 1.3$, as determined in preliminary experiments.

3.2 Language Model

The language model P_L predicts the next chord, regardless of its duration, given the chord sequence it has previously seen. As shown in [21], RNN-based models perform better than n-gram models at this task. We thus adopt this approach, and refer the reader to [21] for details.

To give an overview, we follow the set-up introduced by [28] and use a recurrent neural network for next-chord prediction. The network's task is to compute a probability distribution over all possible next chord symbols, given the chord symbols it has observed before. Figure 3 shows an RNN in a general next-step prediction task. In our case, the inputs z_k are the chord symbols given by $\mathcal{C}(y_{1:T})$.

We will describe in detail the network's hyper-parameters in Section 4, where we will also evaluate the

effect the language models have on chord recognition.

3.3 Duration Model

The duration model P_D predicts whether the chord will change in the next time step. This corresponds to modelling the duration of chords. Existing temporal models induce implicit duration models: for example, an HMM implies an exponential chord duration distribution (if one state is used to model a chord), or a negative binomial distribution (if multiple left-to-right states are used per chord). However, such duration models are simplistic, static, and do not adapt to the processed piece.

An explicit duration model has been explored in [4], where beat-synchronised chord durations were stored as discrete distributions. Their approach is useful for beat-synchronised models, but impractical for frame-wise models—the probability tables would become too large, and data too sparse to estimate them. Since our approach avoids the potentially error-prone beat synchronisation, the approach of [4] does not work in our case.

Instead, we opt to use recurrent neural networks to model chord durations. These models are able to adapt to characteristics of the processed data [21], and have shown great potential in processing periodic signals [1] (and chords do change periodically within a piece). To train an RNN-based duration model, we set up a next-step-prediction task, identical in principle to the set-up for harmonic language modelling: the network has to compute the probability of a chord change in the next time step, given the chord changes it has seen in the past. We thus simplify $P_D(s_t \mid y_{1:t-1}) \widehat{=} P_D(s_t \mid s_{1:t-1})$, as mentioned earlier. Again, see Fig. 3 for an overview (for duration modelling, replace z_k with s_t).

In Section 4, we will describe in detail the hyper-parameters of the networks we employed, and compare the properties of various settings to baseline duration models. We will also assess the impact on the duration modelling quality on the final chord recognition result.

3.4 Model Integration

Dynamic models such as RNNs have one main advantage over their static counter-parts (e.g. n-gram models for language modelling or HMMs for duration modelling): they consider all previous observations when predicting the next one. As a consequence, they are able to adapt to the piece that is currently processed—they assign higher probabilities to sub-sequences of chords they have seen earlier [21], or predict chord changes according to the harmonic rhythm of a song (see Sec. 4.3). The flip side of the coin is, however, that this property prohibits the use of dynamic programming approaches for efficient decoding. We cannot exactly *and* efficiently decode the best chord sequence given the input audio.

Hence we have to resort to approximate inference. In particular, we employ *hashed beam search* [32] to decode the chord sequence. General beam search restricts the search space by keeping only the N_b best solutions up to the current time step. (In our case, the N_b best paths through

all possible chord-time lattices, see Fig. 2.) However, as pointed out in [32], the beam might saturate with almost identical solutions, e.g. the same chord sequence differing only marginally in the times the chords change. Such pathological cases may impair the final estimate. To mitigate this problem, hashed beam search forces the tracked solutions to be diverse by pruning similar solutions with lower probability.

The similarity of solutions is determined by a task-specific *hash function*. For our purpose, we define the hash function of a solution to be the last N_h chord symbols in the sequence, regardless of their duration; formally, the hash function $f_h(y_{1:t}) = \bar{y}_{(k-N_h):k}$. (Recall that $\bar{y}_{1:k} = \mathcal{C}(y_{1:t})$.) In contrast to the hash function originally proposed in [32], which directly uses $y_{(t-N_h):t}$, our formulation ensures that sequences that differ only in timing, but not in chord sequence, are considered similar.

To summarise, we approximately decode the optimal chord transcription as defined in Eq. 1 using hashed beam search, which at each time step keeps the best N_b solutions, and at most N_s similar solutions.

4. EXPERIMENTS

In our experiments, we will first evaluate harmonic language and duration models individually. Here, we will compare the proposed models to common baselines. Then, we will integrate these models into the chord recognition framework we outlined in Section 2, and evaluate how the individual parts interact in terms of chord recognition score.

