BOOTSTRAPPING A SYSTEM FOR PHONEME RECOGNITION AND KEYWORD SPOTTING IN UNACCOMPANIED SINGING

Anna M. Kruspe
Fraunhofer IDMT, Ilmenau, Germany
kpe@idmt.fraunhofer.de

ABSTRACT

Speech recognition in singing is still a largely unsolved problem. Acoustic models trained on speech usually produce unsatisfactory results when used for phoneme recognition in singing. On the flipside, there is no phonetically annotated singing data set that could be used to train more accurate acoustic models for this task.

In this paper, we attempt to solve this problem using the *DAMP* data set which contains a large number of recordings of amateur singing in good quality. We first align them to the matching textual lyrics using an acoustic model trained on speech.

We then use the resulting phoneme alignment to train new acoustic models using only subsets of the *DAMP* singing data. These models are then tested for phoneme recognition and, on top of that, keyword spotting. Evaluation is performed for different subsets of *DAMP* and for an unrelated set of the vocal tracks of commercial pop songs. Results are compared to those obtained with acoustic models trained on the *TIMIT* speech data set and on a version of *TIMIT* augmented for singing. Our new approach shows significant improvements over both.

1. INTRODUCTION

Automatic speech recognition encompasses a large variety of research topics, but the developed algorithms have so far rarely been adapted to singing. Most of these tasks become harder when used on singing because singing data has different characteristics, which are also often more varied than in pure speech [12] [2]. For example, the typical fundamental frequency for women in speech is between 165 and $200Hz$, while in singing it can reach more than $1000Hz$. Other differences include harmonics, durations, pronunciation, and vibrato.

Speech recognition in singing can be used in many interesting practical applications, such as automatic lyrics-to-music alignment, keyword spotting in songs, language identification of musical pieces or lyrics transcription.

A first step in many of these tasks is the recognition of phonemes in the audio recording. We showed in [9] that phoneme recognition is a bottleneck in tasks such as language identification and keyword spotting in singing. Other publications also demonstrate that phoneme recognition on singing is more difficult than on speech [15] [5] [12]. This is further compounded by the fact that models are usually trained on pure speech data.

As shown on a small scale in [5] and [9], recognition gets better when singing is used as part of the training data. This has so far not been done comprehensively due to the lack of singing data sets annotated with phonemes or words.

In this paper, we present a new approach to training acoustic models on actual singing data. This is done by first assembling the data from a set of recordings of unaccompanied singing and the matching textual lyrics. These lyrics are then automatically aligned to the audio data using models trained solely on speech. Next, the resulting annotated data sets are used to train new acoustic models for phoneme recognition in singing. We then evaluate the phoneme recognition results on different subsets of the singing corpus and on an unrelated data set of vocal tracks. Finally, we also use the recognized phonemes to perform keyword spotting.

This paper is structured as follows: We first present the state of the art in section 2 and the data sets in section 3. Then, we describe our proposed approach in more detail in section 4. The experiments and their results are presented in sections 5 and 6. Finally, we give a conclusion in section 7 and make suggestions for future experiments in section 8.

2. STATE OF THE ART

2.1 Phoneme recognition in singing

As described in [12], [2], and [9], there are significant differences between speech and singing audio, such as pitch and harmonics, vibrato, phoneme durations and pronunciation. These factors make phoneme recognition on singing more difficult than on speech. It has only been a topic of research for the past few years.

Fujihara et al. first presented an approach using Probabilistic Spectral Templates to model phonemes in [3]. The phoneme models are gender-specific and only model five vowels, but also work for singing with instrumental accompaniment. The best result is 65% correctly classified frames.

In [4], Gruhne et al. describe a classical approach that employs feature extraction and various machine learning

algorithms to classify singing into 15 phoneme classes. It also includes a step that removes non-harmonic components from the signal. The best result of 58% correctly classified frames is achieved with Support Vector Machine (SVM) classifiers. The approach is expanded upon in [17].

