KEYWORD SPOTTING IN A-CAPELLA SINGING

Anna M. Kruspe
Fraunhofer IDMT, Ilmenau, Germany
Johns Hopkins University, Baltimore, MD, USA
kpe@idmt.fhg.de

ABSTRACT

Keyword spotting (or spoken term detection) is an interesting task in Music Information Retrieval that can be applied to a number of problems. Its purposes include topical search and improvements for genre classification. Keyword spotting is a well-researched task on pure speech, but state-of-the-art approaches cannot be easily transferred to singing because phoneme durations have much higher variations in singing. To our knowledge, no keyword spotting system for singing has been presented yet.

We present a keyword spotting approach based on keyword-filler Hidden Markov Models (HMMs) and test it on a-capella singing and spoken lyrics. We test Mel-Frequency Cepstral Coefficents (MFCCs), Perceptual Linear Predictive Features (PLPs), and Temporal Patterns (TRAPs) as front ends. These features are then used to generate phoneme posteriors using Multilayer Perceptrons (MLPs) trained on speech data. The phoneme posteriors are then used as the system input. Our approach produces useful results on a-capella singing, but depend heavily on the chosen keyword. We show that results can be further improved by training the MLP on a-capella data.

We also test two post-processing methods on our phoneme posteriors before the keyword spotting step. First, we average the posteriors of all three feature sets. Second, we run the three concatenated posteriors through a fusion classifier.

1. INTRODUCTION

Keyword spotting is the task of searching for certain words or phrases (spoken term detection) in acoustic data. In contrast to text data, we cannot directly search for these words, but have to rely on the output of speech recognition systems in some way.

In speech, this problem has been a topic of research since the 1970's [1] and has since seen a lot of development and improvement [11]. For singing, however, we are not aware of any fully functional keyword spotting systems.

Music collections of both professional distributors and private users have grown exponentially since the switch to a digital format. For these large collections, efficient search methods are necessary. Keyword spotting in music collections has beneficial applications for both user groups. Using keyword spotting, users are able to search their collections for songs with lyrics about certain topics. As an example, professional users might use this in the context of synch licensing [4] (e.g., "I need a song containing the word 'freedom' for a car commercial".) Private users could, for example, use keyword spotting for playlist generation ("Generate a playlist with songs that contain the word 'party'.")

In this paper, we present our approach to a keyword spotting system for a-capella singing. We will first look at the current state of the art in section 2. We then present our data set in section 3. In section 4, we describe our own keyword spotting system. A number of experiments on this system and their results are presented in section 5. Finally, we draw conclusions in section 6 and give an outlook on future work in section 7.

2. STATE OF THE ART

2.1 Keyword spotting principles

As described in [13], there are three basic principles that have been developed over the years for keyword spotting in speech:

LVCSR-based keyword spotting For this approach, full Large Vocabulary Continues Speech Recognition (LVCSR) is performed on the utterances. This results in a complete text transcription, which can then be searched for the required keywords. LVCSR-based systems lack tolerance for description errors - i.e., if a keyword is not correctly transcribed from the start, it cannot be found later. Additionally, LVCSR systems are complex and expensive to implement.

Acoustic keyword spotting As in LVCSR-based keyword spotting, acoustic keyword spotting employs Viterbi search to find the requested keyword in a given utterance. In this approach, however, the system does not attempt to transcribe each word, but only searches for the specific keyword. Everything else is treated as "filler". This search can be performed directly on the audio features using an acoustic example, or on phoneme posteriorgrams generated by an acoustic model. In the second case, the algorithm searches for the word's phonemes.

This approach is easy to implement and provides some pronunciation tolerance. Its disadvantage is

the lack of integration of a-priori language knowledge (i.e. knowledge about plausible phoneme and word sequences) that could improve performance.

Phonetic search keyword spotting Phonetic search keyword spotting starts out just like LVCSR-based keyword spotting, but does not generate a word transcription of the utterance. Instead, phoneme lattices are saved. Phonetic search for the keyword is then performed on these lattices. This approach combines the advantages of LVCSR-based keyword spotting (a-priori knowledge in the shape of language models) and acoustic keyword spotting (flexibility and robustness).

