SEARCHING LYRICAL PHRASES IN A-CAPELLA TURKISH MAKAM RECORDINGS

First author
Affiliation1
author1@ismir.edu

Second author
Retain these fake authors in submission to preserve the formatting

Third author
Affiliation3
author3@ismir.edu

ABSTRACT

Search by lyrics, the problem of locating the exact occurrences of a phrase from lyrics in musical audio, is a recently emerging research topic. Unlike key-phrases in speech, lyrical key-phrases have durations that bear important relation to other musical aspects like the structure of a composition. In this work we propose an approach that address the differences of syllable durations, specific for singing. First a phrase is expanded to MFCC-based phoneme models, trained on speech. Then, we apply dynamic time warping between the phrase and audio to estimate candidate audio segments in the given audio recording. Next, the retrieved audio segments are ranked by means of a novel score-informed hierarchical hidden Markov model, in which durations of the syllables within a phrase are explicitly modeled. The proposed approach is evaluated on 12 a-capella audio recordings of Turkish makam music. Relying on standard speech phonetic models, we arrive at an f-measure, comparable to a state-of-the-art work for the related task of keyword-spotting in singing. To the best of our knowledge, this is the first work tackling the problem of search by lyrical key-phrases. We expect that it can serve as a baseline for further research on singing material with similar musical characteristics.

1. INTRODUCTION

Searching by lyrics is the problem of locating the exact occurrences of a snippet from textual lyrics in musical signal. However, searching by a keyword in speech is a well-investigated research task. In keyword spotting a user in interested to find at which time position in speech a relevant keyword (presenting a topic of interest) is spoken. An equivalent of keyword spotting for singing voice is locating the occurrences of lyrics keywords.

However, unlike keyword spotting, we believe that key-phrase detection has higher potential to be integrated with other relevant MIR-applications, because lyrical key-phrases bear semantics in the context of the musical idiom: a line form lyrics is correlated, for example, to musical structure. For most types of music a section-long lyrical phrase is a feature that represents the corresponding structural section (e.g. chorus) in a unique way. Therefore correctly retrieved audio segments for, for example, the first lyrics line for a chorus can serve as a structure discovery tool.

For spoken utterances phonemes have relatively similar duration across speakers. Unlike that, in singing durations of phoneme (especially vowels) have higher variation [6]. When being sung, vowels are prolonged according to musical note values. Therefore adopting an approach from speech recognition might lack some singing-specific semantics, including among others durations of sung syllables.

In this work we investigate searching by lyrics in the case when a query represents an entire section or phrase from the textual lyrics of a particular composition. Unlike keyword-spotting or query-by-humming where a hit would be a document from an entire collection, in our case a hit is the occurrence of a phrase, being retrieved only from the performances of the given composition. In this respect the problem setting is more similar to linking melodic patterns from score to musical audio [7], rather than to keyword-spotting. We assume that the musical score with lyrics is present for the composition of interest.

The proposed approach has been tested on a small dataset of a-cappella performances from a repertoire of Turkish makam music. For a given performance, the composition is known in advance, but no information about the structure is given. Characteristic for makam music is that, in a performance there might be reordering or repetitions of score sections.

2. RELATED WORK

2.1 Keyword spotting in singing

A recent work proved that keyword spotting in singing voice is a hard problem even when singing material is acapella (from pop songs in English) [6]. The authors adopt an approach from spotting keywords for speech, using a compound hidden Markov model (HMM) with keyword and filler model. Keywords are automatically extracted from a collection of lyrics. The best classifier (multi-layer perceptron) yielded an f-measure of 44% , averaged over top 50% of keywords. Notably the achieved results on singing material are not very different from results on spoken utterances of same keywords and data.

