MySong: Automatic Accompaniment Generation for Vocal Melodies

Ian Simon
University of Washington
Seattle, WA
iansimon@cs.washington.edu

Dan Morris
Microsoft Research
Redmond, WA
dan@microsoft.com

Sumit Basu
Microsoft Research
Redmond, WA
sumitb@microsoft.com

ABSTRACT

We introduce MySong, a system that automatically chooses chords to accompany a vocal melody. A user with no musical experience can create a song with instrumental accompaniment just by singing into a microphone, and can experiment with different styles and chord patterns using interactions designed to be intuitive to non-musicians.

We describe the implementation of MySong, which trains a Hidden Markov Model using a music database and uses that model to select chords for new melodies. Model parameters are intuitively exposed to the user. We present results from a study demonstrating that chords assigned to melodies using MySong and chords assigned manually by musicians receive similar subjective ratings. We then present results from a second study showing that thirteen users with no background in music theory are able to rapidly create musical accompaniments using MySong, and that these accompaniments are rated positively by evaluators.

Author Keywords

Music, Hidden Markov Models

ACM Classification Keywords

H5.5. Sound and Music Computing: Methodologies and techniques, Modeling, Signal analysis, Systems.

INTRODUCTION

A songwriter often begins with an idea for a melody, and develops chords and accompaniment patterns to turn that melody into a song. This process is an art traditionally reserved for musicians with knowledge of musical structure and harmony. Musicians often use instruments to experiment with melodies and chords or to find chords to accompany a melody. On the other hand, individuals without knowledge of chords and harmony are generally unable to develop or experiment with musical ideas.

And although songwriting is a craft typically restricted to experienced musicians, a much larger set of people enjoy music and recreational singing. Particularly in light of the current trend toward creation and sharing of audio and video media online, this larger group might be inclined to write music – in fact might tremendously enjoy writing music – if it didn't require years of instrumental and theoretical training and practice. The goal of this work is to enable a creative but musically-untrained individual to get a taste of songwriting and music creation.

In this paper, we introduce MySong, a system that automatically chooses chords to accompany a vocal melody. A user with no experience in music can create a song just by singing into a microphone, and can experiment with different styles and chord patterns without any knowledge of music, using interactions designed to be intuitive to non-musicians.

We present the results of a study in which 30 musicians evaluate accompaniments created with MySong. These results show that scores assigned to these accompaniments were nearly identical to scores assigned to accompaniments created manually by experienced musicians.

We also present the results of a second study showing that thirteen users with no background in songwriting or harmony were able to create music in less than ten minutes using our system, and that this output is subjectively acceptable both to these users and to trained evaluators.

The contributions of this paper are two-fold:

1) We present MySong, a machine-learning-based system for generating appropriate chords to accompany a vocal melody. The parameters that drive this system are designed to be intuitive to non-musicians.
2) We present the results of two studies that validate the effectiveness of this system both in generating subjectively-acceptable accompaniments and in enabling non-musicians to rapidly and enjoyably create accompaniments of their own.

Musical Terminology

Throughout this paper, we attempt to minimize musical terminology, but will need to refer to several musical entities to describe MySong. To ensure that readers of varying musical backgrounds can follow our work, we introduce the requisite terminology here.

We use the term "melody" to refer to the sequence of pitches performed by a vocalist. We use the term "chord

CHI 2008, April 5–10, 2008, Florence, Italy.
Copyright 2008 ACM 978-1-60558-011-1/08/04…$5.00.

sequence" to refer to the series of chords – combinations of musical notes – that are performed to support a vocal melody. For popular music, a melody and an associated chord sequence comprise the primary representation used in music publishing: "lead sheets", which are used by instrumentalists or bands to represent music during performances. The chords in a chord sequence do not overlap in time. We use the term "accompaniment" to refer to the audio realization of a chord sequence; this might refer, for example, to the actual part played by a pianist who is supporting a vocalist. In this paper, "accompaniment" will generally refer to a synthetic piano playing the chords in a chord sequence.

Points of specific interest to musically-trained readers that depend on additional musical terminology will be deferred to footnotes wherever possible.

SUPPLEMENTARY MATERIAL

Because the output of this system is both subjective and auditory, we encourage readers to explore our supplementary materials. We refer the reader to the following Web page, which includes a number of the melodies and accompaniments created for or during our experiments, along with a video figure:

http://research.microsoft.com/~dan/mysong

RELATED WORK

To our knowledge, no system exists for generating accompaniments directly from a vocal melody. However, there has been a variety of prior work in tools and algorithms to help people make music, ranging from new forms of instruments to other systems that generate chords.

Music Creation Tools for Novices

A common approach to allowing novices to create music has been to create devices and interfaces that are pre-programmed with rules/constraints of musical structure, where musical variations are accessible through a set of controls that constrain outputs to musically sensible results. BeatBugs [19] are handheld devices on which each of several users can tap out a rhythm. Two continuous controllers allow users to vary a sound in terms of pitch and degree of ornamentation. The Hyperscore project [6] allows users to construct musical "motifs" out of notes; the volume and pitch of these motifs can then be manipulated using a graphical sketchpad. A more abstract version of this control is used in [11], where a variety of sensors (EEG, mouse motions, etc.) are used to control high-level parameters that drive a music generation module.

While the nature of the inputs and outputs in our project are quite different from these, we build on these authors' notion of providing a small number of controls with which a user can modify musical content in musically sensible ways.