2.2 Keyword spotting in singing

The described keyword spotting principles cannot easily be transferred to music. Singing, in contrast to speech, presents a number of additional challenges, such as larger pitch fluctuation, more pronunciation variation, and different vocabulary (which means existing models cannot easily be transferred).

Another big difference is the higher variation of phoneme durations in singing. Both LVCSR-based keyword spotting and Phonetic search keyword spotting depend heavily on predictable phoneme durations (within certain limits). When a certain word is pronounced, its phonemes will usually have approximately the same duration across speakers. The language model employed in both approaches will take this information into account.

We compared phoneme durations in the TIMIT speech database [7] and our own a-capella singing database (see section 3). The average standard deviations for vowels and consonants are shown in figure 1. It becomes clear that the phoneme durations taken from TIMIT do not vary a lot, whereas some the a-capella phonemes show huge variations. It becomes clear that this especially concerns vowels (AA, AW, EH, IY, AE, AH, AO, EY, AY, ER, UW, OW, UH, IH, OY). This observation has a foundation in music theory: Drawn-out notes are usually sung on vowels.

For this reason, acoustic keyword spotting appears to be the most feasible approach to keyword spotting in singing. To our knowledge, no full keyword spotting system for singing has been presented yet. In [2], an approach based on sub-sequence Dynamic Time Warping (DTW) is suggested. This is similar to the acoustic approach, but does not involve a full acoustic model. Instead, example utterances of the keyword are used to find similar sequences in the tested utterance.

In [5], a phoneme recognition system for singing is presented. It extracts Mel-Frequency Cepstral Coefficients (MFCCs) and Temporal Patterns (TRAPs) which are then used as inputs to a Multilayer Perceptron (MLP). The phonetic output of such a system could serve as an input to a keyword spotting system.

There are also some publications where similar principles are applied to lyrics alignment and Query by Humming [12] [3].

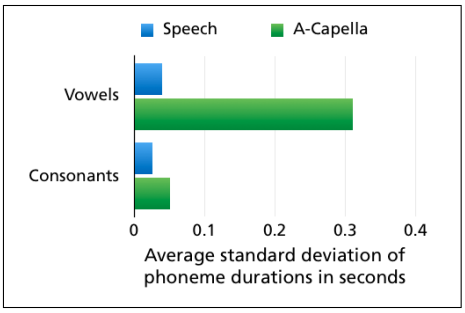

Figure 1: Average standard deviations for vowels and consonants in the TIMIT speech databases (blue) and our a-capella singing data set (green).

3. DATA SET

Our data set is the one presented in [5]. It consists of the vocal tracks of 19 commercial pop songs. They are studio quality with some post-processing applied (EQ, compression, reverb). Some of them contain choir singing. These 19 songs are split up into clips that roughly represent lines in the song lyrics.

Twelve of the songs were annotated with time-aligned phonemes. The phoneme set is the one used in CMU Sphinx [1] and TIMIT [7] and contains 39 phonemes. All of the songs were annotated with word transcriptions. For comparison, recordings of spoken recitations of all song lyrics were also made. These were all performed by the same speaker.

We selected 51 keywords for testing our system. Most of them were among the most frequent words in the provided lyrics. A few were selected because they had a comparatively large number of phonemes. An overview is given in table 1.

4. PROPOSED SYSTEM

Figure 2 presents an overview of our system.

1. Feature extraction We extract Mel-Frequency Cepstral Coefficients (MFCCs), Perceptual Linear Predictive features (PLPs), and Temporal Patterns (TRAPs) [6]. We keep 20 MFCC coefficients and 39 PLP coefficients (13 direct coefficients plus deltas and double-deltas). For the TRAPs, we use 8 linearly spaced spectral bands and a temporal context of 20 frames and keep 8 DCT coefficients.