One of the few attempts to go beyond keywords is the

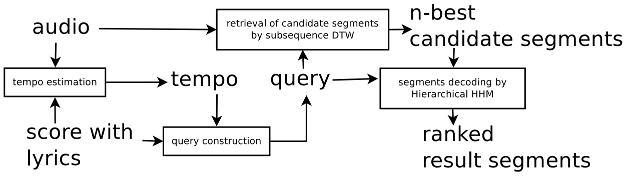
audio
retrieval of candidate segments by subsequence DTW
n-best candidate segments
tempo estimation
tempo
query
segments decoding by Hierarchical HHM
score with lyrics
query construction
ranked result segments

Figure 1. System Overview

work of [3]. Their goal is to automatically link phrases that appear in the lyrics of one song to the same phrase in another song. To this end, a keyword-filler model for detecting characteristic phrases (of 2-3 words) in sung audio. The method has been evaluated on polyphonic audio from Japanese pop. A non-HMM-based approach has been chosen in [2]. The authors propose subsequence dynamic time warping (DTW) to find a subsequence of features from a target recording similar to an example utterance of a keyword. The approach has been evaluated with f-measure of 39%.

In summary, performance of state-of-the-art work in keyword and key-phrase spotting for singing is not sufficiently good for practical applications. A probable reason for this is that hitherto approaches do not take into account the duration of syllables, which, as stated above, is an important factor that distinguishes speech from singing. In addition to that, syllable durations have been shown to be a strong reinforcing cue for the related task of automatically synchronizing lyrics and singing voice [1].

2.2 position-aware HHMMs

The modeling in most of the above mentioned approaches relies on HMMs. A drawback of HMMs is that their capability to model exact state durations is restricted. The wait time in a state is implicitly set to a geometric distribution (derived from the self-transition likelihood).

One alternative to tackle durations can be seen in hierarchical HMMs (HHMM). They allow the inclusion of more than one latent variable [K Murphy] and allow modeling of any relations among variables in terms of probabilistic dependencies. HHMMs can thus represent interdependent musical aspects that have sequential nature. Such an aspect is for example the position in musical score. In [4] it has been shown that the dependence of score position on structural sections makes it possible to link musical performances to score. Inspired by the latter work, we model the position of a musical section in a HHMM, in a way that integrates syllable durations and the features, capturing phoneme acoustics.

3. SYSTEM OVERVIEW

Figure 1 presents an overview of the proposed approach.

We suggest a two-pass retrieval approach: On the first pass a subsequence DTW retrieves a set of candidate audio segments that are acoustically similar to to a query. On the second pass each candidate segment is separately fed into the HHMM, for which we run a Viterbi decoding to assure that only one (the most optimal) path is detected for an audio segment. Any query-to-audio fullpath match is considered as a hit and all hit results are ranked according to the weights derived from the Viterbi decoded path.

3.1 Tempo Estimation

The actual tempo T in a recording is estimated by an external module, which relies on musical score. [sertan]

3.2 Query construction

A selected lyrical phrase serves as a query twice: first a *simple query* for retrieval of candidate segments and then a *duration-informed query* for the decoding with HHMM.

3.2.1 acoustic features

For each of the 38 Turkish phonemes (and for a silent pause model) a 3-state HMM is trained from a corpus of turkish speech. The 3 states represent respectively the beginning, middle and ending acoustic state of a phoneme. The acoustic properties (most importantly the formant frequencies) of spoken phonemes can be induced by the spectral envelope of speech. To this end, we utilize the first 12 MFCCs and their delta to the previous time instant. For each state a 9-mixture Gaussian distribution is fitted on the feature vector.

3.2.2 simple query

For the first step no score-position information is utilized: lyrics is merely expanded to its constituent phoneme models. Let $\lambda_n \in \Lambda$ be a model state at position n in the query, where Λ is a set of all 3×38 states for the 38 Turkish phonemes. No transition probabilities are taken into account.

3.2.3 position-informed query

The idea is to associate each particular score position p_n in a musical section s to a phoneme state λ_n by exploiting the note-to-syllable mappings, present in sheet music.