4.1 Data

We use the following datasets in 4-fold cross-validation. **Isophonics**[3] : 180 songs by the Beatles, 19 songs by Queen, and 18 songs by Zweieck, 10:21 hours of audio; **RWC Popular [15]:** 100 songs in the style of American and Japanese pop music, 6:46 hours of audio; **Robbie Williams [13]:** 65 songs by Robbie Williams, 4:30 of audio; and **McGill Billboard [3]:** 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.

Furthermore, we used the following data sets (with duplicate songs removed) as additional data for training the language and duration models: 173 songs from the **Rock** [11] corpus; a subset of 160 songs from **UsPop2002**[4] for which chord annotations are available[5] ; 291 songs from **Weimar Jazz**[6] , with chord annotations taken from lead sheets of Jazz standards; and **Jay Chou** [12], a small collection of 29 Chinese pop songs.

We focus on the major/minor chord vocabulary, and following [7], map all chords containing a minor third to minor, and all others to major. This leaves us with 25 classes: 12 root notes × {major, minor} and the 'no- chord' class.

	GRU-512	GRU-32	4-gram	2-gram
log-P	−1.293	−1.576	−1.887	−2.393

Table 1. Language model results: average log-probability of the correct next chord computed by each model.

4.2 Language Models

The performance of neural networks depends on a good choice of hyper-parameters, such as number of layers, number of units per layer, or unit type (e.g. vanilla RNN, gated recurrent unit (GRU) [5] or long short-term memory unit (LSTM) [17]). The findings in [21] provide a good starting point for choosing hyper-parameter settings that work well. However, we strive to find a simpler model to reduce the computational burden at test time. To this end, we perform a grid search in a restricted search space, using the validation score of the first fold. We search over the following settings: number of layers $\in \{1, 2, 3\}$, number of units $\in \{256, 512\}$, unit type $\in \{GRU, LSTM\}$, input embedding $\in \{one\text{-}hot, \mathbb{R}^8, \mathbb{R}^{16}, \mathbb{R}^{24}\}$, learning rate $\in \{0.001, 0.005\}$, and skip connections $\in \{on, off\}$. Other hyper-parameters were fixed for all trials: we train the networks for 100 epochs using stochastic gradient descent with mini-batches of size 4, employ the Adam update rule [20], and starting from epoch 50, linearly anneal the learning rate to 0.

To increase the diversity in the training data, we use two data augmentation techniques, applied each time we show a piece to the network. First, we randomly shift the key of the piece; the network can thus learn that harmonic relations are independent of the key, as in roman numeral analysis. Second, we select a sub-sequence of random length instead of the complete chord sequence; the network thus has to learn to cope with varying context sizes.

The best model turned out to be a single-layer network of 512 GRUs, with a learnable 16-dimensional input embedding and without skip connections, trained using a learning rate of 0.005[7] . We compare this model and a smaller, but otherwise identical RNN with 32 units, to two baselines: a 2-gram model, and a 4-gram model. Both can be used for chord recognition in a higher-order HMM [25]. We train the n-gram models using maximum likelihood estimation with Lidstone smoothing as described in [21], using the key-shift data augmentation technique (sub-sequence cropping is futile for finite context models). As evaluation measure, we use the average log-probability of predicting the correct next chord. Table 1 presents the test results. The GRU models predict chord sequences with much higher probability than the baselines.

When we look into the input embedding $\mathbf{v}(\cdot)$, which was learned by the RNN during training from a random initialisation, we observe an interesting positioning of the chord symbols (see Figure 4). We found that similar patterns develop for all 1-layer GRUs we tried, and these patterns are consistent for all folds we trained on. We observe i) that
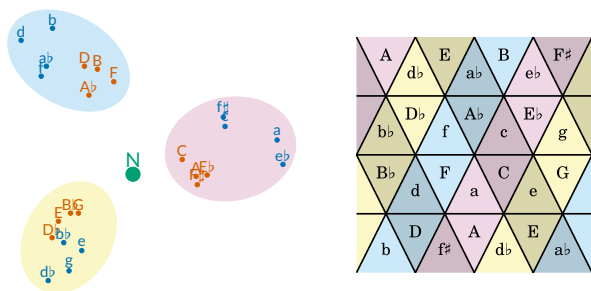
Figure 4. Chord embedding projected into 2D using PCA (left); Tonnetz of triads (right). The "no-chord" class resides in the center of the embedding. Major chords are upper-case and orange, minor chords lower-case and blue. Clusters in the projected embedding and the corresponding positions in the Tonnetz are marked in color. If projected into 3D (not shown here), the chord clusters split into a lower and upper half of four chords each. The chords in the lower halves are shaded in the Tonnetz representation.

chords form three clusters around the center, in which the minor chords are farther from the center than major chords; ii) that the clusters group major and minor chords with the same root, and the distance between the roots are minor thirds (e.g. C, E♭, F♯, A); iii) that clockwise movement in the circle of fifths corresponds to clockwise movement in the projected embedding; and iv) that the way chords are grouped in the embedding corresponds to how they are connected in the Tonnetz.

