# USE OF HIDDEN MARKOV MODELS AND FACTORED LANGUAGE MODELS FOR AUTOMATIC CHORD RECOGNITION

**Maksim Khadkevich**
FBK-irst, Universitá degli studi di Trento,
Via Sommarive, 14 - Povo - 38050, Trento, Italy
khadkevich@fbk.eu

**Maurizio Omologo**
Fondazione Bruno Kessler-irst
Via Sommarive, 18 - Povo - 38050 Trento, Italy
omologo@fbk.eu

## ABSTRACT

This paper focuses on automatic extraction of acoustic chord sequences from a musical piece. Standard and factored language models are analyzed in terms of applicability to the chord recognition task. Pitch class profile vectors that represent harmonic information are extracted from the given audio signal. The resulting chord sequence is obtained by running a Viterbi decoder on trained hidden Markov models and subsequent lattice rescoring, applying the language model weight. We performed several experiments using the proposed technique. Results obtained on 175 manually-labeled songs provided an increase in accuracy of about 2%.

## 1. INTRODUCTION

Among all existing musical styles, western tonal music, which is one of the most popular nowadays, is known for its strong relationship to harmony. Harmonic structure can be used for the purposes of content-based indexing and retrieval since it is correlated to the mood, style and genre of musical composition. Automatic analysis of digital music signals has attracted the attention of many researchers, establishing and evolving the Music Information Retrieval (MIR) community. One of the largest research areas of the interdisciplinary science of MIR is music transcription. A subtask of this problem, which deals with the extraction of harmonic properties of audio signal, is chord recognition. Basically, harmony denotes a combination of simultaneously or progressively sounding notes, forming chords and their progressions. In almost all cases the harmonic structure of a piece of music can be converted into a chord sequence. A great interest in chords can be indicated by a number of websites containing chord databases for existing popular songs. Automatic extraction of harmonic structure can also be of great use to musicologists, who perform harmonic analysis over large collections of audio data.

As in the case of speech recognition, one of the most critical issues in chord recognition is the choice of the acoustic feature set to use in order to represent the waveform in a compact way. One of the most successfully used feature set is chromagram, which can be represented as a sequence of chroma vectors. Each chroma vector, also called Pitch Class Profile (PCP), describes the harmonic content of a given frame. The amount of energy for each pitch class is described by one component in the PCP vector. Since a chord consists of a number of tones and can be uniquely determined by their positions, chroma vectors can be used effectively for chord representation. The chroma feature was firstly introduced for music computing tasks by Fujishima [1]. He proposed a real-time chord recognition system, describing extraction of 12-dimensional chroma vectors from the Discrete Fourier Transform (DFT) of the audio signal and introducing a numerical pattern matching method using built-in chord-type templates to determine the most likely root and chord type. The statistical learning method for chord recognition was suggested by Sheh and Ellis [2]. They exploited the Expectation-Maximization (EM) algorithm to train hidden Markov models, while chords were treated as hidden states. Statistical information about chord progressions in their approach is represented by the state transitions in HMM. The approach of Papadopoulos and Peeters [3] incorporates simultaneous estimation of chord progression and downbeats from an audio file. They paid a lot of attention to possible interaction of the metrical structure and the harmonic information of a piece of music.

Incorporating statistical information on chord progressions into a chord recognition system is an important issue. It has been addressed in several works through different techniques. Mauch and Dixon [4] used one of the simplest forms of $N$-grams – the bigram language model. In the approaches of Papadopoulos and Peeters, Lee and Slaney [3, 5] chord sequence modeling is introduced through state transition probabilities in HMM. In their case "language model" is a part of HMM and is derived from the Markov assumption, where chord probability is defined by only one predecessor. Yoshioka et al. [6] presented an automatic chord transcription system which is based on generating hypotheses about tuples of chord symbols and chord boundaries, and further evaluating the hypotheses, taking into account three criteria: acoustic features, chord progression patterns and bass sounds. This approach was further developed by Sumi et al. [7]. They mainly focused on the interrelationship among musical elements and made an

attempt to efficiently integrate information about bass lines into chord recognition framework. They used two 2-gram models, one for major keys and one for minor keys, which are obtained in advance from real music. A large study on the modeling of chord sequences by probabilistic N-grams was performed by Scholz et al. [8]. Unal et al. [9] used perplexity-based scoring to test the likelihoods of possible transcription sequences.