Let μ be the duration of a short note from score than can be at most one-syllable-long. In this work we opted for 64-th note $\mu = \frac{1}{64}$. For each syllable a reference duration (in units of μ) is derived by aggregating values of its associated musical notes. Then the reference duration is spread among its constituent phonemes in a rule-based manner, resulting in phoneme reference durations R_p. [1]

To query a particular performance, the unit of R_p is converted form μ to τ: number of time frames for μ according to the inferred musical tempo T:

$$\tau = NFS \frac{T}{60} \mu$$

where T is in bpm and NFS is the number of frames per second (set to 100 in this work). Therefore $p_n \in (1, 2, ..., D(s_n))$

[1] In this work a simple rule is applied: consonants are assigned a fixed duration and the rest is assigned to the vowel.

where $D(s_n)$ is the total duration for a section s_n (in units of τ). More formally we define a mapping

$$f(p_n, s_n) \rightarrow \lambda_n \tag{1}$$

that determines the true state of a phoneme sung at time n from the position p_n within a section s_n.

4. RETRIEVAL OF CANDIDATE SEGMENTS

Subsequence-DTW was proved to be effective when the feature series of an audio query are subsequence of features of a target audio [Dittmar] . In our case a query of phoneme models Λ with length M can be seen as subsequence of the series of MFCC features Y with length N, extracted from the whole recording. To this end we define a distance metric for an audio frame y_m and n^{th} model state λ_n as a function of the posterior probability.

$$d(m, n) = -log\, p(y_m|\lambda_n) \tag{2}$$

where for phoneme state model λ_n

$$p(y_m|\lambda_n) = \sum_{c=1}^{9} w_{c,\lambda_n}.N(y_m; \mu_{c,\lambda_n}, \Sigma_{c,\lambda_n}) \tag{3}$$

with N being the Gaussian distribution from a 9-component mixture with weights w_{c,λ_n}. Based on the distance metric 2 a distance matrix D^{NxM} is constructed. In the rest of the paper we will refer to an alignment path $P = \{p_l\}$ with length $l \in (1, L)$, where $p_l = (m, n)$ (a notation proposed in [Mueller]) refers to an entry $d(m, n)$ in D.

4.1 Path computation

To generate an alignment path P we select step sizes $p_l - p_{l-1} = (1, 1), (1, 0), (1, 2)$ corresponding respectively to diagonal, horizontal and skip step according to the notation of [Mueller]. A horizontal step means staying in the same phoneme in next audio frame. The step size $(0, 1)$ is disallowed because each frame has to map to exactly one phonetic model. To counteract the preference for the diagonal and the skip step $(1, 2)$ local weights w_d and w_s are introduced (as suggested in [Mueller]). We set rather high values (empirically found $w_d = 6.5$ and $w_s = 11$) to assure that a path will not exit from long vocals prematurely.

A list of candidate paths $S = \{P*\}$ is computed by iteratively detecting the current path with maximum score. After having detected a path $P*$ with final position in frame $n*$ a small region of 5% of M: $(n * -5\%M,\ n * +5\%M)$ is blacklisted from further iterations. This is an idea inspired by the machine learning principle of simulated annealing [ref] and assures that the iterative procedure will not get stuck in a current maximum and thus retrieve a high amount of paths from its vicinity. We pick 30 (a relatively high number compared to 20 in [Dittmar]), because our goal is to cover as many relevant candidates as possible.

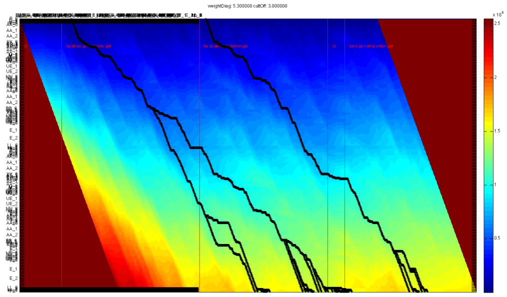

Figure 2. Example candidate audio segment of many overlapped paths (two main paths can be seen)

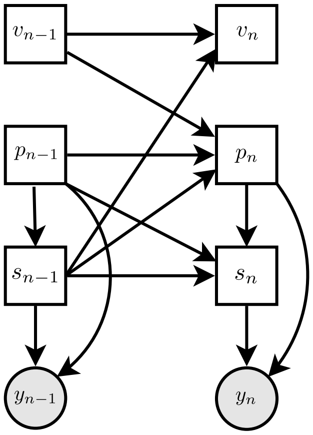

Figure 3. Representation of the HHMM as a dynamic Bayesian network. Hidden variables (not shaded) are v - velocity, p - score position and s - section. The observed feature vector y is not shaded. Squares and circles denote respectively continuous and discrete variables

