uc3m | Universidad Carlos III de Madrid

MSc in Multimedia and Communications (2013–2014)

Automatic Transcription of Lyrics in Monophonic and Poliphonic Songs

by

Miguel Ángel Fernández Torres

Madrid, September 9th, 2014

Supervisor:
Ascensión Gallardo Antolín

AUTOMATIC TRANSCRIPTION OF LYRICS IN MONOPHONIC AND POLYPHONIC SONGS

Miguel Ángel Fernández-Torres[1], Ascensión Gallardo-Antolín[1]
[1]Department of Signal Theory and Communications, Universidad Carlos III, Leganés (Madrid), Spain

ABSTRACT

The paper proposes the implementation of a system for automatic transcription of lyrics in monophonic and polyphonic songs. The basis of the system is an automatic speech recognizer. Taking into account the differences between singing and spoken voice, acoustic models are adapted to singing voice, using several methods, and Language Models (LM) trained on songs lyrics are built. Moreover, background music is attenuated in polyphonic music using the Robust Principal Component Analysis (RPCA) algorithm, trying to facilitate the recognition task avoiding its effect. The results show that, using as adaptation data the same type of tracks that are transcribed then, both adaptation methods and specific LM for songs improve the performance of the baseline system at phoneme- and word-level. However, the use of RPCA over polyphonic songs introduces distortions in singing voice, and therefore, in general, it is not useful for improving the performance of the whole system.

***Index Terms*—** Automatic lyrics transcription, singing voice separation, RPCA, singing adaptation, MLLR, MAP, n-gram language models.

1. INTRODUCTION

Thanks to the growing amount of music devices and services, people are able to find songs in CDs, internet radios and stores, music streaming services or personal music collections, among others. Due to the great number of songs available, there is need for automatic systems that facilitate processing, searching and organization tasks. Knowing a fragment of the textual lyrics of a song could be helpful to identify it and its author searching in lyrics databases. Moreover, lyrics recognition would allow searching in audio databases, automatically transcribing the lyrics of a song being played.

The objective of this work is to develop a system for automatic transcription of lyrics in songs. Apart from its use as music information retrieval system, other applications for the implementation can be considered: an automatic singing-to-lyrics alignment, which finds the temporal relationship between a music audio and its corresponding text; a tool for musicians that allows recognizing immediately lyrics from their new compositions; and an automatic subtitling generation system for live performances, which also facilitates the simultaneous translation into other languages.

The basis of the system proposed is an automatic speech recognizer. Although singing and spoken voice contain similar kind of semantic information and come from the same production physiology, there are differences between them. Vowels are substantially much longer in singing, and the intelligibility is often lower. In addition, singing adds rhythm to the different parameters of speech, such as pitch, loudness and timbre. The pitch range in a singing sentence is usually higher than in a spoken phrase, and it stays approximately constant during a note. On the other hand, songs lyrics are based on a particular vocabulary and syntax, different from those used in dialogues or news text, among others. Our implementation takes into account these aspects, adapting the models to singing and using language models trained on songs lyrics. Furthermore, instrumental accompaniment is as important as singing voice in polyphonic music, which could make even more difficult the recognition task. As can be seen in [1], where the recognition of speech adding several noises is studied, the recognition rate decreases more than 60 percent in the presence of music at a similar volume. In order to try to avoid this negative effect, a source separation algorithm is used to attenuate the presence of music in polyphonic songs.

The paper is structured as follows. Section 2 provides a general review of the state-of-the-art on the fields of singing voice recognition and separation. Section 3 includes the description of the different elements of the developed system. Section 4 presents the databases and the evaluation measures used, and provides an analysis of the results obtained in the experiments. Finally, Section 5 extracts conclusions of the research done and raises some future lines of work.

2. RELATED WORK

There are a lot of works that deal with songs processing but, to our knowledge, almost none poses the automatic recognition of lyrics problem. The complexity of this issue and the need of finding more accurate methods for previous preprocessing tasks to recognition could be two reasons to have barely thought in studying these kind of systems before. Annamaria Mesaros and Tuomas Virtanen consider in [2] the recognition of lyrics, meaning recognition of the phonemes and words from singing voice, where other instruments are used together with singing. Their experiments show that it is possible to adapt speech recognition techniques to singing, and they use also gender-dependent and singer-specific models. A

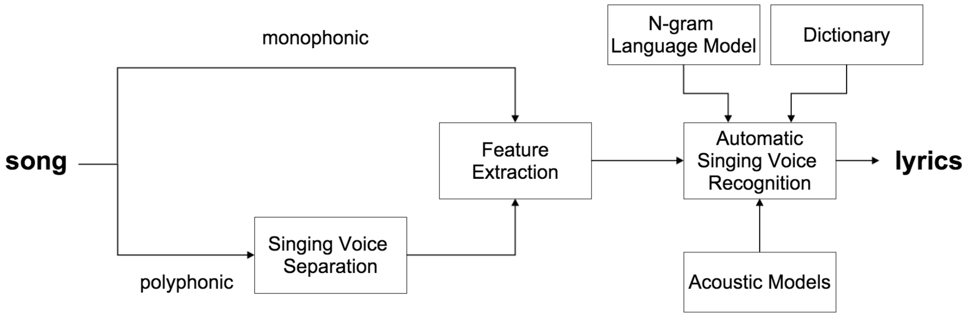

Fig. 1. Schematic diagram of the proposed automatic lyrics transcription system.

singing voice recognition algorithm that is able to automatically recognize a word in a singing signal with background music by using the concept of spectrogram pattern matching is also introduced in [3].