At this time, we cannot provide an explanation for these automatically emerging patterns. However, they warrant a further investigation to uncover why this specific arrangement seems to benefit the predictions of the model.

4.3 Duration Models

As for the language model, we performed a grid search on the first fold to find good choices for the recurrent unit type $\in$ {vanilla RNN, GRU, LSTM}, and number of recurrent units $\in$ {16, 32, 64, 128, 256} for the LSTM and GRU, and {128, 256, 512} for the vanilla RNN. We use only one recurrent layer for simplicity. We found networks of 256 GRU units to perform best; although this indicates that even bigger models might give better results, for the purposes of this study, we think that this configuration is a good balance between prediction quality and model complexity.

The models were trained for 100 epochs using the Adam update rule [20] with a learning rate linearly decreasing from 0.001 to 0. The data was processed in mini-batches of 10, where the sequences were cut in excerpts of 200 time steps (20 s). We also applied gradient clipping at a value of 0.001 to ensure a smooth learning progress.

We compare the best RNN-based duration model with two baselines. The baselines are selected because both are implicit consequences of using HMMs as temporal model, as it is common in chord recognition. We assume a single parametrisation for each chord; this ostensible simplification is justified, because simple temporal models such as HMMs do not profit from chord information, as shown
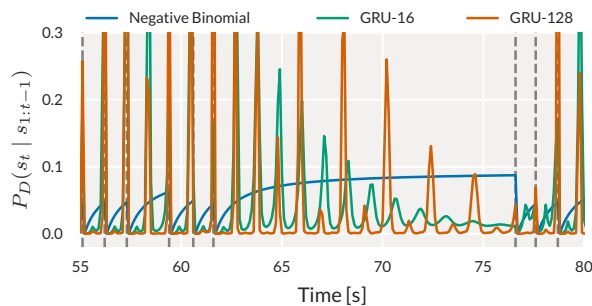
Figure 5. Probability of chord change computed by different models. Gray vertical dashed lines indicate true chord changes.

	GRU-256	GRU-16	Neg. Binom.	Exp.
log-P	−2.014	−2.868	−3.946	−4.003

Table 2. Duration model results: average log-probability of chord durations computed by each model.

by [4, 7]. The first baseline we consider is a *negative binomial distribution*. It can be modelled by a HMM using n states per chord, connected in a left-to-right manner, with transitions of probability p between the states (self-transitions thus have probability $1 - p$). The second, a special case of the first with $n = 1$, is an *exponential distribution*; this is the implicit duration distribution used by all chord recognition models that employ a simple 1-state-per-chord HMM as temporal model. Both baselines are trained using maximum likelihood estimation.

To measure the quality of a duration model, we consider the average log-probability it assigns to a chord duration. The results are shown in Table 2. We further added results for the simplest GRU model we tried—using only 16 recurrent units—to indicate the performance of small models of this type. We will also use this simple model when judging the effect of duration modelling on the final result in Sec. 4.4. As seen in the table, both GRU models clearly out-perform the baselines.

Figure 5 shows the reason why the GRU performs so much better than the baselines: as a dynamic model, it can adapt to the harmonic rhythm of a piece, while static models are not capable of doing so. We see that a GRU with 128 units predicts chord changes with high probability at periods of the harmonic rhythm. It also reliably remembers the period over large gaps in which the chord did not change (between seconds 61 and 76). During this time, the peaks decay differently for different multiples of the period, which indicates that the network simultaneously tracks multiple periods of varying importance. In contrast, the negative binomial distribution statically yields a higher chord change probability that rises with the number of audio frames since the last chord change. Finally, the smaller GRU model with only 16 units also manages to adapt to the harmonic rhythm; however, its predictions between the peaks are noisier, and it fails to remember the period correctly in the time without chord changes.