4.2 Candidate segment selection

We define an overlapped audio segment to be any segment from the target audio, for which any frame y_m belongs to more than one path P. Let us call the set of paths for an overlapped audio segment an overlapped path set (see figure 2). Then as candidate segment is considered the audio segment span from the initial timestamp of the leftmost path until the final timestamp of the rightmost path. We assume that a concentration of found paths within a segment indicates at least one true hit.

5. POSITION-HHMM MODEL

In this section, we present a novel probabilistic model for modeling a lyrical phrase. The main idea of the model is to incorporate the phonetic identities of lyrics and the syllable durations, available from musical score, into a coherent model. The dependence of the observed acoustic features (that capture the phonetic identity) on musical velocity and score position are presented as dynamic Bayesian network [ref] in figure 3. The derivation of model dependencies and parameters is the main contribution of this work.

5.1 hidden variables

1. position from musical score for a section $p_n \in (1, 2, ..., D(s_n))$. $D(s_n = Q)$ is the duration (in reference time frames τ) for a section s_n as defined in section 3.2 Note that the actual time span of a reference time frame is different per recording.

2. velocity $v_n \in (1, 2, ..., V)$. The amount of τ a position pointer has to jump over at next time frame. Staying in state $v_n = 2$, for example, means that the current tempo is steady and around 2 times faster than the reference one.

3. structural section $s_n \in (Q, F)$ where Q is the queried section and F is a filler section. A filler section represents any non-key-phrase audio regions, and practically allows with equal probability being in any phoneme state (see section 5.3)

For the experiments reported in this paper, we choose $V = 5$, $S = 2$ and we set empirically $D(s_n = F) = V$. This assures that even in fastest tempo there is option of entering the filler section. In deriving $D(s_n)$ instead of T we use $2 * T$ to allow handling local tempo fluctuations to a twice-slower tempo when $V = 1$.

The proposed model is different than the model proposed in [Florian], in two aspects:

- $D(s_n)$ is not fixed but depends on the section of interest and the detected tempo of performance

- a section (a pattern in the original model) is not fixed for a recording, but can alternate between states.

Since all the hidden variables are discrete, one can reduce this model to a regular HMM by merging all variables into a single 'meta-variable' x_n:

$$x_n = [v_n,\, p_n,\, s_n] \tag{4}$$

Note that the state space becomes the Cartesian product of the individual variables.

5.2 transition model

Due to the conditional independence relations presented in figure 3, the transition model reduces to

$$\begin{aligned} p(x_n|x_{n-1}) = \quad & p(v_n|v_{n-1}, s_{n-1}) \times \\ p(p_n|v_{n-1}, p_{n-1}, s_{n-1}) \times \quad & p(s_n|p_{n-1}, s_{n-1}, p_n) \end{aligned} \tag{5}$$

5.2.1 velocity transition

$$p(v_n|v_{n-1}) = \begin{cases} pr/2, & v_n = v_{n-1} \pm 1 \\ 1 - pr, & v_n = v_{n-1} \\ 0, & \text{else} \end{cases} \tag{6}$$

where pr is a constant probability of change in velocity.

5.2.2 position transition

The score position is defined deterministically according to:

$$p_n = (p_{n-1} + v_{n-1} - 1) \mod D(s_{n-1}) + 1 \tag{7}$$

where the modulus operator resets the position to be in a beginning of a new section after it exceeds the duration of previous section $D(s_{n-1})$

5.2.3 section transition

$$p(s_n|p_{n-1}, s_{n-1}, p_n) = \begin{cases} p_s(s_n|s_{n-1}), & p_n \leq p_{n-1} \\ 1, & p_n > p_{n-1},\ s_n = s_{n-1} \end{cases} \tag{8}$$

A lack of increase in the position is an indicator that a new section should be started.