On the other hand, some articles consulted for this work deal with the singing voice separation problem, which consists of extracting singing voice from music. An approximation to this task based on speech/music segmentation for automatic transcription of broadcast news is presented in [4]. In this approach, posterior probability based *entropy* and *dynamism* features are integrated over time through a 2-class HMM with minimum duration constraints. In [5] and [6], the non-negative matrix factorization algorithm is used for robust automatic recognition of mixtures of speech and music and for singing voice separation in mono-channel music, respectively. The method employed by our system is the robust principal component analysis proposed in [7], which is explained in the next section.

3. SYSTEM DESCRIPTION

This section describes the architecture proposed for the automatic recognition of lyrics in music. As was mentioned above, the basis of the developed system is an automatic speech recognizer. In order to consider the particular aspects of singing voice, models are adapted to singing, and the language model is built from a database of song lyrics. The system receives as input a fragment of a song. It should be mentioned that the system is not able to differentiate between vocal and nonvocal parts, so only vocal fragments, previously manually segmented, are processed. If the input is a polyphonic song, singing voice is first separated from the instrumental accompaniment. Then, after extracting several features from the input song, the system tries to recognize automatically its lyrics. The elements that compose the automatic lyrics recognition system implemented, which is schematized in Figure 1, are described below.

3.1. Singing Voice Separation from Polyphonic Music

Instrumental accompaniment is as important as singing voice in most of the existing polyphonic songs, sounding at a similar volume. Its effect results in a background noise for the speech recognizer that is necessary to eliminate. In order to do that, Huang et al. proposed in [7] to use a Robust Principal Component Analysis [8], which is the algorithm we apply in our system.

RPCA is a matrix factorization algorithm for solving underlying low-rank and sparse matrices. It is based on the following convex optimization problem:

$$\begin{aligned}&\text{minimize } ||L||_* + \lambda ||S||_1\\&\text{subject to } L + S = M\end{aligned} \qquad (1)$$

where M, L and S are matrices of dimension $n_1 \times n_2$. M contains the spectrogram of polyphonic songs, and L and S are the low-rank and sparse matrices, respectively. $||\cdot||_*$ denote the nuclear norm (sum of singular values), $||\cdot||_1$ the L1-norm (sum of absolute values of matrix entries) and $\lambda > 0$ is a trade-off parameter between the rank of L and the sparsity of S. If we increase the value of λ, the attenuation of the instrumental accompaniment is higher, but also the singing voice is more distorted.

Music can be considered as a low-rank signal (L matrix), since musical instruments can reproduce the same sounds each time they are played, and music usually has an underlying repeating structure. Singing voice, however, has more variation (higher rank, S matrix) and is relatively sparse in time and frequency domains. The separation is performed as follows. First, M is computed, calculated from the Short-Time-Fourier Transform (STFT). Then, the inexact Augmented Lagrange Multiplier (ALM) method is employed to solve the RPCA problem, obtaining the output matrices L and S. Given these, binary time-frequency masking methods are applied for better separation results. Binary time frequency masking M_b is defined as follows:

$$M_b(m,n) = \begin{cases} 1 & |S(m,n)| > \text{gain} \times |L(m,n)| \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

for all $m = 1...n_1$ and $n = 1...n_2$. Applying this mask to the original STFT matrix M, $X_{singing}$ and X_{music} separation matrices for singing voice and instrumental accompaniment, respectively, are:

$$\begin{aligned} X_{singing}(m,n) &= M_b(m,n)M(m,n) \\ X_{music}(m,n) &= (1 - M_b(m,n))M(m,n) \end{aligned} \tag{3}$$

for all $m = 1...n_1$ and $n = 1...n_2$. To obtain waveforms of the estimated components, the phase of the original signal $P = phase(M)$ is recorded and appended to matrices $X_{singing}$ and X_{music} by $X_{singing}(m,n) = X_{singing}e^{jP(m,n)}$, $X_{music}(m,n) = X_{music}e^{jP(m,n)}$, for $m = 1...n_1$ and $n = 1...n_2$, and the inverse STFT (ISTFT) is calculated.

3.2. Automatic Speech Recognition System

The automatic speech recognizer that serves as the basis of our system is a phonetic Hidden Markov Model (HMM) recognizer, implemented using the Hidden Markov Model Toolkit (HTK) [9]. A HMM is composed by one or several states, and transition probabilities between these states. In our case, the emission probability density function of each state is modeled by a Gaussian Mixture Model (GMM).

The starting point of the speech recognizer is a set of identical single-Gaussian monophone HMMs, with the same mean and variance. The transition matrix and the means and variances of the Gaussian components in each state are estimated in the training stage, in order to maximize the likelihood of the observation vectors from the training data. Short-pause models are added, and the silence model is extended slightly. The monophones are then retrained using the Baum-Welch algorithm and, once reasonable monophone HMMs have been obtained, a forced alignment of the training data is done, together with a final re-estimation of the monophone HMMs. Finally, context-dependent triphone HMMs are made from monophones in two steps. First, monophone transcriptions are converted into triphone transcriptions and a set of triphone models is obtained by copying the monophones and re-estimating. Then, similar acoustic states of these triphones are tied to ensure that all state distributions can be robustly estimated.

The recognition system consists of 39 monophone HMMs plus silence and short-pause models, which produce 65.561 triphones. A three state left-to-right HMM is generated for each phone and the silence model, and the short pause is represented by a one state HMM tied to the middle state of the silence model. As features we use 12 mel-frequency cepstral coefficients (MFCCs) plus energy, delta and acceleration coefficients, calculated in 25 ms frames with a 10 ms hop between adjacent frames.

3.3. N-Gram Language Models

The language model consists of a vocabulary and a set of rules describing how the units in the vocabulary (phonemes, syllables, letters or words) can be connected into sequences. Through the use of language models, the linguistic information in speech or singing can be modeled. An n-gram is a sequence of n symbols, and an n-gram language model (LM) [9, 10] is used to predict each symbol in the sequence given its $n-1$ predecessors. Bigrams and trigrams, which are n-grams of size two and three units, respectively, are commonly used in automatic speech recognition.