2. MLP training and phoneme recognition Using each feature data set, we train Multi-Layer Perceptrons (MLPs). MLPs are commonly used to train acoustic models for the purpose of phoneme recognition. We chose a structure with two hidden layers and tested three different dimension settings: 50, 200, and 1000 dimensions per layer. MLPs were trained solely on TIMIT data first, then on a mix of TIMIT and a-capella in a second experiment. The resulting MLPs are then used to recognize phonemes in our a-capella dataset, thus generating phoneme posteriorgrams.

[1] `http://cmusphinx.sourceforge.net/`

Number of Phonemes	Keywords
2	way, eyes
3	love, girl, away, time, over, home, sing, kiss, play, other
4	hello, trick, never, hand, baby, times, under, things, world, think, heart, tears, lights
5	always, inside, drink, nothing, rehab, forever, rolling, feeling, waiting, alright, tonight
6	something, denial, together, morning, friends, leaving, sunrise
7	umbrella, afternoon, stranger, somebody, entertain, everyone
8	beautiful, suicidal

Table 1: All 51 tested keywords, ordered by number of phonemes.

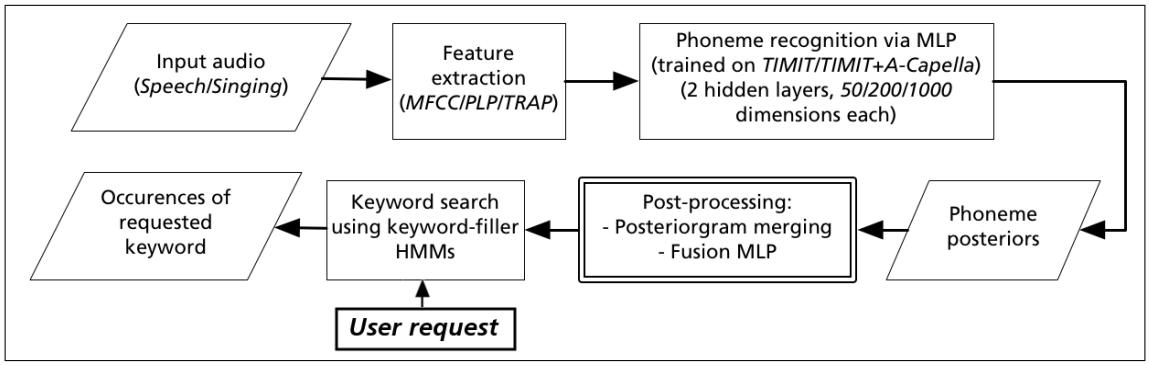

Figure 2: Overview of our keyword spotting system. Variable parameters are shown in italics.

The following two points described optional post-processing steps on the phoneme posteriorgrams.

3a. Posteriorgram merging For this post-processing step, we take the phoneme posterior results that were obtained using different feature sets and average them. We tested both the combinations of PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP.

3b. Fusion MLP classifier As a second post-processing option, we concatenate phoneme posteriors obtained by using different feature sets and run them through a fusion MLP classifier to create better posteriors. We again tested the combinations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP.

4. Keyword spotting The resulting phoneme posteriorgrams are then used to perform the actual keyword spotting. As mentioned above, we employ an acoustic approach. It is based on keyword-filler Hidden Markov Models (HMMs) and has been described in [14] and [8].

In general, two separate HMMs are created: One for the requested keyword, and one for all non-keyword regions (=filler). The keyword HMM is generated using a simple left-to-right topology with one state per keyword phoneme, while the filler HMM is a fully connected loop of states for all phonemes. These two HMMs are then joined. Using this composite HMM, a Viterbi decode is performed on the phoneme posteriorgrams. Whenever the Viterbi path passes through the keyword HMM, the keyword is detected. The likelihood of this path can then be compared to an alternative path through the filler HMM, resulting in a detection score. A threshold can be employed to only return highly scored occurrences. Additionally, the parameter β can be tuned to adjust the model. It determines the likelihood of transitioning from the filler HMM to the keyword HMM. The whole process is illustrated in figure 3.