Model	Root	Maj/Min	Seg.
2-gram / neg. binom.	0.812	0.795	0.804
GRU-512 / GRU-256	**0.821**	**0.805**	**0.814**

Table 3. Results of the standard model (2-gram language model with negative binomial durations) compared to the best one (GRU language and duration models).

4.4 Integrated Models

The individual results for the language and duration models are encouraging, but only meaningful if they translate to better chord recognition scores. This section will thus evaluate if and how the duration and language models affect the performance of a chord recognition system.

The acoustic model used in these experiments was trained for 300 epochs (with 200 parameter updates per epoch) using a mini-batch size of 512 and the Adam update rule with standard parameters. We linearly decay the learning rate to 0 in the last 100 epochs.

We compare all combinations of language and duration models presented in the previous sections. For language modelling, these are the GRU-512, GRU-32, 4-gram, and 2-gram models; for duration modelling, these are the GRU-256, GRU-16, and negative binomial models. (We leave out the exponential model, because its results differ negligibly from the negative binomial one). The models are decoded using the Hashed Beam Search algorithm, as described in Sec. 3.4: we use a beam width of $N_b = 25$, where we track at most $N_s = 4$ similar solutions as defined by the hash function f_h, where the number of chords considered is set to $N_h = 5$. These values were determined by a small number of preliminary experiments.

Additionally, we evaluate exact decoding results for the n-gram language models in combination with the negative binomial duration distribution. This will indicate how much the results suffer due to the approximate beam search.

As main evaluation metric, we use the weighted chord symbol recall (WCSR) over the major/minor chord alphabet, as defined in [30]. We thus compute WCSR = t_c/t_a, where t_c is the total duration of chord segments that have been recognised correctly, and t_a is the total duration of chord segments annotated with chords from the target alphabet. We also report chord root accuracy and a measure of segmentation (see [16], Sec. 8.3). Table 3 compares the results of the standard model (the combination that implicitly emerges in simple HMM-based temporal models) to the best model found in this study. Although the improvements are modest, they are consistent, as shown by a paired t-test ($p < 2.487 \times 10^{-23}$ for all differences).

Figure 6 presents the effects of duration and language models on the WCSR. Better language and duration models directly improve chord recognition results, as the WCSR increases linearly with higher log-probability of each model. As this relationship does not seem to flatten out, further improvement of each model type can still increase the score. We also observe that the approximate beam search does not impair the result by much compared to exact decoding (compare the dotted blue line with the solid one).
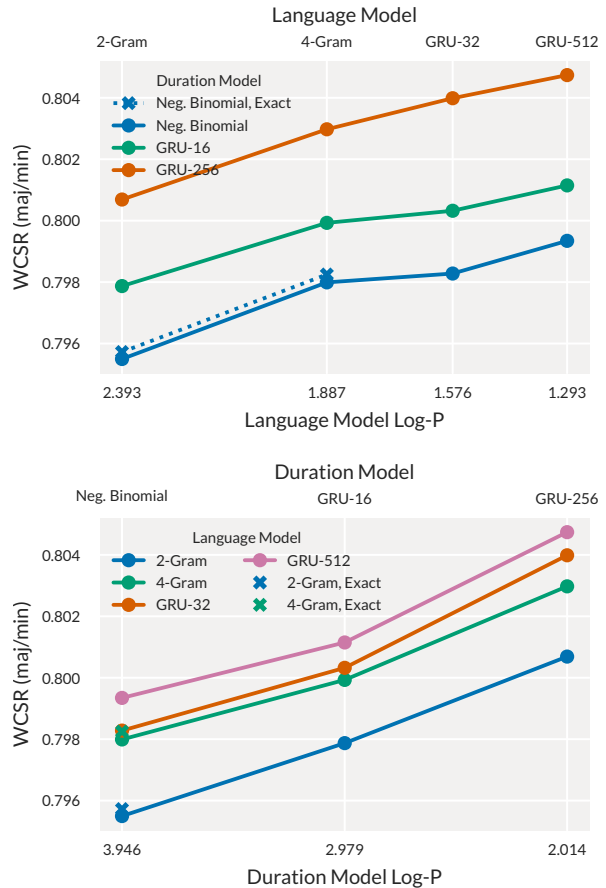
Figure 6. Effect of language and duration models on the final result. Both plots show the same results from different perspectives.

5. CONCLUSION AND DISCUSSION

We described a probabilistic model that disentangles three components of a chord recognition system: the acoustic model, the duration model, and the language model. We then developed better duration and language models than have been used for chord recognition, and illustrated why the RNN-based duration models perform better and are more meaningful than their static counterparts implicitly employed in HMMs. (For a similar investigation for chord language models, see [21].) Finally, we showed that improvements in each of these models directly influence chord recognition results.