Automated Accompaniment via Score-Following

There has been significant work in a related area also referred to as "automated accompaniment"; this work addresses playing back a pre-existing accompaniment that must be appropriately sped up or slowed down to match a soloist. As such, this is quite different from our work, which *generates* accompaniment patterns; however, this work in some cases uses similar techniques and models. Grub and Dannenberg [9] use a probabilistic approach to track a vocalist. Raphael [17] models both discrete score position and continuous tempo for a solo acoustic instrument via a hybrid discrete/continuous graphical model. A variety of other authors have contributed to this area with techniques based on Hidden Markov Models (HMMs), e.g. [12]. Pardo and Birmingham [14] use an augmented HMM whose transitions are modified using structural information from a musical score such as repeats and codas. Schwarz et al. [18] present a two-level HMM that can follow polyphonic scores, though only for MIDI input. Note that methods such as these could be used in conjunction with our work to adapt to the changing tempo of a performer after an initial accompaniment is created.

Buchholz et al. [3] and Klein [10] go beyond standard score-following by constraining/correcting the musical performance of a soloist. The authors hand-coded various rules of jazz improvisation, and the resulting system allows users of varying levels of musical expertise to partake in jazz improvisation. A user performs using a MIDI instrument while the system follows a provided score, plays a predetermined accompaniment, and modifies the user's input to follow the coded constraints.

Automatic Harmonization and Chord Generation

Work on "automatic harmonization" generates monophonic tracks as harmonies for a melody. Allan and Williams [1] use two HMMs to generate chorales in the style of J.S. Bach. The first HMM is used to select a sequence of note intervals to accompany each melody beat, and the second produces finer-scale ornamentations. This harmonization model uses chords as an intermediate representation but is geared toward generating harmony lines for a melody. Gang et al. [8] address the same problem, using chords as an intermediate representation in a neural network.

There is also a small set of prior work on generating chords automatically. Cunha and Ramalho [5] created a system that selects accompanying chords for a melody in real time, i.e., while the melody is performed. Their system combines a neural network with a rule-based approach for detecting recurring chord patterns. Though our underlying model differs significantly from theirs, the target output is similar. Note that our algorithm is capable of predicting chords in real time as well; however, this is a fundamentally different problem than the one we address, as: (a) future information about a melody is unavailable in real time, and (b) predicted chords may interfere with a vocalist's melodic intentions. Paiement et al. [13] use a multilevel graphical model to generate chord progressions to accompany a given melody. Though their model allows for longer-term dependencies than an HMM, it relies on songs being precisely 16 measures long. Chuan and Chew [4] use a series of musical rules combined with a data-driven HMM to generate chord progressions for melodies, but this work is not interactive and does not use vocal input.

We emphasize that none of the work we have reviewed has solved the problem we address: allowing musical novices to generate accompaniments for vocal melodies. Furthermore, none of the systems have been formally evaluated in terms of subjective quality by independent raters. We feel this is an important step in building an interactive system for this task, and as such have invested significant effort into evaluating the quality of our generated output. We describe these efforts in detail in a later section.

MYSONG

Overview and Design Goals

We first describe the process of creating music with MySong from a user's perspective, and then describe MySong's implementation.

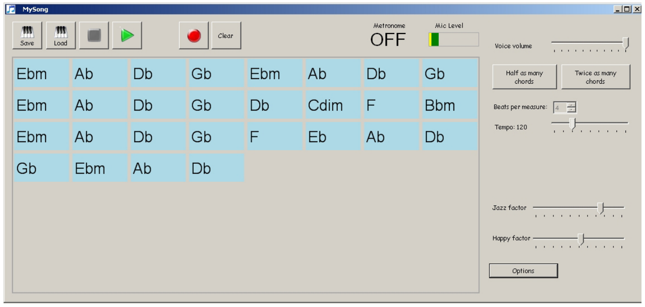
It is important to note that there is not a single correct accompaniment for a particular melody; chord selection will vary among musicians and genres, and a single musician may recognize many appropriate chord sequences for a single melody. Therefore our goal in designing MySong was not to predict the "correct" chords for a given melody, but to produce subjectively appropriate chords, and to allow those chords to vary broadly – always maintaining subjective quality – according to a small set of parameters that are intuitive to a non-musically-trained user.

Interacting with MySong

Figure 1 shows the user interface for MySong. Creating music with MySong begins with recording a vocal melody; the user presses a "record" button and sings along with a computer-generated beat at a user-specified tempo. When the user stops singing, MySong immediately generates a chord sequence that is appropriate for the performed melody. The user can listen to these chords as a piano accompaniment, along with the recorded vocal audio, using familiar "play" and "stop" buttons.

Because there are many accompaniments that are appropriate for a given melody, MySong allows the user to adjust the chords chosen by the system using parameters that are intuitive to non-musicians. One slider allows the user to make the accompaniment happier or sadder; this slider is called the "happy factor". Another slider, the "jazz factor", allows the user to bias the system toward chord patterns that are more traditional or more adventurous. Regeneration of chord sequences is immediate when adjusting these sliders, so users can rapidly explore a variety of accompaniments. Accompaniments can be saved as audio files with or without vocals or as MIDI files.

The recording and playback controls and the happy/jazz slider bars, along with a single slider used to set the tempo of the song, comprise the full set of tools used by non-musicians in the evaluation we describe later.



Metronome OFF

Beats per measure: 4

Tempo: 120

Ebm	Ab	Db	Gb	Ebm	Ab	Db	Gb
Ebm	Ab	Db	Gb	Db	Cdim	F	Bbm
Ebm	Ab	Db	Gb	F	Eb	Ab	Db
Gb	Ebm	Ab	Db				

Figure 1. MySong's user interface. A user can create an accompaniment using the record and playback controls (top) and can vary the musical style of that accompaniment using the "jazz factor" and "happy factor" sliders (lower-right).

Implementation

At its core, MySong uses a Hidden Markov Model to represent a chord sequence and its relationship to a melody. This model essentially represents which melody notes frequently co-occur with each type of chord, and which chords typically precede and follow other chords in the database. The model is trained using a large database of popular music. In this subsection, we describe the process of training this model, then we describe the process of using this model to create chord sequences for vocal melodies, and the mathematical interpretation of the parameters (the "jazz factor" and the "happy factor") available to the user.

