INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Retrieval of textual song lyrics from sung inputs

Anna M. Kruspe

Fraunhofer IDMT, Ilmenau, Germany
kpe@idmt.fraunhofer.de

Abstract

Retrieving the lyrics of a sung recording from a database of text documents is a research topic that has not received attention so far. Such a retrieval system has many practical applications, e.g. for karaoke applications or for indexing large song databases by their lyric content.

In this paper, we present such a lyrics retrieval system. In a first step, phoneme posteriorgrams are extracted from sung recordings using various acoustic models trained on *TIMIT* and a variation thereof, and on subsets of a large database of recordings of unaccompanied singing (*DAMP*). On the other side, we generate binary templates from the available textual lyrics. Since these lyrics do not have any temporal information, we then employ an approach based on Dynamic Time Warping to retrieve the most likely lyrics document for each recording.

The approach is tested on a different subset of the unaccompanied singing database which includes 601 recordings of 301 different songs (12000 lines of lyrics). The approach is evaluated both on a song-wise and on a line-wise scale.

The results are highly encouraging and could be used further to perform automatic lyrics alignment and keyword spotting for large databases of songs.

Index Terms: Lyrics, Text retrieval, Singing, Automatic Speech Recognition, Music Information Retrieval

1. Introduction

Automatic speech recognition on singing has only started to receive attention as a field of research in the past few years [1]. The research so far shows that most tasks are notoriously harder than on speech [2]. The reason for this is a multitude of differences between speech and singing, with most characteristics being much more varied in singing than in speech. Examples include pitch range, phoneme durations, pronunciation variants, semantic content, and many more.

Tasks like phoneme recognition, keyword spotting, or lyrics transcription therefore only achieve relatively low results so far [3][4]. But there is one factor that could be beneficial to all of these tasks: The wide availability of textual lyrics on the internet. In contrast with the mentioned tasks, automatic alignment of lyrics to singing has already produced satisfactory results [5][2]. Therefore, if the lyrics of a song can be found and then aligned, many other applications could profit.

In this paper, we present an approach to the task of automatically retrieving the lyrics for a sung recording from a corpus of known textual lyrics. In order to do this, we first generate phoneme posteriorgrams using various acoustic models, and then perform a search based on Dynamic Time Warping (DTW) to find the most likely lyrics. This approach is tested on a song-wise scale, and on lines of lyrics only.

The paper is structured as follows: In section 2, we briefly sum up the state of the art of related tasks. Our data is described in section 3. In section 4, we present our new approach. Section 5 shows our experiments and results. Finally, we give a conclusion in section 6, and make suggestions for future work in section 7.

2. State of the art

To our knowledge, there is no literature that deals with the task of finding one exact text document from a fixed corpus that corresponds to a spoken input. In a sense, the field of Spoken Document Retrieval can be seen as the opposite of this task (i.e., finding a spoken document corresponding to a text query - although this text query is usually not a transcription of the audio recording) [6]. The field of voice search is also related, although here the result space is much bigger and a more in-depth analysis is necessary to interpret both the query and the possible results [7]. Of course, transcription approaches could be employed to generate a full transcription of a recording and then perform a matching based on the result, but in order to do this, a close-to-perfect transcription would be necessary. This is not yet possible in many scenarios, with singing being one of them [8].

On the side of music specifically, lyrics search has so far only been done on a smaller scale in order to assist other tasks. In [9], an automatic alignment between lyrics and audio is performed, which then later allows searching for certain lyrical phrases in songs. In [10], lyrics information is used to aid in a query-by-singing system. In both cases, the matching textual lyrics are known from the start.

3. Data sets

3.1. Speech data sets

For training our baseline phoneme recognition models, we used the train and test data from *Timit* [11]. Additionally, we trained phoneme models on a modification of *Timit* where pitch-shifting, time-stretching, and vibrato were applied to the audio data. The process is described in [4]. This data set will be referred to as *TimitM*.

3.2. Singing data sets

For training models specific to singing, we used the *DAMP* data set, which is freely available from Stanford University[1][12]. This data set contains more than 34,000 recordings of amateur singing of full songs with no background music, which were obtained from the *Smule Sing!* karaoke app. Each performance is labeled with metadata such as the gender of the singer, the region of origin, the song title, etc. The singers performed 301 English language pop songs. The recordings have good sound quality with (usually) little background noise, but come from a

[1] https://ccrma.stanford.edu/damp/

http://dx.doi.org/10.21437/Interspeech.2016-1272

lot of different recording conditions.