Mesaros presented a complex approach that is based on Hidden Markov Models which are trained on Mel-Frequency Cepstral Coefficients (MFCCs) and then adapted to singing using three phoneme classes separately [15] [14]. The approach also employs language modeling and has options for vocal separation and gender and voice adaptation. The achieved phoneme error rate on unaccompanied singing is 1.06 without adaptation and 0.8 with singing adaptation using 40 phonemes (the error rate greater than one means that there were more insertion, deletion, or substitution errors than phoneme instances). The results also improve when using gender-specific adaptation (to an average of 0.81%) and even more when language modeling is included (to 0.67%).

Hansen presents a system in [5] which combines the results of two Multilayer Perceptrons (MLPs), one using MFCC features and one using TRAP (Temporal Pattern) features. Training is done with a small amount of singing data. Viterbi decoding is then performed on the resulting posterior probabilities. On a set of 27 phonemes, this approach achieves a recall of up to 48%.

Finally, we trained new models for phoneme recognition in singing by modifying speech data to make it more "song-like" [11]. We employed time-stretching, pitch-shifting, and vibrato generation algorithms. Using a model trained on speech data with all three modifications, we obtained 18% correctly classified frames (6% improvement) and a weighted phoneme error rate of 0.71 (0.06 improvement).

Generally, comparing the existing approaches is not trivial since different datasets, different phoneme sets, and different evaluation measures are used.

2.2 Keyword spotting in singing

A first approach to keyword spotting in singing was presented in [9]. This approach employs keyword-filler HMMs which detect the keyword. The recognition is performed on phoneme posteriorgrams, which were generated with acoustic models trained on speech. We obtained F_1 measures of 33% for spoken lyrics and 24% for a-capella singing. Using post-processing techniques on the posteriorgrams, the a-capella result was improved up to 27%.

In [10], we improved upon this result by employing phoneme duration modeling algorithms. The best result on a-capella singing was an F_1 measure of 39%.

In [1], HMM models and position-HMM-DBNs were employed to search for certain phrases of lyrics in traditional Turkish music. The approach obtained an F_1 measure of 13% for the 1-best result.

3. DATA SETS

3.1 Speech data sets

For training our baseline phoneme recognition models, we used the well-known *Timit* speech data set [7]. Its training section consists of 4620 phoneme-annotated English utterances spoken by native speakers. Each utterance is a few seconds long.

Additionally, we also trained phoneme models on a modification of *Timit* where pitch-shifting, time-stretching, and vibrato were applied to the audio data. This process was described in [11]. The data set will be referred to as *TimitM*.

3.2 Singing data sets

3.2.1 Damp

For training models specific to singing, we used the *DAMP* data set, which is freely available from Stanford University [1] [16]. 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 little background noise, but come from a 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.

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

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

#	Keywords
2	eyes
3	love, away, time, life, night
4	never, baby, world, think, heart, only, every
5	always, little

Table 1: All 15 tested keywords, ordered by number of phonemes.

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

3.2.2 Acap

We also ran some tests on a small data set of the vocal tracks of 15 pop songs, which were hand-annotated with phonemes and words. This data set was first presented in [5]. Despite the small size, we provide results on this data set for comparison with our previous approaches, and because the ground truth annotations can be assumed to be correct (in contrast with the automatically generated annotations of the *Damp*-based data sets).

3.3 Keywords

From the 301 different song lyrics of the *Damp* data sets, 15 keywords were chosen by semantic content and frequency to test our keyword spotting algorithm. Each keyword occurs in at least 50 of the 301 songs. The keywords are shown in table 1.