We hope that our contribution facilitates further research in harmonic language and duration models for chord recognition. These aspects have been neglected because they did not show great potential for improving the final result [4, 7]. However, we believe (see [24] for some evidence) that this was due to the improper assumption that temporal models applied on the time-frame level can appropriately model musical knowledge. The results in this paper indicate that chord transitions modelled on the chord level, and connected to audio frames via strong duration models, indeed have the capability to improve chord recognition results.

7. REFERENCES

[1] Sebastian Böck and Markus Schedl. Enhanced Beat Tracking With Context-Aware Neural Networks. In *14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, September 2011.

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio Chord Recognition With Recurrent Neural Networks. In *14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, November 2013.

[3] John Ashley Burgoyne, Jonathan Wild, and Ichiro 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, October 2011.

[4] Ruofeng Chen, Weibin Shen, Ajay Srinivasamurthy, and Parag Chordia. Chord Recognition Using Duration-Explicit Hidden Markov Models. In *13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, October 2012.

[5] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), arXiv:1409.1259*, Doha, Qatar, October 2014.

[6] Taemin Cho. *Improved Techniques for Automatic Chord Recognition from Music Audio Signals*. Dissertation, New York University, New York, 2014.

[7] Taemin Cho and Juan P. Bello. On the Relative Importance of Individual Components of Chord Recognition Systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):477–492, February 2014.

[8] Taemin Cho, Ron J Weiss, and Juan Pablo Bello. Exploring Common Variations in State Of The Art Chord Recognition Systems. In *Proceedings of the Sound and Music Computing Conference (SMC)*, Barcelona, Spain, July 2010.

[9] Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *arXiv:1612.02695*, December 2016.

[10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp 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, February 2016.

[11] Trevor de Clercq and David Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(01):47–70, January 2011.

[12] Junqi Deng and Yu-Kwong Kwok. Automatic Chord estimation on seventhsbass Chord vocabulary using deep neural network. In *2016 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Shanghai, China, March 2016.

[13] Bruno Di Giorgi, Massimiliano Zanoni, Augusto Sarti, and Stefano 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, September 2013.

[14] Takuya Fujishima. Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music. In *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, October 1999.

[15] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC Music Database: Popular, Classical and Jazz Music Databases. In *3rd International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002.

[16] Christopher Harte. *Towards Automatic Extraction of Harmony Information from Music Signals*. Dissertation, Department of Electronic Engineering, Queen Mary, University of London, London, United Kingdom, 2010.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[18] Eric J. Humphrey and Juan P. Bello. Rethinking Automatic Chord Recognition with Convolutional Neural Networks. In *11th International Conference on Machine Learning and Applications (ICMLA)*, Boca Raton, USA, December 2012.

[19] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167*, March 2015.

[20] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR), arXiv:1412.6980*, San Diego, USA, May 2015.

[21] Filip Korzeniowski, David R. W. Sears, and Gerhard 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, April 2018.

[22] Filip Korzeniowski and Gerhard Widmer. Feature Learning for Chord Recognition: The Deep Chroma Extractor. In *17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, August 2016.

[23] Filip Korzeniowski and Gerhard 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, September 2016.

[24] Filip Korzeniowski and Gerhard 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, June 2017.

[25] Filip Korzeniowski and Gerhard Widmer. Automatic Chord Recognition with Higher-Order Harmonic Language Modelling. In *26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.

[26] M. Mauch and S. Dixon. Simultaneous Estimation of Chords and Musical Context From Audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1280–1289, August 2010.

[27] Brian McFee and Juan Pablo Bello. Structured Training for Large-Vocabulary Chord Recognition. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, October 2017.

[28] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048, Chiba, Japan, September 2010.

[29] Meinard Müller, Sebastian Ewert, and Sebastian Kreuzer. Making chroma features more robust to timbre changes. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009.

[30] Johan Pauwels and Geoffroy Peeters. Evaluating automatically estimated chord sequences. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013.

[31] 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*, 2(1):161–174, January 1994.

[32] Siddharth Sigtia, Nicolas Boulanger-Lewandowski, and Simon Dixon. Audio Chord Recognition With A Hybrid Recurrent Neural Network. In *16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, October 2015.

[33] Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based Approach for Automatic Chord Detection Using Refined Acoustic Features. In *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, USA, March 2010.