No lyrics annotations are available for this data set, but we obtained the textual lyrics from the *Smule Sing!* website[2]. These were, however, not aligned in any way. We performed such an alignment on the word and phoneme levels automatically (see section 4.1).

Out of all those recordings, we created several different sub-data sets:

DampB Contains 20 full recordings per song (6000 in sum), both male and female.

DampBB Same as before, but phoneme instances were discarded until they were balanced and a maximum of 250,000 frames per phoneme where left, where possible. This data set is about 4% the size of *DampB*.

DampBB_small Same as before, but phoneme instances were discarded until they were balanced and 60,000 frames per phoneme were left (a bit fewer than the amount contained in *Timit*). This data set is about half the size of *DampBB*.

DampFB and DampMB Using 20 full recordings per song and gender (6000 each), these data sets were then reduced in the same way as *DampBB*. *DampFB* is roughly the same size, *DampMB* is a bit smaller because there are fewer male recordings.

DampTestF and DampTestM Contains one full recording per song and gender (300 each). These data sets were used for testing. There is no overlap with any of the training data sets.

Order-13 MFCCs plus deltas and double-deltas were extracted from all data sets and used in all experiments.

4. Proposed approach

The general lyrics retrieval process is shown in figure 1.

4.1. Lyrics alignment

Since the textual lyrics were not aligned to the singing audio data, we first performed a forced alignment step. A monophone HMM acoustic model trained on *Timit* using HTK was used. Alignment was performed on the word and phoneme levels using lyrics and recordings of full songs.

The resulting annotations were used in the following experiments. Of course, errors cannot be avoided when doing automatic forced alignment. Nevertheless, the results appear to be very good overall, and this approach provided us with a large amount of annotated singing data, which could not feasibly have been done manually [13].

4.2. New acoustic models

Using these automatically generated annotations, we then trained new acoustic models on *DampB*, *DampBB*, *DampFB*, and *DampMB*. Models were also trained on *Timit* and *TimitM*.

All models are DNNs with three hidden layers of 1024, 850, and again 1024 dimensions. The output layer corresponds to 37 monophones. Inputs are MFCCs with deltas and double-deltas (39 dimensions).

4.3. Phoneme recognition

Using these models, phoneme posteriorgrams were then generated on the test data sets (*DampTestF* and *DampTestM*).

4.4. Similarity calculation for textual lyrics

In order to find the matching lyrics for each posteriorgram produced in the previous step, we first generated binary templates for all possible song lyrics on the phoneme scale. These can be seen as oracle posteriorgrams, but do not include any timing information.

Between all of these templates and the query posteriogram, similarity matrices were calculated using the cosine distance. On the resulting matrices, Dynamic Time Warping (DTW) was then performed using the implementation from [14]. An example is shown in figure 2. Since we do not know how long each phoneme stretches in the actual recording and the lyrics templates have different lengths, the length of the warping path should not be a detrimental factor in cost calculation. Therefore, the accumulative cost of the best path was divided by the path length and then retained as a score for each possible lyrics document. In the end, the lyrics document with the lowest cost was chosen as a match (or, in some experiments, the N documents with the lowest costs).

Additionally, we split both the textual lyrics corpus and the sung inputs into smaller segments roughly corresponding to one line in the lyrics each (around 12,000 lines). We then repeated the whole process for these inputs. This allowed us to see how well lyrics could be retrieved from just one single sung line of the song. Sub-sequence DTW could also be used for this task instead of splitting both corpora.

Two optimizations were made to the algorithm. The first one was a sub-sampling of the phoneme posteriorgrams by the factor 10 (specifically, we calculated the mean for 10 consecutive frames). This increased the speed of the DTW for the individual comparisons and also produced better results. We also tested longer windows, but this had a negative impact on the result.

Secondly, squaring the posteriorgrams before the similarity calculation produced slightly better results. This makes the posteriorgrams more similar to the binary lyrics templates used for comparison. We also tried binarizing them, but this emphasized phoneme recognition errors too much.