This paper investigates the applicability of standard and factored language models of high orders (3-gram, 4-gram). Experiments with different back-off strategies for factored language models are carried out.

The rest of the paper is organized as follows: section 2 describes the front-end processing. In section 3 the here adopted HMM-based classification engine is briefly outlined. Language modeling is presented in section 4. Section 5 is devoted to the description of the whole proposed chord recognition system. The experimental results and conclusion are then given in section 6 and section 7, respectively.

## 2. FRONT-END PROCESSING

Before extracting features, the tuning procedure described in [10] is applied in order to find the mis-tuning rate and set the reference frequency $f_{ref}$ for the "A4" tone. The necessity of tuning appears when audio was recorded from instruments that were not properly tuned in terms of semitone scale.

The feature extraction process starts with downsampling the signal to 11025 Hz and converting it to the frequency domain by a DFT applying Hamming window of 185.7 ms with 50% overlapping. The harmonic content is extracted from the frequency range between 100 Hz and 2 kHz only. The main reason for this is the fact that in this range the energy of the harmonic frequencies is stronger than non-harmonic frequencies of the semitones. A sequence of conventional 12-dimensional Pitch Class Profile (PCP) vectors, known as chromagram is used as acoustic feature set. Each element of PCP vector corresponds to the energy of one of the 12 pitch classes. The process of PCP extraction can be decomposed into several steps. After applying DFT, the energy spectrum is mapped to the chroma domain, as shown in (1).

$$n(f_k) = 12\log_2\left(\frac{f_k}{f_{ref}}\right) + 69, n \in \Re^+ \qquad (1)$$

where $f_{ref}$ denotes the reference frequency of "A4" tone, while $f_k$ and $n$ are the frequencies of Fourier transform and the semitone bin scale index, respectively. To reduce transients and noise we apply smoothing over time using median filtering, similarly to Peeters [11] and Mauch et al. [4]. At the last stage semitone bins are mapped to pitch classes, which results in the sequence of 12-dimensional PCP vectors:

$$c(n) = \mathrm{mod}(n, 12) \qquad (2)$$

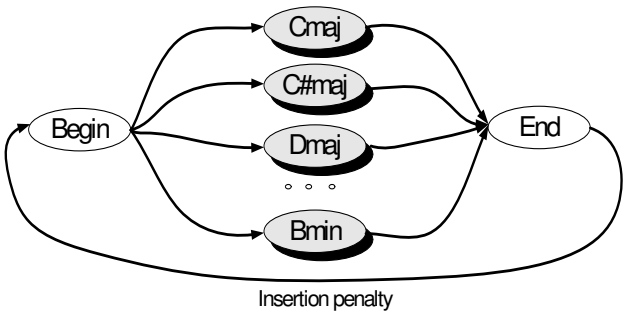


**Figure 1**. Connection scheme of trained models for decoding.

## 3. HIDDEN MARKOV MODELS

Hidden Markov models, which have been successfully used for modeling temporal sequences, are utilized in the proposed approach.

In contrast to many existing approaches [2, 3, 5], where chord is represented as a hidden state in one ergodic HMM, a separate left-to-right model is here created for each chord. In the given system configuration each model consists of 3 hidden states. The entry and exit states of a HMM are non-emitting, while the observation probabilities are identical for all emitting states. Observation vector probabilities in the emitting states can be approximated by a number of Gaussians in 12 dimensions, described by a mean vector and a covariance matrix. The feature vector components are assumed to be uncorrelated with one another, so the covariance matrix has a diagonal form. For each observation we use a mixture of 512 12-dimensional Gaussians. Songs from the training set are segmented according to the ground-truth labels so that each segment represents one chord. Chromagrams extracted from these segments are used for training, which is based on the application of the Baum-Welch algorithm.