Language models estimate the probability of a word sequence, which can be decomposed as a product of conditional probabilities over all i units in the sequence:

$$\hat{P}(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} \hat{P}(w_i | w_1, ..., w_{i-1}) \tag{4}$$

N-gram construction is a three stage process. First, training text is scanned, and its n-grams are counted and stored. Secondly, some words may be mapped to an out of vocabulary class or other class mapping may be applied. Finally, in the final stage, the counts in the resulting *gram* are used to compute n-gram probabilities. The use of an essentially static and finite training text makes difficult to generate a LM which is always well-matched, independent of the recognition task. Moreover, the vocabulary of an n-gram LM is finite and fixed at construction time. For example, if the LM is word-based, new words cannot be added without rebuilding the LM.

It is not possible having a language model with all possible words, so the percentage of out of vocabulary (OOV) words affect the performance of the language model. Although it seems that the vocabulary of the recognizer should be as large as possible to avoid this problem, increasing the vocabulary size increases also the acoustic confusions and not always improve the results. The "goodness" of a language model can be evaluated computing its *perplexity*, which measures how well the LM is able to represent the text to recognize. A good LM should have a small perplexity and a small OOV rate. The influence of the language model in the system can be controlled by the *grammar factor*, and the number of words output by the recognizer is also managed by the *word insertion penalty*. The values of these parameters are determined through a cross validation procedure.

3.4. Adaptation to Singing

It is difficult to find a large songs database to train the recognizer, so acoustic models are trained for speech first, and then a supervised linear adaptation to singing is applied, using a small amount of audio tracks. The adaptation is done offline by finding a set of transforms that reduce the mismatch between the current model set and the adaptation data. HTK [9] performs the adaptation considering maximum likelihood linear transformations such as MLLR and CMLLR, and maximum a-posteriori (MAP) techniques.

3.4.1. Maximum Likelihood Linear Regression (MLLR)

Maximum Likelihood Linear Regression (MLLR) [9, 11] estimates a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system so that each state is more likely to generate the adaptation data. The transformation matrix used to give a new estimation of the adapted mean ξ is given by

$$\hat{\mu} = W\xi \tag{5}$$

where W is the $n \times (n+1)$ transformation matrix, being n the dimensionality of the data, and ξ is the extended mean vector $\xi = [w\ \mu_1\ \mu_2\ ...\ \mu_n]^T$, where w represents a bias offset fixed at 1. Hence W can be decomposed into

$$W = [b\ A] \tag{6}$$

where A represents the $n \times n$ transformation matrix and b is a bias vector.

In Constrained MLLR (CMLLR) [9, 11], the original feature vectors are shifted so that each state of the initial acoustic models is more likely to generate the transformed adaptation data. Now, the transformation matrix is given by

$$\hat{o} = W\zeta \tag{7}$$

where W is the $n \times (n+1)$ transformation matrix, being n the dimensionality of the data, and ζ is the extended observation vector $\zeta = [w\ o_1\ o_2\ ...\ o_n]^T$, where w represents a bias offset fixed at 1. W can be decomposed as same as in MLLR.

In both cases, singing adaptation involves two passes. On the first one, a global adaptation is performed. Then, the second pass uses the global transformation as input to transform the model set, providing better state alignments which are then employed to obtain a set of more specific transformations, using a regression class tree.

3.4.2. Maximum A-Posteriori (MAP)

Maximum A-Posteriori (MAP) [9, 12] adaptation involves the use of prior knowledge about the model parameter distribution. This type of prior is often called informative prior. If we know what the parameters of the model are likely to be before observing the adaptation data, we might be able to make good use of the limited adaptation data. In this case, spoken voice model parameters are used as the informative priors.

The adaptation for the state j and mixture component m of a HMM is given by

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm}+\tau}\bar{\mu}_{jm} + \frac{\tau}{N_{jm}+\tau}\mu_{jm} \tag{8}$$

where τ is a weighting of the a priori knowledge to the adaptation data; N_{jm} is the occupation likelihood of the adaptation data; μ_{jm} is each single mean component in the system; and $\bar{\mu}_{jm}$ is the mean of the observed adaptation data. MAP adaptation requires more adaptation data to be effective when compared to MLLR, because MAP is specifically defined at mixture level.

4. EXPERIMENTS

The experiments carried out using the system described above are explained in this section. First of all, the databases employed and the evaluation measures considered are presented. Then, the recognition results obtained with different language models and singing adaptation methods are provided, discussing the most relevant ones.

4.1. Databases

Several databases are necessary to train and evaluate the system. First of all, the Wall Street Journal database (WSJ0) [13] is used to train the acoustic models of the baseline speech recognition system. Moreover, a LM trained on news text from this database is used. Its vocabulary consists of over 5.000 words.

On the other hand, a LM for songs is obtained using the lyrics of more than 2.500 english songs, downloaded from `http://www.cancionario.net/`. The vocabulary of this LM contains over 20.000 unique words. Due to its large size, a reduced version of this is obtained, keeping the words that appear five or more times in the complete lyrics set. The reduced set includes over 4.500 words. For all the LMs employed, the phonetic transcription of words is extracted from the CMU pronouncing dictionary [14].