5. Experiments and results

5.1. Lyrics retrieval on whole song inputs

In our first experiment, we calculated similarity measures between the lyrics and recordings of whole songs using the described process. We tested this with phoneme posteriorgrams obtained with all five acoustic models on the female and the male test sets (*DampTestF* and *DampTestM*). We then calculated the accuracy on the 1-, 3-, and 10-best results for each song (i.e., how many lyrics are correctly detected when taking into account the 1, 3, and 10 lowest distances?). The results on the female test set are shown in figure 3a, the ones for the male test set in figure 3b.

These results show that phoneme posteriorgrams obtained with models trained on speech data (*Timit*) generally produce the lowest results in lyrics retrieval. The difference between the two test sets is especially interesting here: On the male test set, the accuracy for the single best result is 58%, while on the female set it is only 39%. Previous experiments showed that the phoneme recognition itself performs somewhat worse for female singing inputs. This effect is compounded in these lyrics retrieval results. We assume that this happens because the frequency range of female singing is even further removed from that of speech than the frequency range of male singing is [15]. Even female speech is often performed at the lower end

[2] http://www.smule.com/songs

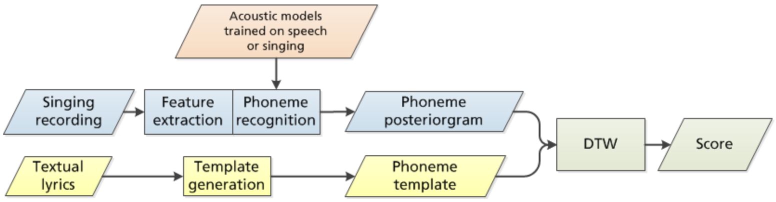
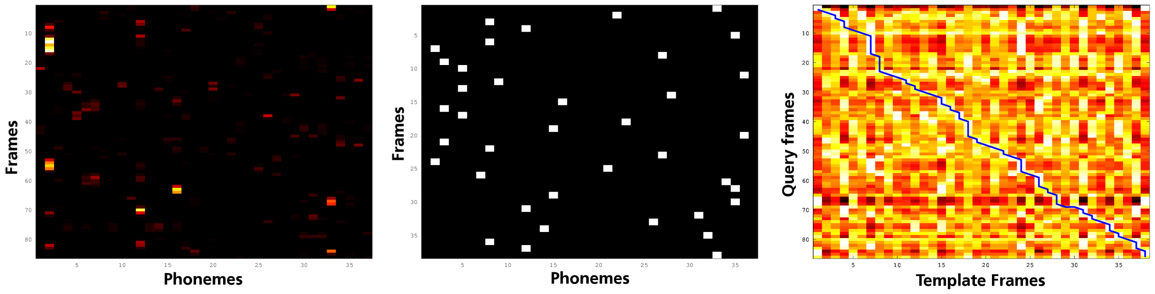


Figure 1: Overview of the lyrics retrieval process.

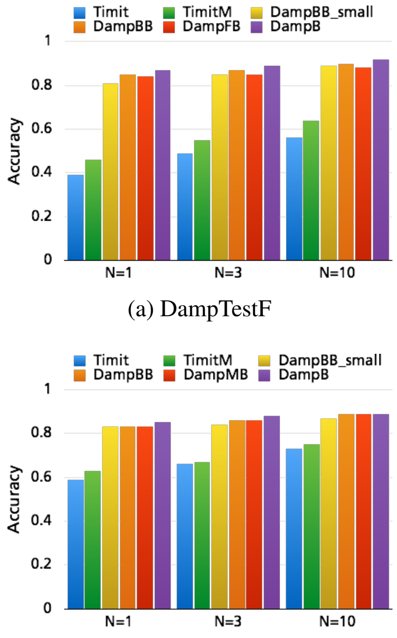


(a) Phoneme posteriorgram

(b) Phoneme template

(c) Similarity matrix with cheapest path (blue)

Figure 2: Example of a similarity calculation: Phoneme posteriorgrams are calculated for the audio recordings **(a)**. Phoneme templates are generated for the textual lyrics **(b)**. Then, a similarity matrix is calculated using the cosine distance between the two, and DTW is performed on it **(c)**. The accumulated cost divided by the path length is the similarity measure.



(a) DampTestF

(b) DampTestM

Figure 3: Accuracies of the results for lyrics detection on the whole song for the *DampTest* sets using five different acoustic models, and evaluated on the 1-, 3-, 10-, 50-, and 100-best results.