This section explains relevant musical concepts, but assumes the reader is familiar with Hidden Markov Models (HMMs). Readers unfamiliar with HMMs are referred to [16] for an overview. We provide sufficient detail for a reader to implement our method, but note that the evaluation we present below and our supplementary material do not depend on understanding these details.

Training Data

We collected a database of 298 "lead sheets", each of which contains a melody and the associated chord sequence. Approximately half of the lead sheets came from wikifonia.org, a public repository of lead sheets. The other half of the lead sheets came from a private collection. The lead sheets in our database reflect popular genres including pop, rock, R&B, jazz, and country music.

Preprocessing

The model-training process begins with some preprocessing of the training database that simplifies further analysis.

The total number of unique chords contained in the database is extremely large, and training a model that treats all of these chords independently would lead to very limited training data for each chord. Musically, chords can be classified into five primary "triads" that contain the three core notes in a chord. Most chords in our database correspond precisely to one of these five types; we refer to the more complex chords as "extended chords". Simplifying an extended chord to its core triad removes some of its associated emotive character, but does not significantly affect the degree to which a chord is musically appropriate for a melody segment or chord sequence. Therefore, we simplify each extended chord in the database to its core triad[1]. We note that our model makes no intrinsic musical assumptions, so given additional training data, it could be re-trained to account for arbitrary chord types.

Popular songs are generally classified into one of 12 musical "keys"; a key essentially represents a distribution of frequently-occurring notes and chords. Key information was available for all of the songs in our training database. A song written in one key can be "transposed" (shifted) to another key simply by increasing or decreasing all pitches in the song equally, without affecting its subjective character. Therefore, we transpose all songs in our database to a single key (C) without any loss of generality.[2]

Learning Chord Transition Probabilities

The next step in the model-training process learns the statistics governing transitions among chords in a song, independent of melody. To do this, the system examines the chord changes in each song in the database and counts the number of transitions observed from every chord type to every other chord type. We treat 'beginning of song' and 'end of song' as 'virtual' chord types. Including these, there are a total of 62 chord types in our database.[3] We thus prepare a table with 62 rows and 62 columns, in which each cell represents the number of times a transition occurred between the corresponding two chords in the database. We will refer to this table – the first of two tables output in our training process – as the *chord transition matrix*.

We can normalize an individual row of this table to compute the probability of each possible chord following a known chord in a chord sequence.

Learning Melody Assignment Probabilities

The next step in the model-training process learns the statistics governing which notes are associated with each chord type. We remind the reader that the chord sequences in our database are non-overlapping sequences in time. We can therefore look at the period during which each individual chord is playing, and count the total duration of each musical note occurring in the melody fragment corresponding to this period. These summed note durations are then inserted into a table containing the total duration of each note observed over each chord for all songs in the database. This table has 60 rows (one for each chord type, excluding the 'start song' and 'end song' chord types) and 12 columns (one for each of the 12 musical notes). Each row of this table is then normalized so it sums to 1.0, so each element of the table represents the probability that we would see each pitch if we looked at one instant in time during which a given chord is being played. We refer to this table as the *melody observation matrix*. Figure 2 shows a graphical representation of a row of this table.

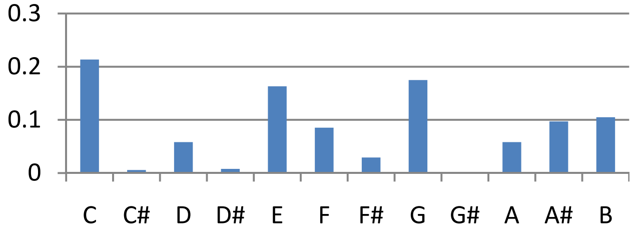


Figure 2: A graphical illustration of a row in our melody observation matrix. The musical notes 'C', 'E', and 'G' in the melody were particularly likely to coincide with this chord.

Since certain notes are very unlikely to appear when certain chords are playing, many combinations of notes and chords will have no observed data. We add a few "imaginary" instances of every note observed for a short duration over every chord; these imaginary note durations are very small relative to any note durations observed in the database. This has the effect of removing zeros in this table and smoothing the distribution somewhat. This manipulation is known in machine learning terms as a "conjugate prior".

Major/Minor Clustering

In practice, each of the 12 musical keys generally occurs in one of two distinct "modes": major and minor[4]. In other words, the set of musical notes associated with each key can be played in sequences that sound happy (major) or sad (minor). These modes are typically associated with different distributions of chords and chord transitions, so we wish to learn those distributions to allow interactive adjustment of the emotional quality of MySong's accompaniments.

Lead sheets, including those that comprise our database, generally contain labels indicating the key of each song, but do not generally include labels indicating each song's mode. If we are to learn the chord distributions representing the major and minor modes, we need to assign training labels to our database indicating whether each song represents a major or minor mode. We thus use a novel, automated clustering procedure for separating our database into major and minor sub-databases. The steps described above for building the chord transition matrix are in practice performed independently for each of these two sub-databases. At present, we do not learn independent *melody observation matrices* for the two sub-databases, as pilot testing indicated that melody observation matrices are similar across modes.

Our procedure for partitioning the database begins by initializing each song with a "guess" regarding its mode, according to a series of simple musical heuristics. We note

[1] We use the major, minor, diminished, augmented, and suspended triads. The "suspended" category includes suspended-seconds, suspended-fourths, and chords appearing with no third.

[2] Songs that include multiple keys are essentially processed as separate songs, one for each single-key region. This represents a small minority of our database.

[3] This refers to the 12 root notes times the 5 triads, plus 2 chord types representing the beginning and end of a song.