$p_s(s_n|s_{n-1})$ is set according to a standard transition matrix where self transitions p_q and p_f for query and filler section respectively can be set according to the expected structure of the target audio signal. In this work we set $p_q = 0$ and$p_f = 0.9$ to impose a low chance of decoding the query section more than once in a candidate audio segment.

5.3 Observation model

The probability of the observed feature vector in position p_n from section s_n is computed for the model state λ_n given by the mapping function $\lambda_n = f(p_n, s_n)$ 1 derived in section 3.2. A similar mapping function has been proposed for the first time in a position-based HHMM in [Andre].

Then

$$p(y_n|p_n, s_n = Q) = p(y_n|\lambda_n)$$

which reduces to applying the distribution defined in 3.

In case of the filler section we allow with equal probability any phoneme state.

$$p(y_n|p_n, s_n = F) = \max_{\lambda \in \Lambda} p(y_n|\lambda)$$

Note that position p_n plays a role only in tracking the total section duration $D(s_n = F)$.

5.4 Inference

An exact inference of the 'meta-variable' x can be performed by means of the Viterbi algorithm. A key-phrase is detected whenever the most-likely path passes through a section $s_n = Q$ and the last position $D(s_n = Q)$ is reached.

5.5 Paths ranking

Then all detected path segments are ranked according to a weight: a weight measures the acoustic matching between the segment and the query. A weight for a segment with length L is computed from the average likelihood:

#section queries Q	32
average cardinality $\bar{C_q}$	3.2
#words per section	5-14
#sections per recording	6-16
#phonemes per section	26-63

Table 1. Statistics about queries (lyrics sections with unique lyrics) in the test dataset

$$w = (\sum_{l=1}^{L} p(p_l))/L \qquad (9)$$

where $p(p_l)$ is the observation likelihood of an entry p_l from the detected path.

6. DATASET

The **anonymous** test dataset consists of 12 a-cappella performances of 9 compositions with total duration of 18:40 minutes, drawn from classical Turkish Makam repertoire. It has been sung by semiprofessional singers and recorded especially for this study. Scores are provided in the machine-readable *symbTr* format [5], which contain marks of section divisions. A performance has been recorded in-sync with the original recording, whereby instrumental sections are left as silence. This assures that the sequence in which sections are performed is kept the same as in the original performance.

We consider as a query q each section from the scores, which has unique lyrics. In a given recording the boundary timestamps of an occurrence of each query have been annotated. Let C_q be the total number of relevant occurrences (cardinality) of a query q. Table 1 presents the average cardinality $\bar{C_q}$ and other relevant statistics about sections. The low values of $\bar{C_q}$ is mainly due to the small number of performances per composition.

7. EVALUATION

7.1 Evaluation metrics

Having a ranked list of occurrences of each lyrical query, the search-by-lyrics can be interpreted as a ranked retrieval problem, in which the users are interested in checking only the top k relevant results [Schuetze]. A common strategy for rejecting irrelevant results is the retrieval of top K occurrences: As relevant are considered only documents above a rank K [Schuetze]. This allows selecting the K with best score, depending on the

A query in our dataset has low average cardinality ($\bar{C_q} = 3.2$). A suitable performance measure when cardinality is low, is the mean average precision (MAP). Following [Sertan] notation, let the relevance of ranked results for a query q be $[r_q(1), ..., r_q(n_q)]$ where n_q is the number of retrieved occurrences and $r_q(k) \in \{0, 1\}$.

Then the precision and the average precision of a query at rank index K are respectively:

$$P_q(K) = \tfrac{1}{K} \textstyle\sum_{k=1}^{K} r_q(k), \quad \bar{P}_q = \tfrac{1}{C_q} \textstyle\sum_{k=1}^{n_q} r_q(k) P_q(k)$$

After retrieving $\bar{P}_q$ for each $q \in Q$, the MAP over all section queries is defined as

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} \bar{P}_q$$

7.2 Experiments

To assess the benefit of the proposed modeling of positions, we conduct a comparison of the performance of the complete system and a baseline version without the position-HHMM. For it, as result set we consider the audio segments corresponding to the list of candidate paths S derived after subsequence-DTW (see section 4.1). Note that DTW-paths were sorted according to their total cost, which represents observation probability. Thus, this ranking is comparable to the ranking procedure, applied after position-HHMM (5.5). We report results at different values for K in table 2.