Before running the recognition task, we extract a chromagram for each song from the test data. There is no preliminary segmentation as done on the training data for which a chroma vector sequence is extracted for each chord segment; only one chromagram is obtained for the whole test song. The trained chord HMMs are connected as shown in figure 1. Such parameter as insertion penalty is introduced, which allows for obtaining labels with different degrees of fragmentation. The Viterbi algorithm is then applied to the test data by using the resulting connected trained model in order to estimate the most likely chord sequence for each song and to produce a chord lattice.

## 4. LANGUAGE MODELING

A lot of different statistical language models have been proposed over years. The most successful among them appeared to be finite state transducers. In Natural Language processing N-grams are used for word prediction. Given $N-1$ predecessors, it can provide the probability of $N$-th element appearing. Language models have a variety of applications such as automatic speech recognition

and statistical machine translation. The main goal of language modeling can be explained as follows: having a sentence, which consists of $K$ words $(w_1, w_2, ...w_K)$, generate a probability model $p(w_1, w_2, ...w_K)$. In most common cases it can be expressed as (3).

$$p(w_1, w_2...w_K) = \prod_t p(w_t|w_1, w_2...w_{t-1}) = \prod_t p(w_t|h_t) \quad (3)$$

where $h_t$ is the history sufficient for determining the probability of $w_t$ word. In standard $N$-gram models the history consists of the immediately adjacent $N-1$ words. For example, in 3-gram model the probability of current word can be expressed as: $p(w_t|w_{t-1}, w_{t-2})$.

While estimating language model parameters, there exists the problem of sparse data. It is caused by the impossibility of producing maximum likelihood estimate of the model, because all combinations of $N$-word sequences are unlikely to be found in the training corpus. Since any training corpus is limited, some acceptable sequences can be missing from it, which leads to setting zero probability to plenty of $N$-grams. In order to cope with the problem, different techniques, such as back-off, smoothing and interpolation are used [12–14]. The main principle of back-off is to rely on lower-order model (e.g $p(w_t|w_{t-1})$) if there is zero evidence for higher-order (e.g. $p(w_t|w_{t-1}, w_{t-2})$) model. The order of dropping variables is known as back-off order. In the case of standard language models it is obvious that information taken from older predecessor will be less beneficial and it should be dropped prior to other predecessors.

In the proposed approach we draw direct analogy between a sentence in speech and a tune in a piece of music. The above-described strategy can be successfully used in chord sequences modeling. In this case a chord is the equivalent of a word and the sequence of chords can be modeled by means of the same technique.

### 4.1 Factored language models

Western music is known to be highly structural in terms of rhythm and harmony. In order to take advantage of mutual dependency between these two phenomena, we have studied the interrelationship between beat structure and chord durations. The number of occurrences as a function of chord duration in beats histogram is shown in figure 2. It is clearly seen that a greater part of chord durations is correlated to the metrical structure (2, 4, 8, 12, 16, 24, 32 beats), which suggests that including also chord durations in the language model is more convenient than analyzing just a sequence of chord symbols. This can be easily done with the help of factored language models (FLMs), which treat a word (chord) as a set of factors. FLMs have been recently proposed by Bilmes and Kirchoff [15] and showed promising results in modeling highly inflected languages, such as Arabic [16].

In a factored language model, a word (chord) can be represented as a bundle of factors: $w_t = \{f_t^1, f_t^2, ..., f_t^K\}$. The probability for FLM is given in (4), where $\pi(f_t^k)$ is