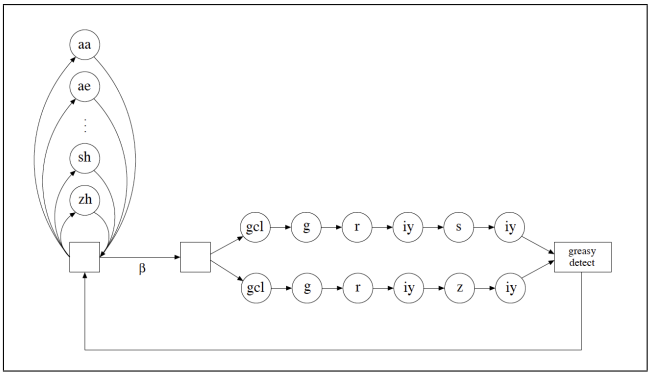

Figure 3: Keyword-filler HMM for the keyword "greasy" with filler path on the left hand side and two possible keyword pronunciation paths on the right hand side. The parameter β determines the transition probability between the filler HMM and the keyword HMM. [8]

We use the F_1 measure for evaluation. Results are considered to be true positives when a keyword is spotted somewhere in an expected utterance. Since most utterances contain one to ten words, we consider this to be sufficiently exact. Additionally, we evaluate the precision of the results. For the use cases described in section 1, users will usually only require a number of correct results, but not necessarily all the occurrences of the keyword in the whole database. We consider a result to be correct when the keyword is found as part of another word with the same pronunciation. The reasoning behind this is that a user who searched

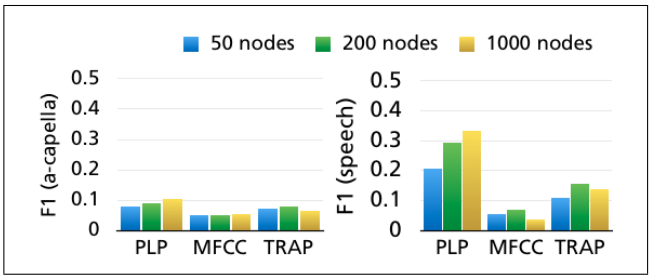

Figure 4: F_1 measures for a-capella data (left) and speech (right) when using PLP, MFCC, or TRAP features. The MLPs for phoneme recognition had two hidden layers with 50, 200, or 1000 nodes each.

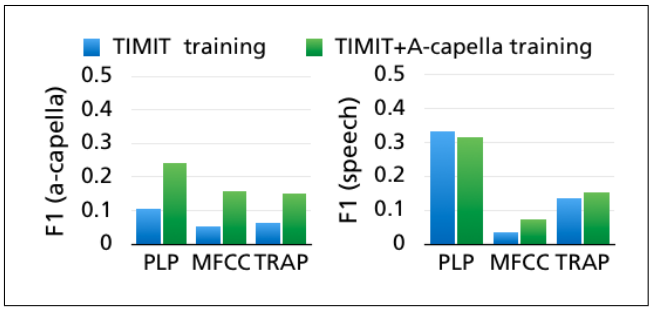

Figure 5: F_1 measures for a-capella data (left) and speech (right) when the recognition is trained only on TIMIT speech data (blue) or on a mix of TIMIT and a-capella data (green).

for the keyword "time" might also accept occurrences of the word "times" as correct.

5. EXPERIMENTS

5.1 Experiment 1: Oracle search

As a precursor to the following experiments, we first tested our keyword spotting approach on oracle posteriorgrams for the a-capella data. This was done to test the general feasibility of the algorithm for keyword spotting on singing data with its highly variable phoneme durations.

The oracle posteriorgrams were generated by converting the phoneme annotations to posteriorgram format by setting the likelihoods of the annotated phonemes to 1 during the corresponding time segment and everything else to 0. A keyword search on these posteriorgrams resulted in F_1 measures of 1 for almost all keywords. In cases where the result was not 1, we narrowed the reasons down to annotation errors and pronunciation variants that we did not account for. We conclude that our keyword-filler approach is generally useful for keyword spotting on a-capella data, and our focus in the following experiments is on obtaining good posteriorgrams from the audio data.