Finally, two different databases are used for testing and adapting the models to singing, denoted as *clean* and *poly*. The first set, provided by Annamaria Mesaros [2], contains monophonic singing recordings of 39 fragments of popular songs, and the second one contains 157 fragments manually obtained after segmenting 25 polyphonic songs that belong to five different genres: *blues, country, jazz, pop* and *rock*. Both databases are divided in two sets: *clean_adapt* and *poly_adapt* for adaptation, which suppose the 60 % of each database, approximately; and *clean_test* and *poly_test* for testing, which include the remaining 40 %. The lengths of the fragments are between 20 and 30 seconds, and the division into subsets is done so that the same song appears in the adaptation or in the test set, not in both. The lyrics of both databases are annotated for the supervised adaptation and the evaluation of the system.

LM: WSJ0 (5K)	clean_test	poly_test	poly_clean_test
Phoneme-level	52.18	37.99	38.04
Word-level	9.30	7.72	7.85

Table 1. Recognition rates (%) obtained for each of the test sets using the baseline automatic speech recognition system with a LM trained on news text from WSJ0 database [13].

4.2. Recognition Results and Discussion

In order to evaluate the improvements provided by the different elements of the system proposed, several experiments are done using as test data the subsets presented above (*clean_test*, *poly_test*) and a clean version of the polyphonic one (*poly_clean_test*), where the instrumental accompaniment has been attenuated using RPCA with the default parameters $\lambda = 1$ and *gain* $= 1$. First, the baseline speech recognizer is tested, using a bigram LM trained on news text from WSJ0 database [13]. Then, this LM is replaced by another bigram one trained on song lyrics. Finally, the recognizer is adapted to singing, considering different supervised techniques. After performing a cross validation procedure, the values chosen for the *grammar factor* and the *word insertion penalty* are $s = 9$ and $p = 4$, respectively. These values remain unchanged in all the experiments.

The recognition performance is evaluated taking into account the correct recognition rate at phoneme- and word-level. This measure depends of the number of substitution (S) and deletion (D) errors with respect to the total number of tested instances N, and is given as

$$\text{correct}\ (\%) = \frac{N - D - S}{N} \times 100. \qquad (9)$$

Furthermore, the effect of applying some of the singing adaptation methods tested is studied calculating the *Relative Error Reduction (RER)* with respect to the baseline system, which is not adapted. Mathematically, *RER* is defined as

$$\text{RER}\ (\%) = \frac{adapted\ (\%) - baseline\ (\%)}{100 - baseline\ (\%)} \times 100, \qquad (10)$$

being *baseline* and *adapted* the correct recognition rates obtained for the baseline and the singing-adapted systems, respectively.

4.2.1. Baseline Speech Recognition System

The results obtained with the baseline automatic speech recognition system and the bigram LM trained on news text composed of over 5.000 words (WSJ0 (5K)) are presented in Table 1. Although the recognition rates are low, due to the complexity of the task proposed, we can achieve acceptable results at phoneme-level. As can be seen, the transcription is better in the case of monophonic music (*clean_test*), thanks to the absence of instrumental accompaniment. However, if RPCA is used to attenuate this in polyphonic music, the recognition rate barely increases, due to the distortion that RPCA introduces over the separated singing voice, which also makes difficult the task to the system.

LM	Vocabulary Size	clean_test	poly_test
WSJ0 (5K)	5194	329.8	415.6
CLEAN	187	4.5	–
POLY	863	–	36.1
SONG (4.5K)	4462	216.1	235.6

Table 2. Perplexity and vocabulary size of bigram word-level language models used in the experiments.

LM		clean_test	poly_test	poly_clean_test
CLEAN	Phoneme-level	56.91	–	–
	Word-level	22.59	–	–
POLY	Phoneme-level	–	39.98	40.55
	Word-level	–	12.16	11.86
SONG (4.5K)	Phoneme-level	53.11	38.32	39.12
	Word-level	10.96	9.81	9.09

Table 3. Recognition rates (%) obtained for each of the test sets using the baseline automatic speech recognition system with LMs trained on the words to transcript (CLEAN, POLY) and songs lyrics (SONG (4.5K)).

4.2.2. Language Model Adapted to Songs

As outlined before, specific vocabulary and syntax are employed to write the lyrics of songs, and it is important to consider them in our system. To do this, the bigram LM trained on news text taken first is substituted in this set of experiments by three bigram language models trained on songs lyrics. One of these is the LM based on songs described above, which includes over 4.500 words (SONG (4.5K)), and the other two contain only the words of the songs sets used for testing (CLEAN, POLY).

Table 2 shows the vocabulary size and the perplexities of the four language models used. The better a language model is, the lower its test-set perplexity. As expected, CLEAN and POLY language models present the lowest perplexities for lyrics texts from *clean_test* and *poly_test* sets used for testing, respectively. Moreover, the perplexity of the SONG language model is lower than that of the WSJ0 one, being this more suitable for the transcription of lyrics in songs.