[4] At present we handle only the major and minor modes; future work and further data collection will enable independent treatment of less-frequently-used modes.

that this is the only aspect of our entire system that uses any musical heuristics, and that these are not precise rules, only initial guesses for a clustering routine[5]. Using these heuristics, we separate the database into "major" and "minor" songs and build the chord transition matrix for both sub-databases as we describe above. We then examine each song in the database and compute the probability of the complete sequence of chord transitions observed in that song, according to the current "major" and "minor" chord transition matrices and re-label it according to which probability is higher. We then build a new chord transition matrix for each mode based on the new set of assignments and repeat the process. This continues until no song changes its classification from major to minor or vice-versa.

The final output from the entire database-processing procedure consists of the chord transition matrices for each mode (major and minor) and the melody observation matrix. This entire training procedure runs once and takes approximately ten minutes. The procedure needs to be repeated only if the training database changes.

MySong: Generating chords for a new melody

We now turn our attention to the interactive MySong application, which uses the matrices produced in our database-processing steps to generate backing chords for a voice (audio) track. As we describe above, a user records a melody with a microphone, and MySong generates chords to accompany that melody. The process of generating chords to accompany a new melody makes the following two assumptions:

1) The voice track being analyzed was performed along with a computer-generated beat and was therefore at a consistent tempo; i.e. it is not necessary to extract timing information from the voice track.
2) Chords are generated at a fixed interval that corresponds to a specific number of beats at the known tempo. We call the duration associated with each chord a "measure", although we highlight that the number of beats in a measure can be changed arbitrarily after a song is recorded, and should not necessarily be interpreted to correspond to the same "measure" that would appear in a lead sheet. For purposes of the present discussion, a "measure" will refer to the fixed duration of each chord in an accompaniment.

Pitch-tracking

The first task in accompanying a new melody is to compute the pitch of the recorded voice. We use the autocorrelation technique proposed by [2], but we do not perform the subsequent dynamic programming step typically performed by pitch-trackers, which primarily serves to eliminate octave errors. We assume that octave information is not relevant to harmonization, so we are able to accelerate MySong's pitch-tracking by bypassing this stage.

The pitch-tracker computes the fundamental frequency of the audio track at 100 sample points per second of audio; we note that this interval is much smaller than the durations that would typically be assigned to musical notes. We do not attempt to extract note durations or timing or otherwise musically interpret the user's rhythmic intentions. It is to our advantage that this frequency pattern will be somewhat noisy but will tend to center around the intended pitch – or at least will contain some content at the intended pitch – for most singers; this provides significant robustness to minor pitch errors that would be lost if we quantized this signal in time into musical notes. This insight allows MySong to produce accompaniments for vocal melodies that cannot be precisely converted into a sequence of musical notes.

The measured frequencies will often not line up precisely with standard musical note frequencies, since the user may be slightly offset in pitch from the nearest "proper" musical key (set of frequently-used notes). We thus find the frequency offset that minimizes the mean-squared-error between each frequency sample and the nearest note in the standard musical scale, and shift the entire sequence of frequencies by this amount. Following this shift, each sample is discretized in frequency to one of the 12 standard musical notes. Octave information is then discarded (i.e., all pitches are shifted into a single octave).

Computing chord/melody probabilities at each measure

For each measure in the recorded melody, MySong sums the total number of samples within this measure that match each of the 12 musical notes. This gives us a 12-element vector $\boldsymbol{x}$ that is equivalent in form to the rows of the melody observation matrix (Figure 2). For each of the 60 possible chord types, we use the distribution of notes that typically appear with this chord (i.e., the appropriate row of the melody observation matrix) to measure how likely it is for the observed distribution of notes (the notes actually recorded by the user) to have occurred assuming this chord is playing. We compute this by taking the dot product of the observation vector $\boldsymbol{x}$ with the log of the appropriate row of the melody observation matrix; this yields the log-likelihood for this chord. For each measure in the recorded voice track, MySong stores a list containing all 60 of these *observation probabilities*.

As we discuss above, the melody observation matrix reflects songs that were transposed into the key of C, so we are implicitly assuming for the moment that the melody we are examining is also in the key of C. There is no reason that this key is more likely than any other key, and we will show shortly how we generalize to all possible keys.

The pitch-tracking and chord-probability-computation steps are run once for every melody recorded by a user, and require approximately two or three seconds of computation time. The subsequent steps, which allow us to choose a chord sequence given the list of chord probabilities at each

[5] The values used to initialize the sets of major- and minor-mode songs are the ratio of I to vii chords, the ratio of IV to ii chords, and the ratio of V→I and III→vii transitions, with higher ratios suggesting the major mode in each case.

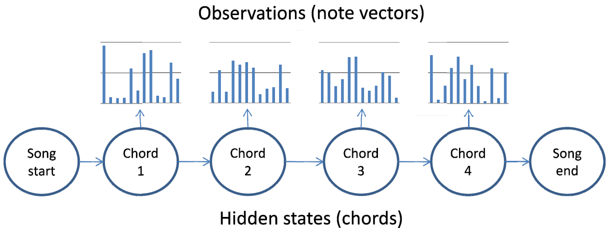


Figure 3: A graphical representation of the Hidden Markov Model used by MySong to compute a chord sequence.

measure, are extremely fast (typically less than 5ms) and are performed in real-time as the user adjusts parameters.