5.2 Experiment 2: A-Capella vs. Speech

For our first experiment, we run our keyword spotting system on the a-capella singing data, and on the same utterances spoken by a single speaker. We evaluate all three feature datasets (MFCC, PLP, TRAP) separately. The recognition MLP is trained on TIMIT speech data only. We also test three different sizes for the two hidden MLP layers: 50 nodes, 200 nodes, and 1000 nodes in each layer. The results are shown in figure 4.

As described in section 2.2, we expected keyword spotting on singing to be more difficult than on pure speech because of a larger pitch range, more pronunciation variations, etc. Our results support this assumption: In speech, keywords are recognized with an average F_1 measure of 33% using only PLP features, while the same system results in an average F_1 of only 10% on a-capella singing.

For both data sets, an MLP with 200 nodes in the hidden layers shows a notable improvement over one with just 50. When using 1000 nodes, the result still improves by a few percent in most cases.

When looking at the features, PLP features seem to work best by a large margin, with TRAPs coming in second. It is notable, however, that some keywords can be detected much better when using MFCCs or TRAPs than PLPs (e.g. "sing", "other", "hand", "world", "tears", "alright"). As described in [5] and [10], certain feature sets represent some phonemes better than others and can therefore balance each other out. A combination of the features might therefore improve the whole system.

Evaluation of the average precision (instead of F_1 measure) shows the same general trend. The best results are again obtained when using PLP features and the largest MLP. The average precision in this configuration is 16% for a-capella singing and 37% for speech. (While the difference is obvious, the result is still far from perfect for speech. This demonstrates the difficulty of the recognition process without a-priori knowledge.)

5.3 Experiment 3: Training including a-capella data

As a measure to improve the phoneme posteriorgrams for a-capella singing, we next train our recognition MLP with both TIMIT and a part of the a-capella data. We mix in about 50% of the a-capella clips with the TIMIT data. They make up about 10% of the TIMIT speech data. The results are shown in figure 5 (only the results for the largest MLP are shown).

This step improves the keyword recognition on a-capella data massively in all feature and MLP configurations. The best result still comes from the biggest MLP when using PLP features and is now an average F_1 of 24%. This step makes the recognition MLP less specific to the properties of pure speech and therefore does not improve the results for the speech data very much. It actually degrades the best result somewhat.

The effect on the average precision is even greater. The a-capella results are improved by 10 to 15 percentage points for each feature set. On speech data, the PLP precision decreases by 7 percentage points.

5.4 Experiment 4: Posterior merging

As mentioned in experiment 2, certain feature sets seem to represent some keywords better than others. We therefore concluded that combining the results for all features could improve the recognition result.

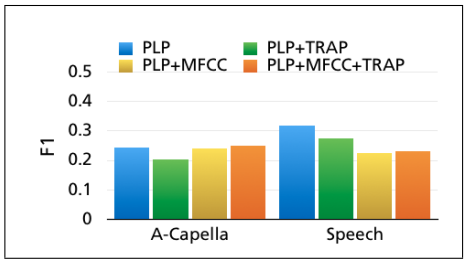

Figure 6: F_1 measures for a-capella data (left) and speech (right) when posteriorgrams for two or three features are merged. The configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP are shown and compared to the PLP only result.

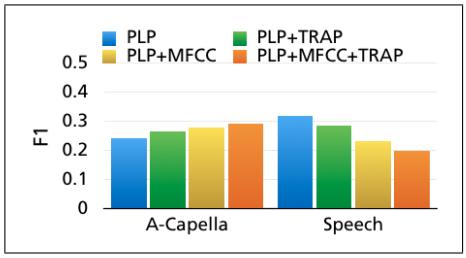

Figure 7: F_1 measures for a-capella data (left) and speech (right) when posteriorgrams for two or three features are fed into a fusion classifier. The configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP are shown and compared to the PLP only result.