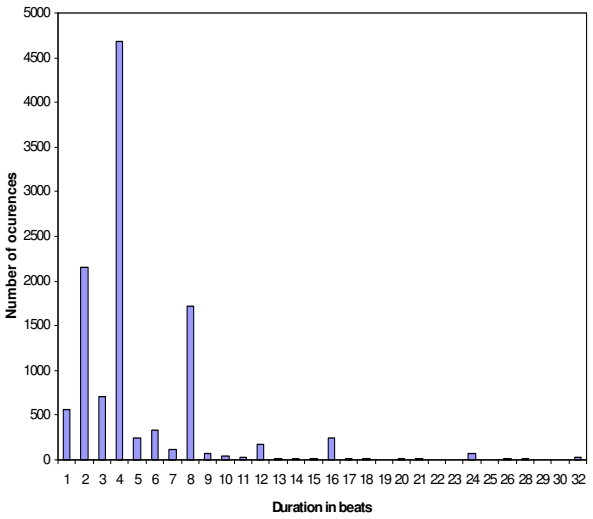


**Figure 2**. Chord Duration Histogram.

a set of variables (parents), which influence the probability of $f_t^k$. In our case to model chord sequences we use two factors: chord label $C_t$ and chord duration $D_t$: $w_t = \{C_t, D_t\}$.

$$p(w_t|h_t) = \prod_k p(f_t^k|\pi(f_t^k)) \quad (4)$$

As opposed to standard language models, where older predecessors give less relevant information at the given time instant, in FLMs there is no obvious order to drop parents $\pi(f_t^k)$. There are a lot of possibilities to choose less informative factors to drop among the others. Moreover, keeping some factors of older predecessors can be of greater benefit than keeping the value of some other factors, which are more relevant to the given time instant. One of the possible solutions is to use "generalized parallel back-off", which was initially proposed and well described by Bilmes and Kirchoff [15]. The main idea is to back-off factors simultaneously. The given set of back-off paths is determined dynamically based on the current values of the variables. (For a more detailed description, see [15]).

At the experimental stage we explore the standard back-off (a) and the parallel back-off (b) techniques, whose graphs are presented in figure 3. In both cases the chronological order is kept, while in the standard back-off case a higher priority to the factor of chord symbol is assigned. The arrows are marked with the factor being dropped at the current back-off step; blocks include the variables that influence the probability of chord label being estimated.

## 5. CHORD RECOGNITION SYSTEM

The full scheme of chord recognition system is depicted in figure 4.

Feature extraction part has been described in section 2. The beat extraction algorithm used here is introduced by Dixon [17] and is exploited as a separate module, called

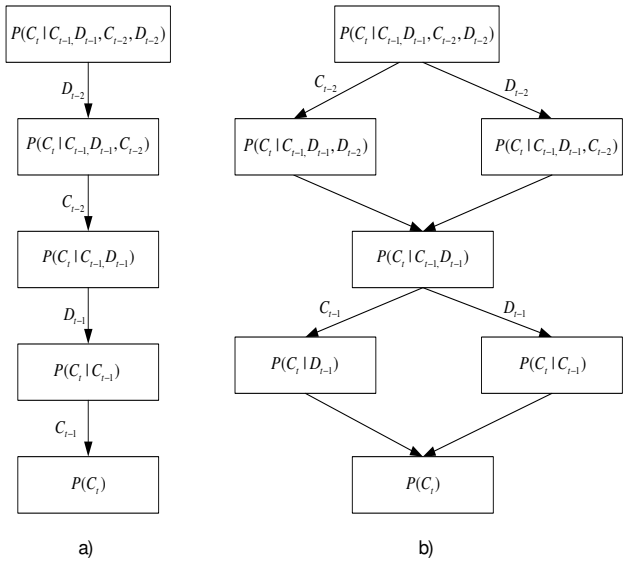


**Figure 3**. Standard back-off (a) and parallel back-off (b) graphs for tri-gram LM.

*BeatRoot* [1] .

The key detection module utilizes the approach suggested by Peeters [11], where trained HMMs are used to find the best score from 24 possible keys for the given sequence of chroma vectors for each test song. In the suggested system the key is assumed to be constant.

