TRAINING SPEAKER RECOGNITION MODELS WITH RECORDING-LEVEL LABELS

Tanel Alumäe

Laboratory of Language Technology
Tallinn University of Technology, Estonia

ABSTRACT

In this paper, we investigate training speaker recognition models using coarse-grained speaker labels provided only at the recording level. The approach is based on the recently proposed weakly supervised training method that allows to train a speaker recognition deep neural network using a special cost function that doesn't need segment-level annotations. Experiments are conducted on the VoxCeleb corpus. We show that without using any reference segment-level labeling, the method can achieve 1% speaker recognition error rate on the official VoxCeleb closed set speaker recognition test set, as opposed to 5.4% that was previously reported. By training a x-vector based speaker verification system on the resegmented and relabeled VoxCeleb corpus, we can achieve 4.57% EER on the VoxCeleb speaker verification test set which is a 17% relative improvement over the best system that uses the official VoxCeleb speaker annotations.

Index Terms— Speaker recognition, VoxCeleb, weakly supervised training

1. INTRODUCTION

Speaker identification models are usually trained on data where the speech segments corresponding to the target speakers are hand-annotated. However, the process of hand-labelling speech data is expensive and doesn't scale well, especially if a large set of speakers needs to be covered. This makes such models difficult to manage and to deploy. For training speaker verification models, a large corpus of speaker-segmented data is needed for training a speaker embedding system, in addition to the data from the enrolled speakers.

Recently, we proposed a method to train speaker identification models using only the information about speakers appearing in each of the recordings in training data, without any segment level annotation [1]. Obtaining or creating such training data is much easier than segment-annotated data. During training, speaker diarization and i-vector extraction is used to map different speakers in each recording to fixed-dimensional vectors. A deep neural network (DNN) is then trained using a special objective function that encourages the model to assign a similar average distribution of labels to the i-vectors of a single show as the annotation in the training data. Experimental results were performed on the VoxCeleb dataset of YouTube videos [2], where the method resulted in 94.6% speaker identification accuracy, greatly outperforming a baseline system that uses face identification for obtaining training data annotations.

This paper describes some improvements to the recently proposed weakly supervised speaker recognition method and applies the same method for speaker verification. Instead of using the speaker recognition DNN trained with weak supervision directly for speaker identification, it is used for relabeling the automatically segmented VoxCeleb corpus, resulting in almost four times more annotated speaker data than is provided with the official VoxCeleb annotations. Conventional i-vector [3] or x-vector [4] and LDA/PLDA [5] based speaker recognition methods can then be applied.

The remainder of the paper is organized as follows. First, we briefly describe the VoxCeleb corpus that the experiments are based on and that partly inspired this work. The next section gives an outline of the method for training a DNN for speaker identification, using only recording-level labels. Then, experiments with speaker identification and speaker verification are described, showing the effectiveness of the technique. Some directions for future work are given in the conclusion.

2. THE VOXCELEB CORPUS

When working with video data, speech segments that correspond to certain speakers can be identified based on face identification. This approach was used for constructing the VoxCeleb database [2] which contains hundreds of thousands of 'real world' utterances for over 1000 celebrities. The database was collected from YouTube, using the following workflow. First, a target speaker list was constructed, using an intersection of popular celebrity names and persons appearing in the VGG Face dataset [6]. Second, 50 top YouTube videos were retrieved for each person, using a query "<name> interview". Third, a face detector was used to identify faces in video frames. Fourth, audio-video synchronization between detected faces and speech was determined, in order to identify

This research has been supported by the Centre of Excellence in Estonian Studies (CEES, European Regional Development Fund).

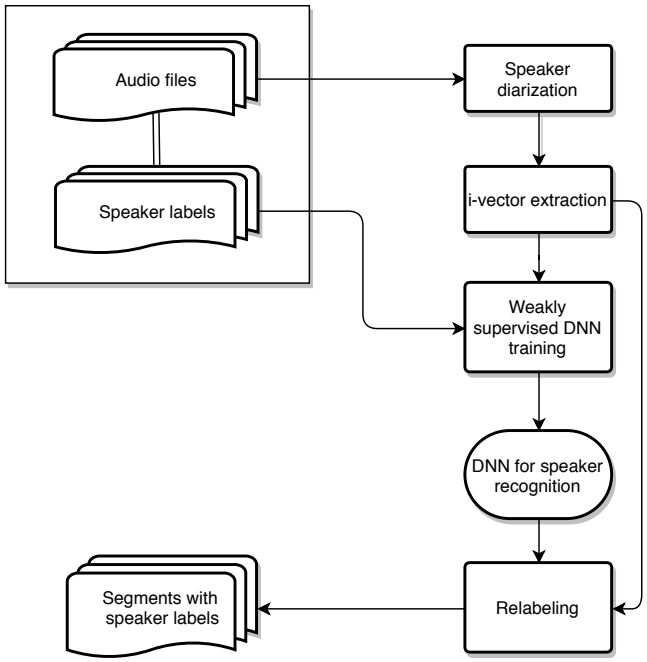

Fig. 1. *Architecture of the weakly-supervised training process.*

which face in the video corresponds to the current speaker (if any). Finally, a face verification system was used to determine whether the face of the active speaker corresponds to the target person of the video. This results in a database where high-confidence video segments corresponding to the target speakers are automatically annotated and can be used for training speaker identification models. Experiments showed that a speaker identification model trained on such data achieves 80.5% accuracy, using a closed set identification task. The paper also introduces a speaker verification task based on the corpus, using a subset of the speakers. The paper reports 7.8% equal error rate (EER) of the best system on the verification task.