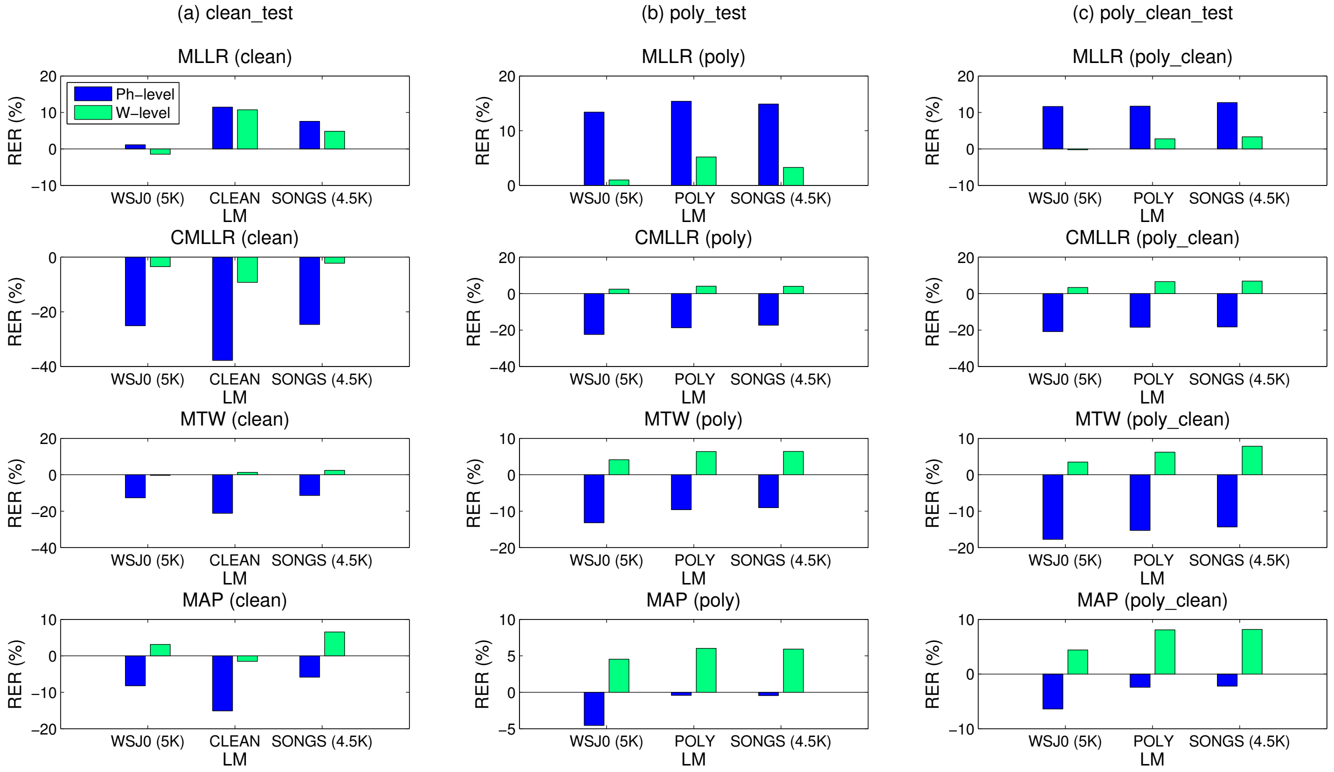

Fig. 2. Relative error reduction (RER (%)) at phoneme- and word-level with respect to the baseline system (not adapted) for the different adaptation methods used (MLLR, CMLLR, MTW and MAP), considering for adaptation and testing the same type of tracks (clean, poly, poly_clean).

The recognition rates obtained for the three LMs adapted to songs are presented in Table 3. Comparing these with those of the baseline system included in Table 1, it can be stated that the use of a language model adapted to the context in which recognition is performed improves the results at phoneme- and word-level. If we use language models trained on the songs lyrics to transcript (CLEAN, POLY), the improvement is significant, especially in the case of monophonic music. Although it is not possible to obtain these LMs in a real scenario, we might be able to increase the recognition rates extracting and processing all the information possible from each song to transcribe, trying to limit the vocabulary size and building the most precise LM. Furthermore, the LM trained on songs lyrics (SONG (4.5K)) allows to achieve better rates than the one trained on news text (WSJ0 (5K)). It can be appreciated that the fact of attenuating the instrumental accompaniment in polyphonic music does not facilitate the transcription, being the results in this case even worse. In the following experiments, models are adapted to singing voice, attempting again to enhance the recognition.

4.2.3. Adaptation to Singing

Four different methods are tested for adapting the models to singing voice: MLLR, CMLLR and MAP, which were explained above, and a version of MLLR where the variances are not updated called MTW. MLLR, CMLLR and MTW carry out a single global transform in the first pass of the adaptation, and 8 transformations in the second pass. On the other hand, the value of τ is fixed to 0.001 in the case of MAP adaptation, given more importance to the mean of the observed adaptation data with respect to the mean of the original models. As adaptation data we use the two subsets presented before (*clean_adapt*, *poly_adapt*), and a clean version of the polyphonic one (*poly_clean_adapt*), where the instrumental accompaniment has been attenuated by applying RPCA with the default parameters. Three adaptations are therefore performed with each of the methods (*clean*, *poly*, *poly_clean*). In total, twelve different adaptations are tested for each of the LM considered before.

All the results obtained for each of the test subsets are presented in the Tables 6 and 7 of the Appendix. From the evaluation of these results it can be seen that the best performance is achieved if the models are adapted with the same type of tracks that are transcribed then. Even in the case of clean polyphonic music it is better to use as adaptation data clean polyphonic tracks than monophonic ones. In order to facilitate the discussion of the results depending of the adaptation method applied, the relative error reduction (RER) at phoneme- and word-level with respect to the not-adapted ba-