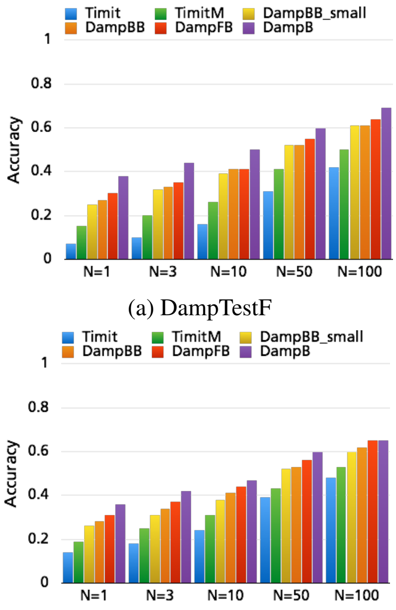
of the female singing frequency range. The frequency range of male singing is better covered when training models on speech recordings (especially when speech recordings of both genders are used).

This effect is still visible for the *TimitM* models, which is a variant of *Timit* that was artificially made more "song-like". However, the pitch range was not expanded too far in order to keep the sound natural.

The results improve massively when acoustic models trained on any of the *Damp* singing corpora are used. The difference between the male and female results disappears, which supports the idea that the female pitch range was not covered well by the models trained on speech. Using the models trained on the smallest singing data set (*DampBB_small*), which is slightly smaller than *Timit*, the results increase to 81% and 83% for the single best result on the female and the male test set respectively. With the models trained on the *DampBB* corpus, which is about twice as big, they increase slightly more to 85% on the female test set. Gender-specific models of the same size do not improve the result in this case.

Finally, the results obtained with the acoustic models trained on the largest singing corpus (*DampB*) provide the very best results at accuracies of 87% and 85%.

For some applications, working with the best N instead of just the very best result could be useful (e.g. for presenting a selection of possible lyrics to a user). When the best 3 results can be taken into account, the accuracies on the best posteriorgrams rise to 89% and 88% on the female and male test sets respectively. When the best 10 results are used, they reach 92% and 89%.



(a) DampTestF

(b) DampTestM

Figure 4: Accuracies of the results for lyrics detection on separate lines of sung lyrics for the *DampTest* sets using five different acoustic models, and evaluated on the 1-, 3-, 10-, 50-, and 100-best results.

5.2. Lyrics retrieval on line-wise inputs

In our second experiment, we performed the same process, but this time only used lines of sung lyrics as inputs (usually a few seconds in duration). Costs were then calculated between the posteriorgrams of these recordings and all 12,000 available lines of lyrics. Lines with fewer than 10 phonemes were not taken into account.

We then evaluated whether a line from the correct song was retrieved in the N-best results. In this way, confusions between repetitions of a line in the same song did not have an impact on the result. However, repetitions of lyrical lines across multiple songs are a possible source of confusion. The results for the female test set are shown in figure 4a, the ones for the male test set in figure 4b.

Again, we see the difference between both test sets when generating posteriograms with the *Timit* models. The accuracy on the best result is 14% for the male test set, but just 7% for the female test.

The results for the *Damp* models show the same basic tendencies as before, although naturally much lower. For the single best result, the accuracies when using the *DampB* model are 38% and 36% on the female and male test sets respectively. For this task, gender-dependent models produce slightly higher results than the mixed-gender ones of the same size.

5.3. Sources of error

To find possible starting points for improving the algorithm, we took a closer look at songs where lyrics could not be retrieved at all across the various acoustic models. Some sources of error repeatedly stuck out:

Unclear enunciation Some singers pronounced words very unclearly, often focusing more on musical performance than on the lyrics.

Accents Some singers sung with an accent, either their natural one or imitating the one used by the original singer of the song.

Young children's voices Some recordings were performed by young children.

Background music Some singers had the original song with the original singing running in the background.

Speaking in breaks Some singers spoke in the musical breaks.

Problems in audio quality Some recordings had qualitative problems, especially loudness clipping.

For most of these issues, more robust phoneme recognizers would be helpful. For others, the algorithm could be adapted to be robust to extraneous recognized phonemes (particularly for the speaking problem). Others may not be solvable with an approach like ours at all. In those cases, a combination with melody recognition could be a solution.

On the other hand, many of these problems would presumably not play a role when using professional recordings.