4. PROPOSED APPROACH

4.1 Lyrics alignment

Since the textual lyrics were not aligned to the singing audio data, we first performed an automatic alignment step. A monophone HMM acoustic model trained on *Timit* using HTK was used. Alignment was performed on the word and phoneme levels. This is the same principle of so-called "Forced Alignment" that is commonly used in Automatic Speech Recognition [8] (although it is commonly done on shorter utterances). We hand-checked some examples and found the alignment to already be very good over-all. Of course, errors cannot be avoided when doing automatic forced alignment. We considered repeating this process with the newly trained models, but preliminary tests suggested that this would not improve the alignments very much.

The resulting annotations were used in the following experiments. This approach provided us with a large amount of annotated singing data, which could not feasibly have been done manually.

4.2 New acoustic models

Using these automatically generated annotations, we then trained new acoustic models on *DampB*, *DampBB*, *DampBB_small*, *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 with a sigmoid activation function. The output layer corresponds to 37 monophones. Inputs are either frame-wise MFCCs (39 dimensions) or MFCCs with 4 context frames on either side (351 dimensions).

4.3 Phoneme recognition and evaluation

Using these models, phoneme posteriorgrams were then generated on the test data sets (*DampTestF*, *DampTestM*, and *Acap*). For all non-gender dependent acoustic models, results over both of the *DampTest* sets were averaged.

The recognized phonemes were then evaluated using the percentage of correct frames, the phoneme error rate, and the weighted phoneme error rate as evaluation measures (see [11]). In the case of the *DampTest* data sets, the results were compared to the automatic alignment results, which is a potential source of error.

4.4 Keyword spotting and evaluation

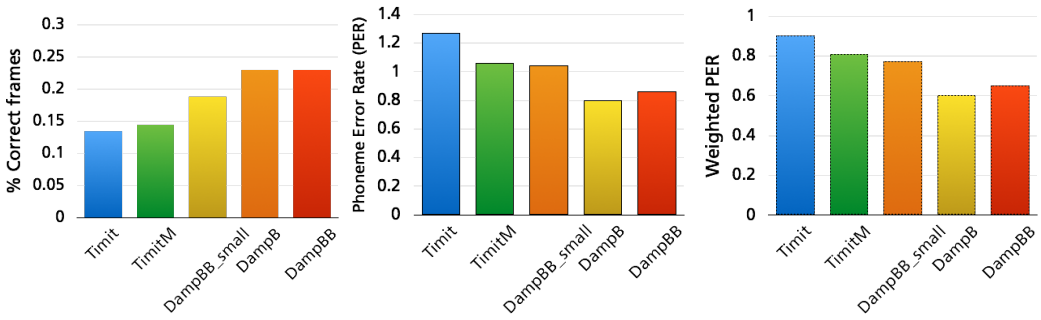
Finally, the phoneme posteriorgrams were evaluated for keyword spotting. A keyword-filler HMM algorithm was employed. Keyword-filler HMMs consist of two sub-HMMs: One to model the keyword and one to model everything else (=filler). The keyword HMM has a simple left-to-right topology with one state per keyword phoneme. The filler HMM is a fully connected loop of all phonemes. When the Viterbi path with the highest likelihood passes through the keyword HMM rather than the filler loop, the keyword is detected. We previously employed this approach in [9]. However, the evaluation data set based on *Damp* is a different, much bigger set of recordings. The keyword set was changed to better reflect frequently occurring words in these songs. Additionally, the keyword detection was performed on whole songs, which may be more realistic for practical applications. For comparison, results on our old data set (*Acap*) for whole songs are also provided below. Song-wise F_1 measures were calculated for evaluation.

5. PHONEME RECOGNITION EXPERIMENTS AND RESULTS

5.1 Experiment A: Comparison of models trained on *Timit* and *Damp* data sets

In our first experiments, we generated phoneme posteriorgrams on the data sets *DampTestF* and *DampTestM* using the models trained on the two variants of *Timit* and on the three differently-sized *Damp* training sets that were not split by gender. The results are averaged over both sets. For comparison, we also generated these posteriorgrams on *Acap*. The results for the *DampTest* sets in terms of percentage of correct frames, phoneme error rate, and weighted phoneme error rate are shown in figure 1.