To this end, we tested merging the phoneme posteriorgrams between the MLP phoneme recognition step and the HMM keyword spotting step. In order to do this, we simply calculated the average values across the posteriors obtained using the three different feature data sets. This was done for all phonemes and time frames. Keyword spotting was then performed on the merged posteriorgrams. We tested the configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP. The results are shown in figure 6.

Posterior merging seems to improve the results for a-capella singing somewhat and works best when all three feature sets are used. The F_1 measure on a-capella singing improves from 24% (PLP) to 27%. It does not improve the speech result, where PLP remains the best feature set.

5.5 Experiment 5: Fusion classifier

After the posterior merging, we tested a second method of combining the feature-wise posteriorgrams. In this second method, we concatenated the posteriorgrams obtained from two or all three of the feature-wise MLP recognizers and ran them through a second MLP classifier. This fusion MLP was trained on a subset of the a-capella data. This fusion classifier generates new, hopefully improved phoneme posteriorgrams. HMM keyword spotting is then performed on these new posteriorgrams. We again tested the configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP. The results are shown in figure 7. The fusion classifier improves the F_1 measure for a-capella singing by 5 percentage points. The best result of 29% is obtained when all three feature sets are used. Precision improves from 24% to 31%. However, the fusion classifier makes the system less specific towards speech and therefore decreases the performance on speech data.

5.6 Variation across keywords

The various results we presented in the previous experiments varies widely across the 51 keywords. This is a common phenomenon in keyword spotting. In many approaches, longer keywords are recognized better than shorter ones because the Viterbi path becomes more reliable with each additional phoneme. This general trend can also be seen in our results, but even keywords with the same number of phonemes vary a lot. The precisions vary similarly, ranging between 2% and 100%.

When taking just the 50% of the keywords that can be recognized best, the average F_1 measure for the best approach (fusion MLP) jumps from 29% to 44%. Its precision increases from 31% to 46%. We believe the extremely bad performance of some keywords is in part due to the small size of our data set. Some keywords occurred in just one of the 19 songs and were, for example, not recognized because the singer used an unusual pronunciation in each occurrence or had an accent that the phoneme recognition MLP was not trained with. We therefore believe these results could improve massively when more training data is used.

6. CONCLUSION

In this paper, we demonstrated a first keyword spotting approach for a-capella singing. We ran experiments for 51 keywords on a database of 19 a-capella pop songs and recordings of the spoken lyrics. As our approach, we selected acoustic keyword spotting using keyword-filler HMMs. Other keyword spotting approaches depend on learning average phoneme durations, which vary a lot more in a-capella singing than in speech. These approaches therefore cannot directly be transferred.

As a first experiment, we tested our approach on oracle phoneme posteriorgrams and obtained almost perfect results. We then produced "real world" posteriorgrams using MLPs with two hidden layers which had been trained on TIMIT speech data. We tested PLP, MFCC, and TRAP features. The training yielded MLPs with 50, 200, and 1000 nodes per hidden layer. We observed that the 200 node MLP produced significantly better results than the 50 node MLPs in all cases ($p < 0.0027$), while the 1000 node MLPs only improved upon this result somewhat. PLP features performed significantly better than the two other feature sets. Finally, keywords were detected much better in speech than in a-capella singing. We expected this result due to the specific characteristics of singing data (higher variance of frequencies, more pronunciation variants).

We then tried training the MLPs with a mixture of TIMIT speech data and a portion of our a-capella data. This improved the results for a-capella singing greatly.

We noticed that some keywords were recognized better when MFCCs or TRAPs were used instead of PLPs. We therefore tried two approaches to combine the results for

all three features: Posterior merging and fusion classifiers. Both approaches improved the results on the a-capella data. The best overall result for a-capella data was produced by a fusion classifier that combined all three features (29%).

As expected, keyword spotting on a-capella singing proved to be a harder task than on speech. The results varied widely between keywords. Some of the very low results arise because the keyword in question only occurred in one song where the singer used an unusual pronunciation or had an accent. The small size of our data set also poses a problem when considering the limited number of singers. The acoustic model trained on speech data and a part of the a-capella data might be subject to overfitting to the singers' vocal characteristics.