On the training stage, features extracted from waveforms are used to train hidden Markov models, while chord labels from training corpus are used as an input for language model parameter estimation. Language model training includes training either standard LMs or FLMs. For training standard LMs chord sequences taken from the training labels are used as input. For building text for FLM the information combined from beat extraction module and the training labels is used. For each chord symbol from ground-truth labels we estimate the duration in beats and produce an output in the form: "C-(chord type):D-(duration)". To minimize the problem of sparse data, all duration values are quantized by a relatively-small set of or integer values. Our codebook consists of the following values: 1, 2, 3, 4, 6, 8, 12, 16, 24 and 32 beats. The suggested codebook is supposed to be well-suited for the pop songs. This assumption is made on the basis of metrical analysis of the Beatles data (see fig. 2). The suggested scheme however might not be sufficient while modeling jazz or other genres.

In order to make our system key invariant, a key transformation technique is proposed here. In fact, the training corpus might not contain some type of chords and chord transitions due to the fact that keys with a lot of accidentals are much less widespread (G# maj, Ab min). Moreover, while estimating chord transition probabilities the relative change in the context of the given key (e.g. tonic – dominant – subdominant) is more relevant than exact chord names. For training data we have ground-truth table of keys for each song, while for test data we estimate key in the key detection module. Then, similar to training HMMs, by applying circular permutation, features and labels are converted to the Cmaj (in case of major key) or to Amin (in case of minor key). After the decoding procedure in order to produce final labels (in the original key of the analyzed song) obtained labels are converted back using the same scheme.

Similar to the approach of multiple-pass decoding, which has been successfully used in speech recognition [14], the decoding procedure consists of two steps. During the first step time-and-space efficient bigram language model is applied on the stage of Viterbi decoding, producing a lattice. A lattice can be represented by a directed graph, where nodes denote time instants and arcs are different hypotheses. Since lattices contain the information on the time boundaries, it is possible to make an estimation of duration in beats for each hypothesis. During the second step the obtained lattice is rescored applying more sophisticated language models (trigram and higher) on the reduced search space. Since the main problem is to extract chord labels, it is not necessary to model chord duration probabilities explicitly. Our decoding scheme, applying language modeling, is based on Viterbi decoding and subsequent lattice rescoring, where lattices contain the information on possible chord boundaries. Chord durations are used only to define chord label probabilities and the resulting chord boundaries are obtained from the lattices. Generally, standard LMs do not take into account duration factor at all, the only important thing here is just a sequence of labels. The advantage of FLM is that when applying the language model weight on the stage of lattice rescoring, chord durations contribute to the probabilities of different hypotheses in the lattice.

Standard LMs are manipulated using HTK [2] tools, while FLMs are managed using SRILM [18] toolkit, since HTK does not support this type of language models.

## 6. EXPERIMENTS

Evaluation of the proposed system was performed on the songs taken from 12 Beatles albums, ground-truth annotations for which were kindly provided by C. A. Harte [19]. The system can distinguish 24 different chord types (major and minor for each of 12 roots). 7th, min7, maj7, minmaj7, min6, maj6, 9, maj9, min9 chords are merged to their root triads; suspended augmented and diminished chords are discarded from the evaluation task. The percentage of duration of discarded chords results to be 2.71% of the whole material. In order to prevent the lack of training data (some chord types can appear only few times in the training corpus) only two models are trained: C-major and C-minor. For this purpose, all chroma vectors obtained from labeled segments are mapped to the C-root using circular permutation. After that mean vectors and covariance matrices are estimated for the two models. All the other models can be obtained by a circular permutation procedure.