Models trained on the modified version of *Timit* already show some improvement over plain *Timit* [11], but even the small *Damp* training set improves the result significantly more. As mentioned before, this data set is actually smaller than *Timit*. The percentage of correct frames rises



(a) Percentage of correct frames (b) Phoneme error rate (c) Weighted phoneme error rate

Figure 1: Evaluation measures for the results obtained on the *DampTest* data sets using models trained on *Timit* and on various *Damp*-based data sets.

from 13% to 19%, the phoneme error rate sinks from 1.27 to 1.04, and the weighted phoneme error rate from 0.9 to 0.77.

When the whole set of 6000 recordings is used for training (*DampB*), the percentage of correct frames even rises to 23%, while the phoneme error rate falls to 0.8 and the weighted phoneme error rate to 0.6. When using the smaller, more balanced version (*DampBB*), these results are somewhat worse, but not much, with 23% correct frames, a phoneme error rate of .86, and a weighted phoneme error rate of 0.65. This is particularly interesting because this data set is only 4% the size of the bigger one and training is therefore much faster.

The results on the *Acap* data set show a similar improvement, but are better in general. The percentage of correctly classified frames jumps from 12% to 22%, 25%, and 27% for *DampBB_small*, *DampB*, and *DampBB* respectively. The weighted phoneme error rate sinks from 0.8 to 0.69, 0.61, and 0.56. Since this data set is has been annotated by hand and is completely different material from the training data sets, we are confident that our approach is able to model the properties of each phoneme, rather than reproducing the model that was used for aligning the singing training sets. The somewhat better values might be caused by these more accurate annotations, too, or by the fact that these are recordings of professional singers who enunciate more clearly.

5.2 Experiment B: Influence of context frames

We then ran the same experiment again, but this time used models that were trained with 4 context frames on either side of each input frame. This provides more long-term information. The results are shown in figure 2. (In each figure, the "No context" part is the result from the previous experiment).

Surprisingly, using context frames did not improve the result in any case except for the *DampBB_small* models. Since this is the smallest data set, this improvement might happen just because the context frames virtually provide more training data for each phoneme. In the other cases, there already seems to be a sufficient amount of training data and the context frames may blur the training data instead of providing more information about the context of each phoneme. Additionally, it is possible that this approach compounds error that were made in the automatic alignment in the case of the bigger *Damp* training data sets.

The same effect can be observed when calculating these values on the hand-annotated *Acap* test data set. We therefore decided to not employ context frames in the following experiments. This also speeds up the training process.

5.3 Experiment C: Comparison of gender-dependent models

Finally, we generated phoneme posteriorgrams using models that were only trained on recordings of the same gender. I.e., for phoneme recognition on the *DampTestF* set, we used a model trained only on female singing recordings (*DampFB*). The results are shown in figure 3. (Note that the results for *DampB* and *DampBB* are different from the previous experiments because the test data sets were split by gender).

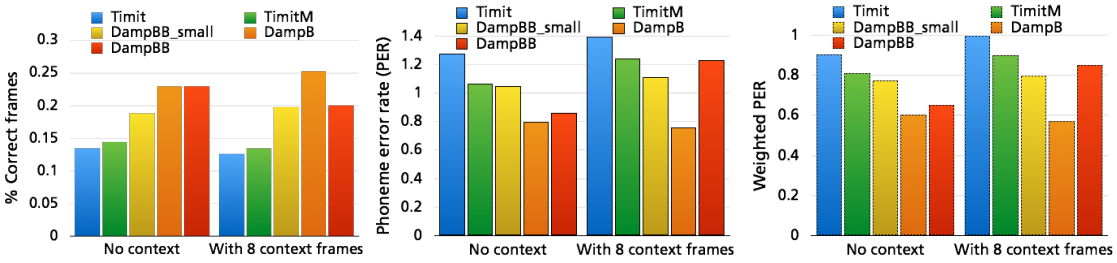
Surprisingly, the results do not improve when using gender-specific acoustic models. The percentage of correct frames, when compared to the results using the models trained on the *DampBB* drops sligthly from 23% to 21% for the female test set, and stays at 23% for the male one. The weighted phoneme error rate rises from 0.65 to 0.68 and from 0.65 to 0.69 for the female and male test sets respectively.