6. Conclusion

In this paper, we presented an approach to retrieving the matching lyrics for a singing recording from a fixed database of 300 textual lyrics. To do this, we first extract phoneme posteriorgrams from the audio and generate phoneme templates from all possible lyrics. We then perform Dynamic Time Warping on all combinations to obtain distance measures.

When the whole song is used as the input, we obtain an accuracy of 86% for the single best result. If the 10 best results are taken into account, this rises to 91%. When using only short sung lines as input, the mean 1-best accuracy for retrieving the correct song lyrics of the whole song is 37%. For the best 100 results, the accuracy is 67%.

An interesting result was the difference between the female and the male test sets: On the female test set, retrieval with models trained on speech was significantly lower than on the male set (39% vs. 58% on the song-wise task). We believe this happens because the frequency range of female singing is not covered well with speech data only. When using acoustic models trained on singing, this difference disappears and the results become significantly higher in general. Even for a model trained on less data than that contained in *Timit*, the average accuracy is 82%.

When looking at possible sources of error, many of them had to do with enunciation issues (clarity, accents, or children's voices) or issues with the recording itself (background music, clipping, extraneous speaking). These problems would not be as prevalent in professional recordings. However, some of them could be fixed with adaptations to the algorithm.

7. Future work

As mentioned before, we would like to improve our algorithm to be more robust to the detected error sources. Other possible points of improvement include the choice of the distance metric or of the acoustic models. Preliminary tests suggest that combining results of different phoneme recognizers could improve the over-all result.

We have not tested this approach on singing with background music yet, which could be an interesting next step. So far, only a fixed corpus of possible lyrics was taken into account. Opening the approach up to larger databases would make it more flexible. This could be combined with Semantic Web technologies to automatically find lyrics on the internet.

When the space of possible lyrics becomes larger, techniques for scalability will be necessary. One such idea could be a rough search with smaller lyrical "hashes" to find possible matches, and then a refinement with our current approach. This is similar to techniques that are already used in audio fingerprinting [16].

8. References

[1] A. Mesaros and T. Virtanen, "Recognition of phonemes and words in singing." in *ICASSP*. IEEE, 2010, pp. 2146–2149.

[2] H. Fujihara and M. Goto, *Multimodal Music Processing*. Dagstuhl Follow-Ups, 2012, ch. Lyrics-to-audio alignment and its applications.

[3] A. Mesaros and T. Virtanen, "Automatic recognition of lyrics in singing." *EURASIP J. Audio, Speech and Music Processing*, vol. 2010, 2010.

[4] A. M. Kruspe, "Training phoneme models for singing with "songified" speech data," in *15th International Conference on Music Information Retrieval (ISMIR)*, Malaga, Spain, 2015.

[5] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *DaFX-08*, Espoo, Finland, 2008.

[6] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees, "The TREC Spoken Document Retrieval Track: A Success Story," in *in Text Retrieval Conference (TREC) 8*, 2000, pp. 16–19.

[7] Y.-Y. Wang, D. Yu, Y.-C. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Processing Magazine (Special Issue on Spoken Language Technology)*, May 2008.

[8] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH*, 2011.

[9] M. Müller, F. Kurth, D. Damm, C. Fremerey, and M. Clausen, "Lyrics-based audio retrieval and multimodal navigation in music collections," in *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007*, Budapest, Hungary, 2007, pp. 112–123.

[10] C.-C. Wang and J.-S. R. Jang, "Improving query-by-singing/humming by combining melody and lyric information," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 4, pp. 798–806, Apr. 2015.

[11] J. S. Garofolo et al., "TIMIT Acoustic-Phonetic Continuous Speech Corpus," Linguistic Data Consortium, Philadelphia, Tech. Rep., 1993.

[12] J. C. Smith, "Correlation analyses of encoded music performance," Ph.D. dissertation, Stanford University, 2013.

[13] A. M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *17th International Conference on Music Information Retrieval (ISMIR)*, New York, NY, USA, 2016.

[14] D. P. W. Ellis, "Dynamic Time Warp (DTW) in Matlab," 2003, web resource, Last checked: 03/30/16. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/pvoc/

[15] J. Sundberg, *The Psychology of Music*, 3rd ed. Academic Press, 2012, ch. 6. Perception of singing.

[16] A. L. Wang, "An industrial-strength audio search algorithm," in *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, Baltimore, MD, USA, 2003, pp. 7–13.