[1] http://www.elec.qmul.ac.uk/people/simond/beatroot/index.html

[2] http://htk.eng.cam.ac.uk/

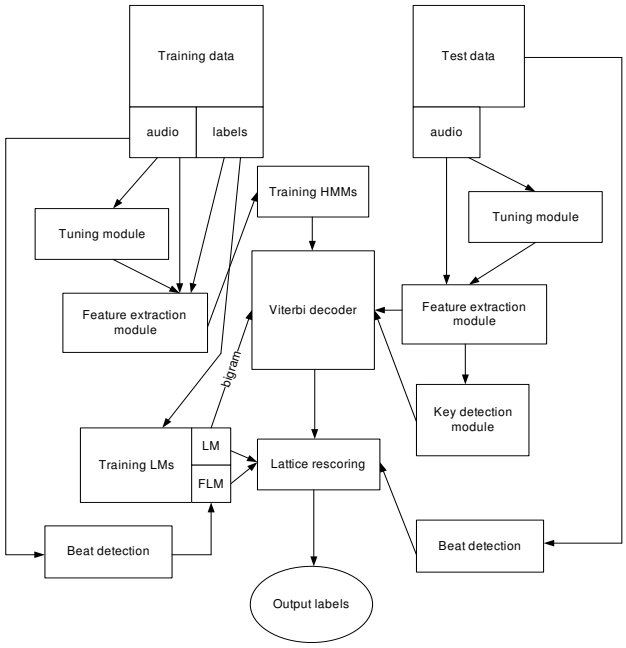


**Figure 4**. Chord recognition system.

For evaluation, the recognition rate measure was used, which in the given case corresponds to the total duration of correctly classified chords divided by the total duration of chords, as reported in the following:

$$rec.rate = \frac{|recognized\_chords| \cap |ground - truth\_chords|}{|ground - truth\_chords|} \quad (5)$$

The evaluation was performed frame by frame, as it was done under the MIREX [3] competition. In our experiments 3-gram and 4-gram language models were used. While working with FLMs, we exploited standard and generalized parallel back-off strategies (see figure 3; 4-gram graphs have the same structure and can be obtained from 3-gram graphs by adding one level).

It is worth mentioning that applying different language model weights on the stage of lattice rescoring one can obtain different recognition rates. Figure 5 indicates how recognition rate depends on the LM weight. In this case the curves correspond to the LM- and FLM-based systems; experiments were conducted on the fold 1 with 4-gram configuration.

In order to estimate the increase in performance introduced by including LM block and in order to compare efficiency of standard and factored language models, a 5-fold cross-validation was accomplished on the given data set. The folds were built in a random way and there is high album overlap. The recognition rates are shown in Table 1. Here "bl" is baseline system, "3lm" "3flm" "3flmgpb" are trigram configurations with key transformation for standard LM, FLM, and FLM with generalized parallel back-off respectively, "4lm" "4flm" "4flmgpb" are 4-gram configurations. For any of the given configurations, an average standard deviation of about 15% was also observed,

[3] http://www.music-ir.org/mirex/2008/index.php/Main_Page

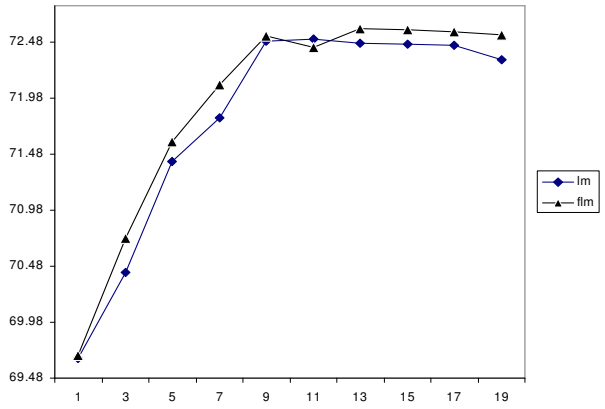


**Figure 5**. Recognition rate as a function of LM weight.

which was derived from the recognition rates computed on a song-by-song basis.

Experimental results showed that introducing language modeling increases the performance of the system, while generalized parallel back-off strategy for FLM did not show any advantages over standard back-off for the chord recognition task. Meanwhile, using FLM show very slight improvement (0.25 %) in comparison to standard LM. The differences in the output labels for LMs and FLMs are mainly on the junctions of chords. While using standard LM one can get a slight boundary deviation from its ground-truth value (e.g. 1 beat), using FLM fixes this in most cases because it takes into account the duration factor. That is why the difference in recognition rates is so small.

## 7. CONCLUSION

In this paper a set of experiments on chord recognition task including language modeling functionality as a separate layer has been conducted. The experimental results in a 5-fold cross-validation were conducted on a commonly used database of the songs by the Beatles. Factored language models were compared with standard language models and showed small increase in performance for the task. The main advantage of FLMs is that they possess a better chord recognition ability on the chord junctions. Comparing back-off techniques, we can assume that using generalized parallel back-off for the chord recognition task does not result in better performance.