This might happen because the training data sets are slightly smaller, but, more likely, because some variation in the singing voices might be lost when using training data of only one gender. In singing, pitches cover a broader range than in speech. This effect might take away some of the improvement usually seen in speech recognition when using gender-specific models.

6. KEYWORD SPOTTING EXPERIMENTS AND RESULTS

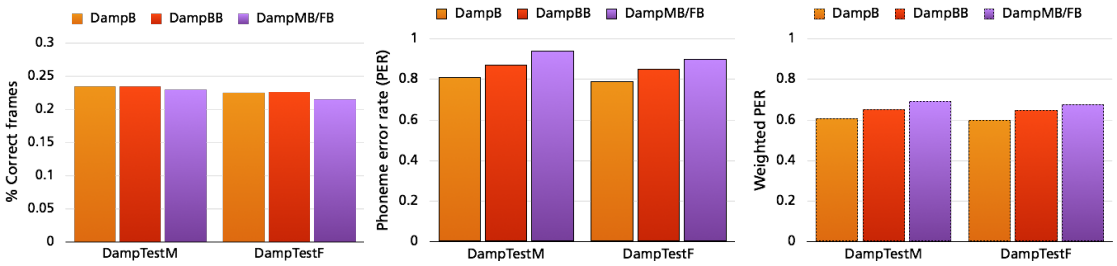
6.1 Experiment D: Comparison of models trained on *Timit* and *Damp* data sets

We then performed keyword spotting on the phoneme posteriorgrams from Experiment A. The results in terms of F_1



(a) Percentage of correct frames (b) Phoneme error rate (c) Weighted phoneme error rate

Figure 2: Evaluation measures for the results obtained on the *DampTest* data sets using models trained on *Timit* and on various *Damp*-based data sets with no context and with 8 context frames.



(a) Percentage of correct frames (b) Phoneme error rate (c) Weighted phoneme error rate

Figure 3: Evaluation measures for the results obtained on the *DampTestM* and *DampTestF* data sets using models trained on *Damp*-based data sets, mixed and split by gender.

measure across the whole *DampTest* sets are shown in figure 4a. Figure 4b show the results of the same experiment on the small *Acap* data set.

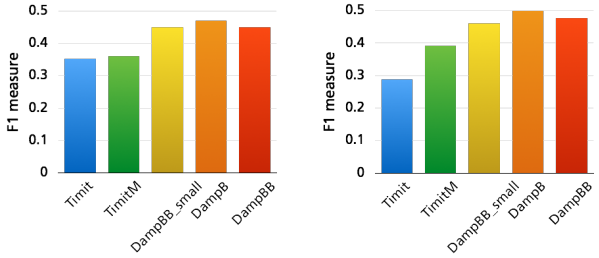
Across all keywords, we obtain a document-wise F_1 measure of 0.35 using the posteriorgrams generated with the *Timit* model on the *DampTest* data sets. This result is slightly higher for the *TimitM* models and rises to 0.45 using the model trained on the small *DampBB_small* singing data set. Surprisingly, the model trained on *DampBB* is only slightly better than the much smaller one. Using the very big *DampB* training data set, the F_1 measure reaches 0.47.

On the hand-annotated *Acap* test set, the difference is even more pronounced, rising from 0.29 for the *Timit* model to 0.5 for *DampB*. This might, again, be caused by the more accurate annotations or by the higher-quality singing. Additionally, the data set is much smaller with fewer occurrences of each keyword, which could emphasize both positive and negative tendencies in the detection.