Test Set	LM	Adaptation	Correct Transcription	Recognized
clean_test	**WSJ0 (5K)**	**MLLR (clean)**	I'm going deeper underground	On telling to turn on the ground
		MAP (clean)		I am die a wing Peter and out
	CLEAN	**MLLR (clean)**		I am killing day they're underground
		MAP (clean)		I going I town now
	SONG (4.5K)	**MLLR (clean)**		I'd darling a turn on the ground
		MAP (clean)		I are darling Pete around
	WSJ0 (5K)	**MLLR (clean)**	Things that I will go through	Things debt Iowa to to through
		MAP (clean)		Things debt I would
	CLEAN	**MLLR (clean)**		Things that I away a go through
		MAP (clean)		Things at I way road all to
	SONG (4.5K)	**MLLR (clean)**		Things dead I away let go through
		MAP (clean)		Things to that I would all to
poly_test	**WSJ0 (5K)**	**MLLR (poly)**	I walk this empty street	And out hope to examine the state
		MTW (poly)		The like this M. E. street
		MAP (poly)		On a study and to I.
	POLY	**MLLR (poly)**		And out rock to Jackson east
		MTW (poly)		Oh walk this empty street
		MAP (poly)		Oh a and He
	SONG (4.5K)	**MLLR (poly)**		And out hot legs
		MTW (poly)		A walk this empty street
		MAP (poly)		Ah ah and He's
	WSJ0 (5K)	**MLLR (poly)**	My shallow heart's the only thing that's beating	Company will come on the space on
		MTW (poly)		The auto parts the only thing that's betting
		MAP (poly)		Hit us into out heads in the audit and
	POLY	**MLLR (poly)**		With a lot heart's the at all drifted hey that's bit
		MTW (poly)		My shallow heart's the only thing that's beating
		MAP (poly)		Sat how all heart It's a it on
	SONG (4.5K)	**MLLR (poly)**		You'll come on the dark space and at
		MTW (poly)		Sky whoa whoa hearts the only thing that's beating
		MAP (poly)		The instant whoa whoa whoa whoa It's a it on
poly_clean_test	**WSJ0 (5K)**	**MLLR (poly_clean)**	I walk this empty street	A lot is and its stated another big
		MTW (poly_clean)		A low this M. P. street
		MAP (poly_clean)		Along this attempt E. street
	POLY	**MLLR (poly_clean)**		A lot is and you stay to me
		MTW (poly_clean)		I walk this empty street
		MAP (poly_clean)		I walk this empty street
	SONG (4.5K)	**MLLR (poly_clean)**		A rock it isn't he stated
		MTW (poly_clean)		I walk this empty street
		MAP (poly_clean)		I walk this empty street
	WSJ0 (5K)	**MLLR (poly_clean)**	My shallow heart's the only thing that's beating	You could match its top onto parts making a look at I.
		MTW (poly_clean)		The side meeting on parts the only that an odd thing that's meeting
		MAP (poly_clean)		Might sell out parts the only that can own thing that's meeting
	POLY	**MLLR (poly_clean)**		To tide goes out heart's the at on the check at I.
		MTW (poly_clean)		My shallow heart's the only that and thing that's beating
		MAP (poly_clean)		A shallow heart's the only that and thing that's beating
	SONG (4.5K)	**MLLR (poly_clean)**		Too much to hide all our hearts they cut a look at a kid
		MTW (poly_clean)		Take you so I shall our hearts the only that and thing that's beating
		MAP (poly_clean)		Tonight I shall our hearts the only that and thing that's beating

Table 4. Examples of recognized fragments in monophonic and polyphonic songs, using different language models and singing adaptation methods.

seline system is shown in Figure 2 for the adaptations tested of each method that provide the best recognition rates in each test subset.

As can be seen in Figure 2, while MLLR is the singing adaptation method with the highest RER at phoneme-level for the three test subsets, the use of MAP technique provides a significant improvement in the recognition at word-level, which is the main objective of the system. MTW method works even better than MAP when we try to recognize lyrics in polyphonic songs without attenuating the instrumental accompaniment. However, not all the methods tested outperform the baseline system. CMLLR, for example, decreases the correct recognition rates at phoneme-level, and barely increases these at word-level. The recognition results are better again in the case of monophonic songs at phoneme-level, but similar at word-level to those achieved if the transcription of lyrics is carried out in polyphonic songs.

It should be remarked in the case of polyphonic music that RER barely decreases if the LMs trained only on the test songs lyrics (CLEAN, POLY) are substituted by the general LM trained on songs lyrics (SONG (4.5K)). Moreover, RER is higher if this LM is used instead of the LM trained on news texts (WSJ0 (5K)). This shows once again the importance of using a language model adapted to songs. Considering this LM and employing the MAP method, a recognition rate around $15 - 16\,\%$ at word-level for monophonic and polyphonic songs is achieved. The fact of determining a closed vocabulary that contains mainly the words to recognize would facilitate the automatic transcription of lyrics. In that case, rates could be increased to approximately $31\,\%$ if CLEAN is used as LM and MLLR as adaptation method in monophonic songs, and to $17 - 19\,\%$ if POLY is used as LM and MAP as adaptation technique in polyphonic songs.

It has been observed in some experiments with polyphonic tracks that the system tends to "hum" the song and repeat words, due to the presence of background music. When using RPCA to attenuate the music, this effect is sometimes prevented. However, the relative error reduction barely increases at word-level, being even lower at phoneme-level, due to the distortions that the algorithm introduces over the singing voice when it is separated from the instrumental part, which generate the insertion of wrong words in the transcription. It can be concluded, therefore, that, in general, (partly) removing the background music does not improve significantly the recognition rates.

Table 4 includes some examples of recognized fragments in monophonic and polyphonic songs, using the different LM and MLLR, MAP and MTW techniques, and Table 5 summarizes the best recognition rates obtained for each of the test sets at phoneme- and word-level. As commented above, they show that MAP and MTW methods provide the best transcriptions in *clean_test* and *poly_test* sets, respectively. Both MAP and MTW works well in *poly_clean_test* set, and it can be appreciated, comparing its examples with their corresponding ones in *poly_test*, that the attenuation of the instrumental part allows to recognize better some words, but introduces new ones due to the noise and distortion added to tracks, not being possible to improve the overall recognition rate.