However, the suggested system has a number of limitations: assuming the key of the song constant, one can not cope with key changes. A deeper study on different model smoothing and selection techniques as those addressed by Scholz et al. [8] could be reprised.

In general, experimental results showed that utilizing language models leads to an increase in accuracy by about 2%. This relatively small difference in performance may be due to the size of vocabulary for the chord recognition task in comparison with that of many speech recognition applications. The performance of chord recognition sys-

| data | bl | 3lm | 3flm | 3flmgpb | 4lm | 4flm | 4flmgpb |
|---|---|---|---|---|---|---|---|
| fold 1 | 70.81 | 72.22 | 72.55 | 72.56 | 72.39 | 72.53 | 72.27 |
| fold 2 | 70.23 | 70.78 | 71.15 | 71.51 | 71.09 | 71.38 | 71.25 |
| fold 3 | 65.87 | 66.81 | 66.59 | 67.01 | 67.22 | 66.89 | 67.17 |
| fold 4 | 66.20 | 67.15 | 67.60 | 67.61 | 67.64 | 67.62 | 67.51 |
| fold 5 | 66.19 | 69.73 | 69.72 | 68.55 | 68.55 | 69.72 | 69.77 |
| average | 67.86 | 69.34 | 69.52 | 69.45 | 69.38 | 69.63 | 69.59 |

**Table 1**. Evaluation results: recognition rates.

tems is perhaps influenced primarily by relevance and accuracy of the extracted features and related acoustic modeling.

## 8. REFERENCES


[1] Takuya Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proceedings of the International Computer Music Conference*, Beijing, 1999.

[2] A. Sheh and D. P. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proc. 4th International Conference on Music Information Retrieval*, 2003.

[3] H. Papadopoulos and G. Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Proc. ICASSP*, 2008.

[4] Matthias Mauch and Simon Dixon. A discrete mixture model for chord labelling. In *Proceedings of the 2008 ISMIR Conference*, Philadelphia, 2008.

[5] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2), february 2008.

[6] T. Yoshioka, T.Kitahara, K. Komatani, T. Ogata, and H.G. Okuno. Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, Barcelona, 2004.

[7] K. Sumi, K. Itoyama, K. Yoshii, K. Komatani, T. Ogata, and H. G. Okuno. Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation. In *Proceedings of the 2008 ISMIR Conference*, Philadelphia, 2008.

[8] R. Scholz, E. Vincent, and F. Bimbot. Robust modeling of musical chord sequences using probabilistic n-grams. In *Proc. ICASSP*, 2009.

[9] E. Unal, P. Georgiou, S. Narayanan, and E. Chew. Statistical modeling and retrieval of polyphonic music. In *Proc. IEEE MMSP*, 2007.

[10] M. Khadkevich and M. Omologo. Phase-change based tuning for automatic chord recognition. In *Proceedings of DAFX*, Como, Italy, 2009.

[11] G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the 2006 ISMIR Conference*, Victoria, Canada, 2006.

[12] J. Goodman. A bit of progress in language modeling. In *Computer, Speech and Language*, 2001.

[13] F. Jelinek. Statistical methods for speech recognition. In *MIT Press*, 1997.

[14] D. Jurafsky and J. H. Martin, editors. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2000.

[15] J. Bilmes and K. Kirchoff. Factored language models and generalized parallel backoff. In *HLT-NAACL*, 2003.

[16] K. Kirchhoff, D. Vergyri, K. Duh, J. Bilmes, and A. Stolcke. Morphology-based language modeling for arabic speech recognition. In *Computer, Speech and Language*, 2006.

[17] S. Dixon. Onset detection revisited. In *Proceedings of DAFX*, McGill, Montreal, Canada, 2006.

[18] A. Stolcke. Srilm. an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, 2002.

[19] C. Harte and M. Sandler. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 2005 ISMIR Conference*, 2005.