6.2 Experiment E: Comparison of gender-dependent models

We also performed keyword spotting on the posteriorgrams generated with the gender-dependent models from Experiment C. The results are shown in figure 5.

In contrast to the phoneme recognition resultes from Experiment C, the gender-dependent models perform slightly better for keyword spotting than the mixed one of the same size, and almost as good as the one trained on much more data (*DampB*). The F_1 measures for the female test set are 0.48 for the *DampB* model, 0.45 for the *DampBB* model, and 0.46 for the *DampFB* model. For the male test set, they are 0.46 and 0.45 for the first two, and 0.46 for the *DampMB* model.



(a) F_1 measures for keyword spotting results on the *DampTest* data sets.

(b) Keyword spotting results on the *Acap* data set.

Figure 4: F_1 measures for keyword spotting results using posteriorgrams generated with various acoustic models.

6.3 Experiment F: Individual analysis of keyword results

Figure 6 shows the individual F_1 measures for each keyword using the best model (*DampB*), ordered by their occurrence in the *DampTest* sets from high to low (i.e. number of songs which include the song). There appears to be a tendency for more frequent keywords to be detected more accurately. This happens because a high recall is often achievable, while the precision depends very much on the accuracy of the input posteriorgrams. The more frequent a keyword, the easier it also becomes to achieve a higher precision for it.

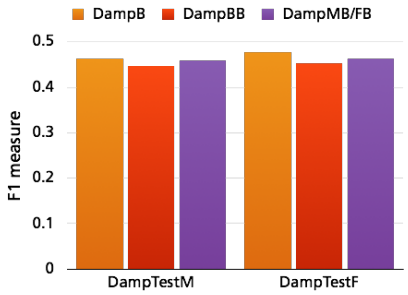


Figure 5: F_1 measures for keyword spotting results on the *DampTestM* and *DampTestF* data sets using mixed and gender-dependent models.

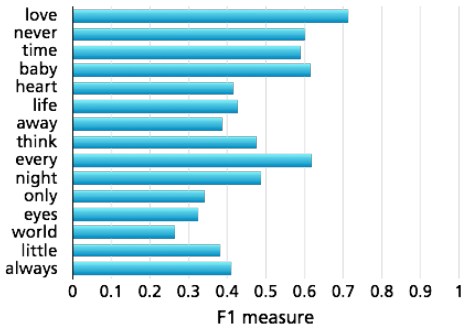


Figure 6: Individual F_1 measures for the results for each keyword, using the acoustic model trained on *DampB*.

As shown in literature [18], the detection accuracy also depends on the length of the keyword: Keywords with more phonemes are usually easier to detect. This might explain the relative peak for "every", "little", and "always", in contrast to "eyes" or "world". Since keyword detection systems tend to perform better for longer words and most of our keywords only have 3 or 4 phonemes, this result is especially interesting.

One potential source of error are sequences of phonemes that overlap with our keywords, but are not included in the calculation of the precision. Equally spelled words were included, but split phrases or other spellings were not (e.g. "away" as part of "castaway" would be counted, but "a way" would not be counted as "away"). This might artificially lower our results and we will look into possibilities for improvement in the future. Additionally, only one pronunciation for each keyword was provided, but there may be several possible.

7. CONCLUSION

In this paper, we trained new acoustic models on a large corpus of unaccompanied singing recordings. Since no annotations for these existed, we first had to automatically align lyrics to them. The new models could then directly be trained on these automatic annotations. To our knowledge, this has not been done before for singing.

We trained three different models with mixed gender recordings: One on 6000 full recordings of 301 songs, one on just 4% of this data, and one which was balanced by phonemes and is roughly half the size of the medium-sized one. We then tested their performance on two other subsets of the same corpus which did not overlap with the training data, and on a small unrelated data set of commercial vocal tracks which were hand-annotated.