Choosing the best chord sequence

MySong uses a Hidden Markov Model (HMM) to model a song, where each HMM node represents one measure. We will use terminology consistent with standard HMM representations here, and we assume the reader has some familiarity with Hidden Markov Models. The possible states for each node are the 60 possible chords, and the observed data are the pitch vectors $\boldsymbol{x}$ at each measure. Given these data, we use the Viterbi algorithm [7], which computes the most likely sequence of states for a Hidden Markov Model, to compute the most likely sequence of chords. The probability of generating each observation vector $\boldsymbol{x}$ given all possible chords proceeds as we describe in the previous section. State transition probabilities are drawn directly from the chord transition matrix, which we computed when we processed our training database. We note that two chord transition matrices actually exist, one for major-mode songs and one for minor-mode songs. We will assume for now that we are generating chords for a major-mode song; we explain below how we handle the two modes.

Figure 3 shows a graphical representation of our model. We note that the transitions from "song start" to each possible chord, and the transitions from each possible chord to "song end", were represented in the chord transition matrix we built from our database, so these do not require special handling at this stage.

Given all of these probabilities, the Viterbi algorithm chooses the most likely chord sequence for the recorded melody and assigns the selected chords to the model nodes.

The "Jazz Factor": Implementation

To compute the probability of each chord at each measure in the forward part of the Viterbi algorithm, we need to assume that each chord is the "correct" chord for this measure, and sum the corresponding log-probability of the observed notes with the log-probability of the transition to this chord from each possible chord at the previous node. We introduce a weighting factor at this stage – the *jazz factor* – that explicitly multiplies the log-probability of the observed notes. This observation weight helps the system choose between "good chord progressions" (low weighting of the observed notes) and "chords that very closely fit the input notes" (high weighting of the observed notes). In practice a high value for this parameter makes the selected chords more "interesting" or "surprising", while a low value for this parameter tends to produce fairly conventional chord progressions. This value is mapped directly to the user-controlled *jazz factor* slider (Figure 1).

The "Happy Factor": Implementation

Thus far we have assumed that we are generating chords for a major-mode song and thus have a single chord transition matrix. In practice, as we describe above, we actually have chord transition matrices for both the major and minor modes. In order to allow the user to draw on the emotional characteristics of both modes, we actually blend the two transition matrices according to the "happy factor" slider shown in Figure 1. When the slider is all the way to the right, we use the major-mode chord transition matrix, which tends to produce "happier" chords. When the slider is all the way to the left, we use the minor-mode chord transition matrix, producing "sadder" chords. When the slider is anywhere in between, we take a combination of the two transition matrices to produce an intermediate transition matrix. This allows the user to gradually adjust the feel of an accompaniment from "happy" to "sad".

Key Determination

Thus far, we have assumed that the input melody is in the key of C, so we could directly use the transition and observation matrices computed from the training database. In practice, there is no reason to believe that the melody is in the key of C, so we repeat the entire procedure described in this section twelve times, once for each of the twelve possible keys. At each iteration, we assume the melody is in the corresponding key and shift the entire melody into the key of C, then run the chord-generation procedure described above. We record the overall probability of the chord sequence chosen for each key, and choose the key with the highest associated probability.

This procedure runs extremely quickly for typical songs, so the computation time required to process twelve keys (less than 50ms on a typical PC) is not a significant issue. In practice the bulk of the time spent processing a melody is devoted to pitch-tracking, which does not need to be repeated for each possible key or each adjustment of the user-specified parameters.

Accompaniment Generation

We have now described the procedure MySong uses to select chords for a vocal melody. At present, we use a simple, pre-defined pattern of chord-dependent piano notes to play this chord sequence back to the user as an accompaniment for his voice. We point out that software exists for taking a chord sequence and generating an audio accompaniment [15]. MySong is able to directly automate this software to generate more diverse instrumental accompaniments once chords have been selected.

In the following sections, we will discuss two studies we conducted to evaluate MySong.

EVALUATION 1: EVALUATING CHORD SELECTION

The first experiment we conducted assesses the ability of MySong's automatic chord-generation algorithm to produce subjectively appropriate chord patterns for vocal melodies, relative to a human musician and a state-of-the-art commercial system for generating chord sequences. The only commercially-available system for generating chord sequences, to our knowledge, is "Band-in-a-Box" (BIAB) [15], which is primarily a system for generating accompaniment audio *from* chords, but includes a module for determining chords from a melody. The BIAB system represents the state of the art in determining chords from a musical melody, but was not designed for vocal input, which cannot yet be reliably and automatically converted to a "clean" musical melody. We therefore are *not* evaluating the *quality* of BIAB's chord selection mechanism per se; rather, we use this comparison to highlight the importance of designing a chord-selection system specifically for vocal audio. Anecdotally, when processing melodies read directly from sheet music, BIAB's chord-selection system does quite well, and our evaluation should *not* be used to judge the quality of this component of BIAB.

It is also important to note that there is not a single correct accompaniment pattern for a particular melody; chord selection will vary among musicians and among genres, and a single musician may recognize multiple accompaniment patterns for a single melody. Therefore our goal in conducting this experiment was not to produce and compare "correct" chords for a given melody, but to test objectively the following two hypotheses:

1) MySong can be used to rapidly produce chord sequences that are, in terms of subjective quality, in the *range* of human-assigned chords.
2) MySong produces chord sequences that are, in terms of subjective quality, superior to a state-of-the-art system designed for selecting chords for musical melodies but not vocal audio.

Evaluation 1: Methodology

Accompaniment Preparation

Twenty-six vocal melody clips were recorded, all by the same vocalist, ranging from thirteen to twenty-five seconds in length. Only melodies by independent artists were used, to ensure novelty to evaluators, and none of the melodies had been supplied to any accompaniment system at any point prior to the experiment.

In the interest of complete disclosure, we note that approximately half of these melodies were authored by the experimenters, but all were authored more than six months before the initial conception of any aspect of this project and were never tested with any accompaniment system prior to this experiment. The motivation for this decision was to allow us to release, in support of this paper, data used for this evaluation, which would not have been possible with melodies under commercial copyright. We highlight that the supplementary material for this paper thus includes a set of melodies and accompaniments that *were not filtered at all based on the output of each accompaniment system*. We felt this was important to allowing readers to judge the output of our system.