Set	**LM**	**Best Adaptation Method**		**correct** (%)
clean_test	**WSJ0 (5K)**	**Phoneme-level**	**MLLR (clean)**	52.72
		Word-level	**MAP (clean)**	12.13
	CLEAN	**Phoneme-level**	**MLLR (clean)**	61.85
		Word-level	**MLLR (clean)**	30.90
	SONG (4.5K)	**Phoneme-level**	**MLLR (clean)**	56.65
		Word-level	**MAP (clean)**	16.78
poly_test	**WSJ0 (5K)**	**Phoneme-level**	**MLLR (poly)**	46.30
		Word-level	**MAP (poly)**	11.90
	POLY	**Phoneme-level**	**MLLR (poly)**	49.21
		Word-level	**MTW (poly)**	17.75
	SONG (4.5K)	**Phoneme-level**	**MLLR (poly)**	47.49
		Word-level	**MTW (poly)**	15.57
poly_clean_test	**WSJ0 (5K)**	**Phoneme-level**	**MLLR (poly_clean)**	45.25
		Word-level	**MAP (poly_clean)**	11.90
	POLY	**Phoneme-level**	**MLLR (poly_clean)**	47.52
		Word-level	**MAP (poly_clean)**	18.98
	SONG (4.5K)	**Phoneme-level**	**MLLR (poly_clean)**	46.87
		Word-level	**MAP (poly_clean)**	16.51

Table 5. Best recognition rates (%) obtained for each of the test sets using the automatic speech recognition system adapted to singing and the different language models.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a system for automatic transcription of lyrics in songs. Despite the complexity of the task, it is possible to obtain acceptable results starting from an automatic speech recognizer, at least at phoneme-level. Furthermore, it can be concluded that, due to the differences between singing and spoken voice, it is necessary to adapt the acoustic models to singing voice and use a language model trained on songs lyrics, in order to improve the performance at word-level. It has also been shown that models have to be adapted with the same type of tracks that are transcribed (monophonic, polyphonic or clean polyphonic) to obtain the best recognition rates. The more the system can be adapted to the song that is going to be transcribed, the better the recognition. This would require extracting and processing all the information possible from the song, attempting to limit the vocabulary size and determining the most precise language model.

With respect to the different adaptation methods evaluated, while MLLR provides the highest RER at phoneme-level, MAP and MTW achieve the best results at word-level in monophonic and polyphonic songs, respectively. Finally, with respect to the type of song to be transcribed, the recognition of lyrics in monophonic songs is more effective, due to the absence of background music. However, if we attenuate the instrumental part in polyphonic music with RPCA, the results barely improve, because this algorithm introduces distortions over the separated singing voice that generate the insertion of wrong words in the transcription.

Among the future work lines devised, three can be highlighted. First, since it has seen that models have to be adjusted to the input track to transcript as much as possible, it is intended to detect first the genre of the song, adapting those only with tracks of the same genre, and generating LM based on lyrics belonging to this genre. Second, a text processing stage after the transcription task is proposed, to avoid humming as well as the repetition of words, which is frequent in the case of polyphonic songs. Finally, it has been observed that the transcription of lyrics is better in those tracks where the singer's voice pitch does not change so much. If we were able to achieve this always in a prior voice equalization stage, maybe the recognition could be enhanced.

6. REFERENCES

[1] Chanwoo Kim and Richard M. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for robust speech recognition," in *ICASSP*. 2012, pp. 4101–4104, IEEE.

[2] Annamaria Mesaros and Tuomas Virtanen, "Automatic recognition of lyrics in singing," *EURASIP J. Audio Speech Music Process.*, vol. 2010, pp. 4:1–4:7, Jan. 2010.

[3] Peerapol Khunarsal, Chidchanok Lursinsap, and Thanapant Raicharoen, "Singing voice recognition based on matching of spectrogram pattern.," in *IJCNN*. 2009, pp. 1595–1599, IEEE.

[4] Jitendra Ajmera, Iain McCowan, and Hervé Bourlard, "Robust HMM-based speech/music segmentation.," in *ICASSP*. 2002, pp. 297–300, IEEE.

[5] Bhiksha Raj, Tuomas Virtanen, Sourish Chaudhuri, and Rita Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition.," in *INTERSPEECH*, Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, Eds. 2010, pp. 717–720, ISCA.

[6] A. Chanrungutai and C.A. Ratanamahatana, "Singing voice separation in mono-channel music," in *Communications and Information Technologies, 2008. ISCIT 2008. International Symposium on*, 2008, pp. 256–261.

[7] Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis.," in *ICASSP*. 2012, pp. 57–60, IEEE.

[8] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust Principal Component Analysis?," *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, June 2011.

[9] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*, Cambridge University Engineering Department, Cambridge, UK, 2009.

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA, 2008.

[11] Marc Ferras, Cheung-Chi Leung, Claude Barras, and Jean-Luc Gauvain, "MLLR techniques for speaker recognition," in *Odyssey*. 2008, p. 23, ISCA.

[12] Chin-Hui Lee and J. Gauvain, "Speaker adaptation based on map estimation of hmm parameters," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, April 1993, vol. 2, pp. 558–561 vol.2.

[13] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Continous Speech Recognition (CSR-I) Wall Street Journal (WSJ0) news, complete," 1993.

[14] Carnegie Mellon University, "Carnegie Mellon University (CMU) pronouncing dictionary," Tech. Rep., Carnegie Mellon University, 2008.