Unfortunately, no direct comparison to results of [6] or [Fuji] is possible because these works rely on speech models for languages different than ours. Furthermore, in [Fuji] the accuracy of the key-phrase spotting module is not reported, but instead only the percentage of the links correctly connecting key-phrases form a song to another song. On creating a link for a given key-phrase only the candidate section with highest score for a song has been considered, which might ignore any other true positives. In [6] a result is considered true positive if a keyword is detected within an expected audio clip. The authors argue that since a clip spans one line of lyrics (only 1-10 words) this is sufficiently exact.

7.3 Results

The results confirm the expectation that the performance of subsequence DTW alone is inferior. Retrieving relevant candidate paths in the set S seemed to be very dependent on the weights w_d and w_s for the diagonal and skip steps. We noted that adapting weights for a recording according to detected tempo T might be beneficial, but did not conduct related experiments in this work. The optimal values ($w_d = 6.5$ and $w_s = 11$) in fact guaranteed good coverage of relevant segments in the slowest estimated tempo T.

The MAP for the complete system has a peak when considering first 3 hits from the ranked result list. This is expected because $\bar{C_q} = 3.2$. In general MAPS after decoding of candidate segment with position-modeling, is substantially better than the baseline, which suggests that modeling syllable durations is beneficial. Another reason might be that the position-HHMM models tempo and is thus insucceptible to the difference between the tempo indicated in the score and the real performance tempo.

K	1	2	3	4	6	8
DTW	28.3	23.7	38.3	36.2	37	39
HHMM	48.5	59.3	**64.6**	61	31.3	32.4

Table 2. MAPs (in percent) for ranked result segments for two system variants: baseline with sebseuqnce-DTW and complete with position-HHMM

8. CONCLUSION

In this study we have investigated an important problem that has started to attract attention of researchers only recently. We tackle the linking between audio and structural sections from the perspective of lyrics: a proposed a method for searching in musical audio the occurrences of a characteristic section-long lyrical phrase. We presented a novel HHMM-based model for tracking in a score-informed way sung phonemes and their durations. Evaluation on a-cappella material from Turkish makam music shows that the search with HHMM brings substantial improvement compared to a baseline system, unaware of syllable-duration information.

We plan to focus future work on applying the proposed model to the case of polyphonic singing. We expect further, that this work this can serve as a baseline for further research on singing material with similar musical characteristics.

We want to point as well that, the proposed score-informed scheme is applicable not necessarily only when musical scores are available. Scores can be replaced by any format, from which duration information can be inferred: for example annotated melodic contour or singer-created indications along the lyrics.

9. REFERENCES

[1] On the use of lyrical duration from musical score for automatic lyrics-to-audio alignment.

[2] Christian Dittmar, Pedro Mercado, Holger Grossmann, and Estefanıa Cano. Towards lyrics spotting in the syncglobal project. In *Cognitive Information Processing (CIP), 2012 3rd International Workshop on*, pages 1–6. IEEE, 2012.

[3] Hiromasa Fujihara, Masataka Goto, and Jun Ogata. Hyperlinking lyrics: A method for creating hyperlinks between phrases in song lyrics. In *ISMIR*, pages 281–286, 2008.

[4] Andre Holzapfel, Umut Simsekli, Sertan Sentürk, and Ali Taylan Cemgil. Section-level modeling of musical audio for linking performances to scores in turkish makam music.

[5] M Kemal Karaosmanoğlu. A turkish makam music symbolic database for music information retrieval: Symbtr. *Proc. Int. Society for Music Information Retrieval (ISMIR)*, 2012.

[6] Anna M Kruspe and IDMT Fraunhofer. Keyword spotting in a-capella singing.

[7] Senturk. Composition identification.