Each melody was loaded into MySong by two trained experts (both experienced musicians), who were given no more than five minutes to adjust the two free parameters (the "happy" and "jazz" factors) and reach consensus on an appropriate chord pattern. Note that it would have been unfair to both MySong and BIAB to allow no manual intervention at all, since both are *designed* to allow limited user input into chord selection. Experts were not allowed to edit chords manually, change keys, or perform any other operations that would not be accessible to a target (non-musically-trained) user. In practice, two minutes was long enough in each case to reach consensus.

The pitches transcribed from each melody were exported to a MIDI file and loaded into BIAB along with the vocal audio. The same two experts, also trained in BIAB, were again given no more than five minutes per song to adjust the parameters affecting chord selection in this system. Again, experts were not allowed to perform any operations that would not be accessible to a non-musically-trained user, such as editing chords or changing keys.

The same two experts also used traditional musical mechanisms for assigning chords to a melody; they had access to musical instruments and were allowed to listen to the melody as needed. Again, five minutes were allowed for each melody, and assigners were instructed to reach consensus on a subjectively-acceptable set of chords.

The three chord sequences for each song were rendered to audio files using a fixed pattern of synthesized piano notes.

The decision to use the same audio recordings for all conditions limits MySong, which is designed as an interactive system that encourages multiple recordings. However, to present a fair comparison across conditions, it was necessary to accept this limitation. This evaluation thus reflects only MySong's core algorithm, not the complete application. We would expect a user able to fully explore a melody with MySong to produce even better results.

Accompaniment Evaluation

30 volunteers, all musicians recruited through a musician-specific mailing list, were asked to download an application that presented each of the 26 melodies in a randomized order. Each melody was presented as a pair of accompaniments, from two of the three systems. We note that within a pairing, the vocal audio was identical; the only variation arose from the chords selected by the systems represented. The order in which accompaniments were selected was randomized, but each participant saw each possible ordered pairing of the three systems (MySong, manual, BIAB) four times. Since there are six such ordered pairings and 26 melodies, each participant also saw two additional pairings that were selected at random.

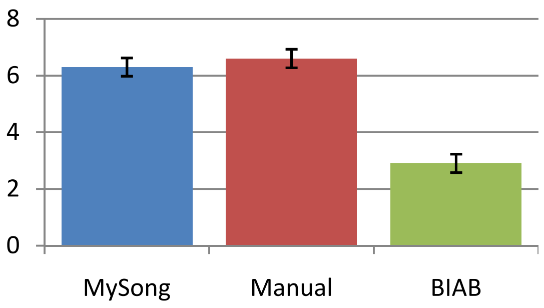


Figure 4: Mean scores assigned to accompaniments produced by in each condition: MySong, manual-assignment, and Band-in-a-Box. MySong closely approaches the rating assigned to manually-assigned accompaniments.

Participants were asked to rate each accompaniment on its subjective quality and subjective appropriateness for the melody, on a scale of 1 to 10. Though subjective preference will inevitably lead to variation in the spread of participants' responses within this scale, participants were specifically instructed to carefully consider the *relative* subjective quality of each pair of accompaniments and assign scores such that the preferred accompaniment received a higher score. Tied scores within a pairing were allowed. Participants were required to listen to both accompaniments before they could assign ratings, and to rate all songs before completing the experiment.

Evaluation 1: Results

Mean scores assigned to the MySong, manual-assignment, and BIAB conditions were 6.3, 6.6, and 2.9, respectively (Figure 4). Because diverse raters were asked to subjectively rate accompaniments under each condition (MySong, BIAB, and manual-assignment), we performed a two-way ANOVA to isolate the effect of condition on rating. ANOVA results show main effects of both condition (F(1,2) = 1166, $p < 0.01$) and rater (F(1,29) = 29, $p < 0.01$). The effect of rater is expected; a large group of raters will inevitably have somewhat divergent means on a subjective scale. We are interested in the effect of condition, and explore this in post-hoc tests, which reveal a significant difference between the MySong and BIAB conditions, and between the manual and BIAB conditions, but *not* between the MySong and manual conditions.

Figure 4 shows the mean scores assigned to accompaniments in each of the three conditions along with the standard error of each mean; we highlight that MySong closely approaches the ratings given to manually-assigned chord sequences. We also counted the number of times each condition was preferred in a direct comparison; these results are presented in Table 1. When a MySong accompaniment was paired directly against a manual accompaniment, the manual accompaniment was preferred 121 times, the MySong accompaniment was preferred 95 times, and 48 times the accompaniments were assigned identical scores. These results are extremely encouraging and support the claim that MySong produces appropriate accompaniments.

We stress once again that these chord sequences were by no means the only appropriate sequences for each melody, and the goal of this experiment was to show that MySong's chords were in the range of manually-assigned chords.

Comparison	Wins for each Condition
MySong vs. Manual	MySong 95, Manual 121, Ties 48
MySong vs. BIAB	MySong 242, BIAB 6, Ties 14
Manual vs. BIAB	Manual 235, BIAB 6, Ties 13

Table 1: The "win/loss records" of each condition when paired against each other condition.

EVALUATION 2: SONG ACCOMPANIMENT BY NOVICES

While our first evaluation demonstrates that MySong can be used to rapidly select chords that are subjectively appropriate for a vocal melody, it does not demonstrate the system's broader goal: allowing users with no knowledge of music theory to produce musical accompaniments. Our second study aims to evaluate MySong's usability for musically-untrained users.