APPENDIX: RECOGNITION RATES OBTAINED USING AN AUTOMATIC SPEECH RECOGNITION SYSTEM ADAPTED TO SINGING

LM	Adaptation Method		clean_test	poly_test	poly_clean_test
WSJ0 (5K)	MLLR (clean)	Phoneme-level	52.72	29.69	30.62
		Word-level	7.97	7.25	7.04
	MLLR (poly)	Phoneme-level	44.23	46.30	44.12
		Word-level	4.32	8.62	7.89
	MLLR (poly_clean)	Phoneme-level	45.42	44.91	45.25
		Word-level	5.81	7.89	7.64
	CMLLR (clean)	Phoneme-level	40.18	19.06	18.63
		Word-level	6.15	4.56	3.97
	CMLLR (poly)	Phoneme-level	33.46	24.10	22.41
		Word-level	5.81	9.94	7.38
	CMLLR (poly_clean)	Phoneme-level	33.16	19.20	25.14
		Word-level	6.31	6.10	10.96
	MTW (clean)	Phoneme-level	46.13	14.99	12.49
		Word-level	8.97	3.16	2.22
	MTW (poly)	Phoneme-level	39.46	29.81	20.90
		Word-level	4.98	11.52	5.16
	MTW (poly_clean)	Phoneme-level	39.04	22.49	27.05
		Word-level	6.15	6.83	11.05
	MAP (clean)	Phoneme-level	48.25	13.45	11.89
		Word-level	12.13	2.47	2.18
	MAP (poly)	Phoneme-level	36.71	35.17	28.23
		Word-level	5.15	11.90	7.21
	MAP (poly_clean)	Phoneme-level	37.73	29.56	34.08
		Word-level	6.81	9.13	11.90
SONG (4.5K)	MLLR (clean)	Phoneme-level	56.65	31.31	31.52
		Word-level	15.28	8.70	7.89
	MLLR (poly)	Phoneme-level	47.44	47.49	45.19
		Word-level	10.63	12.76	10.49
	MLLR (poly_clean)	Phoneme-level	47.11	45.87	46.87
		Word-level	10.47	11.95	12.12
	CMLLR (clean)	Phoneme-level	41.57	19.95	19.23
		Word-level	8.97	5.03	5.63
	CMLLR (poly)	Phoneme-level	35.32	27.64	25.69
		Word-level	10.47	13.44	11.01
	CMLLR (poly_clean)	Phoneme-level	34.09	21.29	27.98
		Word-level	10.96	9.64	15.32
	MTW (clean)	Phoneme-level	47.78	16.36	13.91
		Word-level	13.12	4.27	2.99
	MTW (poly)	Phoneme-level	40.81	32.75	24.41
		Word-level	7.31	15.57	9.60
	MTW (poly_clean)	Phoneme-level	40.98	25.14	30.40
		Word-level	8.14	10.54	16.21
	MAP (clean)	Phoneme-level	50.36	14.51	13.19
		Word-level	16.78	5.42	4.39
	MAP (poly)	Phoneme-level	38.28	38.04	30.89
		Word-level	9.30	15.15	10.41
	MAP (poly_clean)	Phoneme-level	38.99	31.52	37.78
		Word-level	8.47	11.09	16.51

Table 6. Recognition rates (%) obtained for each of the test sets using the automatic speech recognition system adapted to singing with several techniques and language models trained on news text from WSJ0 database [13] (WSJ0 (5K)) and song lyrics (SONG (4.5K)).

LM	Adaptation Method		clean_test	poly_test	poly_clean_test
CLEAN	**MLLR (clean)**	**Phoneme-level**	61.85	–	–
		Word-level	30.90	–	–
	MLLR (poly)	**Phoneme-level**	48.71	–	–
		Word-level	20.43	–	–
	MLLR (poly_clean)	**Phoneme-level**	49.43	–	–
		Word-level	19.77	–	–
	CMLLR (clean)	**Phoneme-level**	40.64	–	–
		Word-level	15.45	–	–
	CMLLR (poly)	**Phoneme-level**	32.83	–	–
		Word-level	12.96	–	–
	CMLLR (poly_clean)	**Phoneme-level**	34.69	–	–
		Word-level	14.62	–	–
	MTW (clean)	**Phoneme-level**	50.23	–	–
		Word-level	23.59	–	–
	MTW (poly)	**Phoneme-level**	39.12	–	–
		Word-level	11.96	–	–
	MTW (poly_clean)	**Phoneme-level**	39.80	–	–
		Word-level	12.29	–	–
	MAP (clean)	**Phoneme-level**	50.40	–	–
		Word-level	21.43	–	–
	MAP (poly)	**Phoneme-level**	36.08	–	–
		Word-level	11.79	–	–
	MAP (poly_clean)	**Phoneme-level**	32.78	–	–
		Word-level	12.24	–	–
POLY	**MLLR (clean)**	**Phoneme-level**	–	31.64	32.63
		Word-level	–	12.07	11.18
	MLLR (poly)	**Phoneme-level**	–	49.21	45.95
		Word-level	–	16.72	14.16
	MLLR (poly_clean)	**Phoneme-level**	–	46.48	47.52
		Word-level	–	14.76	14.29
	CMLLR (clean)	**Phoneme-level**	–	19.54	20.05
		Word-level	–	5.97	7.21
	CMLLR (poly)	**Phoneme-level**	–	28.73	27.16
		Word-level	–	15.70	13.99
	CMLLR (poly_clean)	**Phoneme-level**	–	21.55	29.57
		Word-level	–	9.43	17.62
	MTW (clean)	**Phoneme-level**	–	16.55	13.53
		Word-level	–	6.61	4.65
	MTW (poly)	**Phoneme-level**	–	34.21	25.25
		Word-level	–	17.75	11.01
	MTW (poly_clean)	**Phoneme-level**	–	26.42	31.46
		Word-level	–	11.09	17.32
	MAP (clean)	**Phoneme-level**	–	14.23	13.03
		Word-level	–	7.72	6.44
	MAP (poly)	**Phoneme-level**	–	39.73	32.12
		Word-level	–	17.45	11.60
	MAP (poly_clean)	**Phoneme-level**	–	37.56	39.11
		Word-level	–	13.46	18.98

Table 7. Recognition rates (%) obtained for each of the test sets using the automatic speech recognition system adapted to singing with several techniques and language models trained on words to transcript (CLEAN, POLY).