In contrast, the recognition worked almost perfectly for keywords with more training data. Keyword length also played a role. When using only the 50% best keywords, the average F_1 measure increased by 15 percentage points. Finally, there are many applications where precision plays a greater role than recall, as described in section 4. Our system can be tuned to achieve higher precisions than F_1 measures and is therefore also useful for these applications. We believe that the key to better keyword spotting results lies in better phoneme posteriorgrams. A larger a-capella data set would therefore be very useful for further tests and would provide more consistent results.

7. FUTURE WORK

As mentioned in section 2, more sophisticated keyword spotting systems for speech incorporate knowledge about plausible phoneme durations (e.g. [9]). In section 2.2, we showed why this approach is not directly transferable to singing: The vowel durations vary too much. However, consonants are not affected. We would therefore like to start integrating knowledge about average consonant durations in order to improve our keyword spotting system. In this way, we hope to improve the results for the keywords that were not recognized well by our system.

Following this line of thought, we could include even more language-specific knowledge in the shape of a language model that also contains phonotactic information, word frequencies, and phrase frequencies. We could thus move from a purely acoustic approach to a phonetic (lattice-based) approach.

We will also start applying our approaches to polyphonic music instead of a-capella singing. To achieve good results on polyphonic data, pre-processing will be necessary (e.g. vocal activity detection and source separation).

8. REFERENCES

[1] J. S. Bridle. An efficient elastic-template method for detecting given words in running speech. In *Brit. Acoust. Soc. Meeting*, pages 1 – 4, 1973.

[2] C. Dittmar, P. Mercado, H. Grossmann, and E. Cano. Towards lyrics spotting in the SyncGlobal project. In *3rd International Workshop on Cognitive Information Processing (CIP)*, 2012.

[3] H. Fujihara and M. Goto. Three techniques for improving automatic synchronization between music and lyrics: Fricative detection, filler model, and novel feature vectors for vocal activity detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 69–72, Las Vegas, NV, USA, 2008.

[4] H. Grossmann, A. Kruspe, J. Abesser, and H. Lukashevich. Towards cross-modal search and synchronization of music and video. In *International Congress on Computer Science Information Systems and Technologies (CSIST)*, Minsk, Belarus, 2011.

[5] J. K. Hansen. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *9th Sound and Music Computing Conference (SMC)*, pages 494–499, Copenhagen, Denmark, 2012.

[6] H. Hermansky and S. Sharma. Traps – classifiers of temporal patterns. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, pages 1003–1006, Sydney, Australia, 1998.

[7] J. S. Garofolo et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus. Technical report, Linguistic Data Consortium, Philadelphia, 1993.

[8] A. Jansen and P. Niyogi. An experimental evaluation of keyword-filler hidden markov models. Technical report, Department of Computer Science, University of Chicago, 2009.

[9] K. Kintzley, A. Jansen, K. Church, and H. Hermansky. Inverting the point process model for fast phonetic keyword search. In *INTERSPEECH*. ISCA, 2012.

[10] A. M. Kruspe, J. Abesser, and C. Dittmar. A GMM approach to singing language identification. In *53rd AES Conference on Semantic Audio*, London, UK, 2014.

[11] A. Mandal, K. R. P. Kumar, and P. Mitra. Recent developments in spoken term detection: a survey. *International Journal of Speech Technology*, 17(2):183–198, June 2014.

[12] A. Mesaros and T. Virtanen. Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010(4), January 2010.

[13] A. Moyal, V. Aharonson, E. Tetariy, and M. Gishri. *Phonetic Search Methods for Large Speech Databases*, chapter 2: Keyword spotting methods. Springer, 2013.

[14] I. Szoeke, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, and J. Cernocky. Phoneme based acoustics keyword spotting in informal continuous speech. In V. Matousek, P. Mautner, and T. Pavelka, editors, *TSD*, volume 3658 of *Lecture Notes in Computer Science*, pages 302–309. Springer, 2005.