Evaluation 2: Demographics

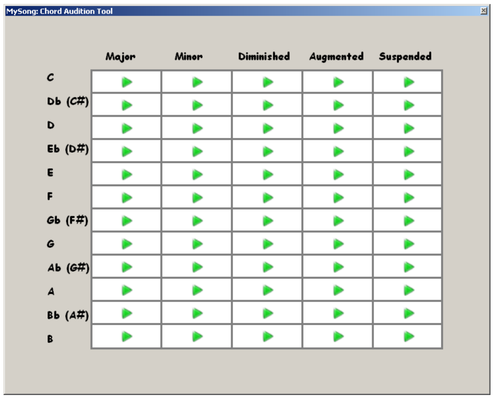
Thirteen participants (eight male) were recruited from a pool of information workers; a call for participants requested volunteers who had no background in chords or music theory, who could "carry a tune" but were not necessarily "good" singers, and who were willing to sing several short song clips for an experiment. Prior to the experiment, we interviewed participants to confirm that they did not have any background in chords or music theory. We did not remove any participants based on vocal ability, and will show below that our participants demonstrated a wide range of vocal skills.

Evaluation 2: Methodology

Before our experiment, participants were asked to select five melodies, each 20 to 30 seconds in length, which they were comfortable singing as part of the experiment. This typically represented one verse or one chorus from a popular song. Four of these melodies were used in our experiment, and one was designated as a "practice" song for use when initially learning the tools used in the experiment. The four melodies used for the study were randomly assigned to the two experimental conditions (described below) before the experiment began.

Participants were instructed to imagine they were the lead singer in a band and wanted to do a "re-make" of their selected songs. Their task for the session was to prepare appropriate chord sequences for each song that they would provide to their "band members". Participants were told that they could diverge from the chords or feel of the original songs; their goal was to prepare a subjectively-appropriate accompaniment.

We felt it was important to evaluate MySong relative to a comparison system in order to quantify MySong's utility in enabling users to complete this task. Since we specifically recruited participants who are unfamiliar with chords, it would have been unreasonable to ask participants to manually select chords from the unfamiliar vocabulary of



MySong: Chord Audition Tool

	Major	Minor	Diminished	Augmented	Suspended
C	▶	▶	▶	▶	▶
Db (C#)	▶	▶	▶	▶	▶
D	▶	▶	▶	▶	▶
Eb (D#)	▶	▶	▶	▶	▶
E	▶	▶	▶	▶	▶
F	▶	▶	▶	▶	▶
Gb (F#)	▶	▶	▶	▶	▶
G	▶	▶	▶	▶	▶
Ab (G#)	▶	▶	▶	▶	▶
A	▶	▶	▶	▶	▶
Bb (A#)	▶	▶	▶	▶	▶
B	▶	▶	▶	▶	▶

Figure 5: The "Chord Audition Tool" used in our comparison condition.

music. To our knowledge, no alternative systems exist for allowing non-musicians to create accompaniments, so we created a comparison system we refer to as the "Chord Audition Tool" (Figure 5).

In this condition, users were presented with a grid of musical note names and chord types; clicking any space in the grid played the corresponding chord. Participants could explore this grid while listening to their recorded melodies to find chords that matched their recordings, and used a simple GUI to assign a selected chord to each measure of their recording. While this is still a difficult task, we stress that it makes it possible for users who are unfamiliar with chords to experiment with chords and pitches, and we will highlight below that many participants in fact *enjoyed* and learned from this tool, and in some cases were able to complete the task this way despite having no familiarity with chords. Consequently, we feel that this condition not only represents but *exceeds* the state-of-the-art tools available to non-musicians for experimenting with chordal accompaniments.

Each participant prepared accompaniments for two songs using MySong, and two songs using the Chord Audition Tool. Participants were allotted ten minutes per song, and were allowed to record the song as many times as was needed in both conditions within that ten-minute period. The order of conditions was counterbalanced across participants. Each participant received a brief tutorial on each system before working with it, and was given up to five minutes to experiment with each system using his or her selected "practice song". At the end of the session, participants completed a Likert-scale questionnaire to assess their impressions of each system.

Subsequent to all participants' sessions, three musicians listened, in random order, to all four accompaniments produced by each participant and independently rated each accompaniment on a subjective scale from 1 to 10. Note that the evaluators did not know which tool had produced each accompaniment. Evaluators also scored the *vocal* performance in each recording, and were asked to answer the yes/no question "Did this participant succeed in creating an acceptable accompaniment for this melody?"

Evaluation 2: Results

Questionnaire Data

Participants were presented with Likert-scale questions to evaluate their subjective responses to each system. Table 2 shows the mean responses for each of the four questions asked in each condition. We note that MySong received more positive responses to each of these questions than the comparison condition, and that all differences were significant according to signed-ranks tests ($p < 0.02$).

Of particular interest is the fact that the median response to the statement "I felt that I could get to a satisfactory accompaniment rapidly with [the MySong] system" was "strongly agree"; only a single participant disagreed with this statement. We are extremely encouraged by this result; participants completely unfamiliar with chords and with highly varying vocal abilities felt they were able to create chordal accompaniments for songs in just *ten minutes*, and enjoyed the experience.

Evaluation Data

Each participant was assigned a mean score in each condition across both songs and all three evaluators; the mean score assigned to each participant's MySong accompaniments was 6.4 (sd 1.3), and the mean score assigned to each participant's comparison-condition accompaniment was 3.4 (sd 1.5). This difference was significant according to a paired t-test ($p < 0.01$).

We considered a participant to have successfully achieved the goal of producing an appropriate accompaniment for a melody if all three evaluators agreed that the participant completed the task. According to this criterion, 73% of all accompaniments created using MySong were considered appropriate, compared to 23% in the comparison (Chord Audition) condition. Similarly, 85% of participants created at least one appropriate accompaniment using MySong, compared to 31% for the comparison condition. These results are summarized in Table 3.