In all cases, the three new models showed a strong improvement over those trained only on speech. Even the model trained on the smallest set produced a jump in correctly classified frames from 13% to 19%, and in weighted phoneme error rate from 0.9 to 0.77 on the large test set. With the model trained on the medium-sized data set, we obtained 23% correct frames and a weighted phoneme error rate of 0.65. With the biggest one, the weighted phoneme error rate falls to 0.6. The results are similar on the small hand-annotated test set.

We also tested acoustic models with 8 context frames, and models trained on gender-specific data. Neither of them showed improvement over the first ones.

We then performed keyword spotting for 15 keywords on phoneme posteriorgrams generated with these new models using a keyword-filler approach. The resulting F_1 measure rises from 0.35 for the models trained on speech to 0.47 for our new models. This result is especially interesting because most of the keywords have few phonemes. For keyword spotting, gender-dependent models perform slightly better than mixed-gender models of the same size.

8. FUTURE WORK

So far, we have only tested this approach using MFCC features. As shown in our previous experiments [9], other features like TRAP or PLP may work better on singing. So-called log-mel filterbank features have also been used successfully with DNNs [6]. Another interesting factor is the size and configuration of the classifiers, of which we have only tested one so far. Since the alignment appears to provide valid training data, we believe the features and model configuration could be the greatest sources of improvement.

We showed that there is only a slight amount of improvement between the model trained on all 6000 songs and the one trained only on 4% of this data. It would be interesting to find the exact point at which additional training data does not further improve the models. On the evaluation side, a keyword spotting approach that allows for pronunciation variants or sub-words may produce better results. Language modeling might also help to alleviate some of the errors made during phoneme recognition.

These models have not yet been applied to singing with background music, which would be interesting for practical applications. Since this would probably decrease the result when used on big, unlimited data sets, more specified systems would be more manageable, e.g. for specific music styles, sets of songs, keywords, or specialized applications. Searching for whole phrases instead of short keywords could also make the results better usable in practice.

As shown in [13] and [2], alignment of textual lyrics and singing already works well. A combined approach that also employs textual information could be very practical.

9. REFERENCES

[1] G. Dzhambazov, S. Sentürk, and X. Serra. Searching lyrical phrases in a-capella turkish makam recordings. In *Proceedings of the 16th International Conference on Music Information Retrieval (ISMIR)*, Malaga, Spain, 2015.

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

[3] H. Fujihara, M. Goto, and H. G. Okuno. A novel framework for recognizing phonemes of singing voice in polyphonic music. In *WASPAA*, pages 17–20. IEEE, 2009.

[4] M. Gruhne, K. Schmidt, and C. Dittmar. Phoneme recognition on popular music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, September 2007.

[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] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine*, 2012.

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

[8] D. Jurafsky and J. H. Martin. *Speech and language processing: An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2009.

[9] A. M. Kruspe. Keyword spotting in a-capella singing. In *15th International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.

[10] A. M. Kruspe. Keyword spotting in a-capella singing with duration-modeled HMMs. In *EUSIPCO*, Nice, France, 2015.

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

[12] A. Loscos, P. Cano, and J. Bonada. Low-delay singing voice alignment to text. In *Proceedings of the ICMC*, 1999.

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

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

[15] A. Mesaros and T. Virtanen. Recognition of phonemes and words in singing. In *ICASSP*, pages 2146–2149. IEEE, 2010.

[16] J. C. Smith. *Correlation analyses of encoded music performance*. PhD thesis, Stanford University, 2013.

[17] G. Szepannek, M. Gruhne, B. Bischl, S. Krey, T. Harczos, F. Klefenz, C. Dittmar, and C. Weihs. *Classification as a tool for research*, chapter Perceptually Based Phoneme Recognition in Popular Music. Springer, Heidelberg, 2010.

[18] A. J. K. Thambiratnam. *Acoustic keyword spotting in speech with applications to data mining*. PhD thesis, Queensland University of Technology, 2005.