Evaluators scored each vocal performance on a scale of 1 to 10, where 10 represented a nearly-professional singer, 5 represented a "typical Karaoke-goer", and 1 represented nearly non-pitched voice. We note that scores assigned ranged from 1 to 10, with the mean score for all performances by each participant ranging from 3.0 to 8.3. The overall mean score was 4.9. We present this data only

Question	MySong	Comparison
I enjoyed using this system.	4.3 (0.6)	2.5 (1.0)
I felt I was able to create music that sounded good using this system.	4.1 (0.8)	2.4 (1.2)
I felt that I could get to a satisfactory accompaniment rapidly with this interface.	4.3 (0.9)	1.8 (1.0)
I would use this interface for fun if I had this software.	4.3 (0.8)	2.8 (1.3)

Table 2: Mean responses to five-point Likert-scale questions (5 = strongly agree) presented to participants after using both systems to create accompaniments, with standard deviations.

	MySong	Comparison
Mean accompaniment score for all participants	6.4 (1.3)	3.4 (1.5)
Overall percentage of successful accompaniments	73%	23%
Percentage of users successfully accompanying at least one melody	85%	31%

Table 3: A comparison of participants' accompaniment success in the MySong and comparison conditions.

to confirm that the participants in this study were not necessarily exceptional vocalists, so the utility of MySong that is demonstrated by this evaluation can be expected to apply to users with a broad range of vocal abilities.

DISCUSSION

We believe that our first evaluation, coupled with the supplementary media provided with this paper, confirm that MySong is able to rapidly produce subjectively-appropriate accompaniments for vocal melodies, and validates the core algorithms presented in this paper.

We are particularly enthusiastic about the results of our second evaluation. We highlight that participants in this study were able to create musical accompaniments in less than 10 minutes following only a brief tutorial. We ask readers who are unfamiliar with chords and harmony to imagine just how difficult it would be to complete this task in ten minutes, and we ask musically-trained readers to imagine completing this task before your musical training.

All participants gave positive subjective ratings to MySong, and indicated an interest in continued use of MySong. We imagine that given more time to work with the system, participants could produce even better results, and could begin experimenting with creative aspects of music that would otherwise be completely inaccessible to them. Further evaluation will focus on MySong's potential not only as an accompaniment-creation tool, but as a creativity-support and songwriting tool; we hope to demonstrate that novices can create *original* music with this system.

CONCLUSION AND FUTURE WORK

Additional development will focus on improving and diversifying the audio generated by MySong; the system is already able to supply chords interactively to a pattern-based arrangement tool [15], which results in compelling audio output (examples are available at the supplementary-materials URL provided above). Several study participants indicated that MySong would be of significant value for *learning* music theory; we are thus excited about exploring educational applications of this technology.

In conclusion, the contributions of this paper are:

1) We describe a system that automatically selects chords to accompany a vocal melody; this system can be stylistically guided according to intuitive parameters.

2) We present results from two evaluations of this system, one demonstrating the subjective quality of the system's output, and the other demonstrating that participants with no knowledge of chords are able to create musical accompaniments using our system.

ACKNOWLEDGMENTS

We thank Scott Switzer and Wikifonia for training data, Jim Truher for scripting, Meredith Morris for proofreading, Rico Malvar for FFT code, and reviewers for feedback. We finally thank our study participants, who courageously volunteered to sing in an unfamiliar environment.

REFERENCES

1. Allan, M., Williams, C.K.I. Harmonising Chorales by Probabilistic Inference. NIPS 2005.
2. Boersma, P, Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. Proc Inst Phonetic Sci, v17, 1993.
3. Buchholz, J., Lee, E., Borchers, J. coJIVE: A System to Support Collaborative Jazz Improvisation. Technical Report, AIB-2007-04, RWTH Aachen Univ, 2007.
4. Chuan, C.-H., Chew, E. A Hybrid System for Automatic Generation of Style-Specific Accompaniment. 4th Intl Joint Workshop on Computational Creativity, June 2007.
5. Cunha, U., Ramalho, G. (1999). An Intelligent Hybrid Model for Chord Prediction. Organised Sound 4(2), 115-19. Cambridge University Press.
6. Farbood, M., Pasztor, E., Jennings, K. Hyperscore: A Graphical Sketchpad for Novice Composers. IEEE Comp Graph and Appl, Jan 2004.
7. Forney, D. The Viterbi algorithm. Proc IEEE 61(3), 1973.
8. Gang, D., Lehman, D., Wagner, N. Tuning a Neural Network for Harmonizing Melodies in Real-Time. ICMC 1998.
9. Grub, L., Dannenberg, R. A Stochastic Method of Tracking a Vocal Performer. ICMC 1997.
10. Klein, J. A Pattern-Based Software Framework for Computer Aided Jazz Improvisation. PhD Thesis, RWTH Aachen, 2005.
11. Obrenovic, Z. A flexible system for creating music while interacting with the computer. ACM MM 2005.
12. Orio, N., Déchelle, F. Score Following Using Spectral Analysis and Hidden Markov Models. ICMC 2001.
13. Paiement, J.-F., Eck, D., Bengio, S. Probabilistic Melodic harmonization. Proc Canandian Conf AI, 2006.
14. Pardo, B., Birmingham, W. Modeling Form for On-line Following of Musical Performances. AAAI 2005.
15. PG Music Inc: Band-in-a-Box. http://www.pgmusic.com
16. Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications. Proc IEEE, 77(2) (1989).
17. Raphael, C. A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores. ICMIR 2004.
18. Schwarz, D., Orio, N., Schnell, N. Robust Polyphonic MIDI Score Following with Hidden Markov Models. ICMC 2004.
19. Weinberg, G. Interconnected Musical Networks – Bringing Expression and Thoughtfulness to Collaborative Music Making. PhD Thesis, MIT, 2003.