Since its release, the VoxCeleb corpus has been used in various speaker recognition and speaker diarization studies, e.g. [7, 8, 9, 10, 11].

3. TRAINING SPEAKER IDENTIFICATION MODELS WITH RECORDING LEVEL LABELS

The method that we first proposed in [1] relies on an annotated set of speakers appearing in each audio recording. The set of speakers does not need to be exhaustive: only the speakers that need to be identifiable by the system must be included in the sets.

Outline of the training process is depicted in Figure 1. First, we apply a speaker diarization system to the training data. This step partitions the recording into homogeneous segments, discards non-speech segments and clusters the speech segments that are likely uttered by the same speaker.

Next, we use an i-vector extractor to compute i-vectors for all speakers in all recordings. The i-vector extractor can be trained on available labeled training data, possibly from another domain. Alternatively, the i-vector extractor can be trained on the automatically clustered speakers of the training set. We experimented with both methods and found no clear difference between those alternatives.

Here we give some useful notations for the rest of this section. Let $\mathcal{X} \subset R^D$ denote the D-dimensional feature space (i.e., the i-vector space) and $\mathcal{Y} = \{1, 2, ..., C\}$ the set of target speaker identities. The training corpus $\mathcal{D}$ contains a set of N audio recordings $\{X_n\}_{n=1,2,...,N}$ where each recording X_n contains a set of i-vectors for the diarized speakers $X_n = \{x_{mn}\}_{m=1,2,...,M_n}$, with $x_{mn} \in \mathcal{X}$. The training corpus also contains the corresponding sets of speaker labels for each recording $\{Y_n\}_{n=1,2,...,N}$ where $Y_n \subset \mathcal{Y}$. Note that the number of speaker labels does not have to agree with number of i-vectors for the corresponding recording, and the correspondence between x_{mn} and the elements in $\{Y_n\}$ is not known. The task is to train a model to classify i-vectors based on the speaker identities, i.e., to learn a classification function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from the training dataset.

We use a feed-forward neural network to learn the mapping from the i-vector space to the speaker label space. First, we augment the set of target speaker identities with a special $\langle unk \rangle$ identity reserved for unknown speakers, $\mathcal{Y}' = \mathcal{Y} \cup \{\langle unk \rangle\}$. The neural network takes i-vectors as input and has $|\mathcal{Y}'|$ outputs. The softmax function is used in the final layer of a neural network.

The neural network is trained using an objective function defined at the recording level, not at the single training sample level as usually. Specifically, we want the neural network to predict a similar set of speakers for the set of i-vectors as is defined in the metadata. To achieve this, we first define the expected average distribution over the speaker labels for each recording as

$$\bar{\tilde{p}}_n(y_i) = \begin{cases} \frac{1}{|X_n|}, & \text{if } y_i \in Y_n \\ max(0, 1 - \frac{|Y_n|}{|X_n|}), & \text{if } y_i = \langle unk \rangle \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The recording level objective function is the Kullback-Leibler divergence between the expected average distribution and model's expected average conditional distribution:

$$D(\bar{\tilde{p}}||\bar{\hat{p}}_\theta) = \sum_y \bar{\tilde{p}} \log \frac{\bar{\tilde{p}}}{\bar{\hat{p}}_\theta} \quad (2)$$

The idea of this objective function is that we don't know the exact correspondence between the i-vectors and speakers of a recording, but we want exactly a single i-vector to be assigned to each speaker of the show, and the rest (if any) to be absorbed by the class that is reserved for unknown speakers. Clearly, the assumption that only one i-vector corresponds to

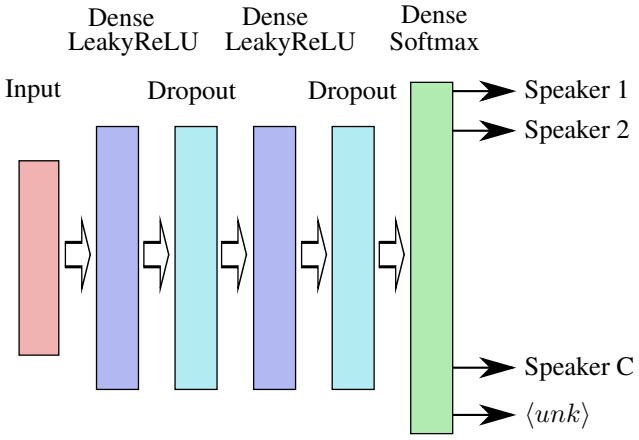

Fig. 2. *Architecture of the speaker identification DNN.*

a labeled speaker is not always true: in broadcast news, a news anchor could be speaking at the beginning of the news show with background music, and later without music, which usually causes the speaker diarization module to split the single speaker into two pseudo-speakers. Similarly, a reporter could speak both in a studio setting and with a noisy background in a single news show, also resulting in two i-vectors. However, we haven't found this to cause any noticeable problems.

The proposed method works only if the recordings in the training data have sufficiently different speaker distributions. Note that the objective function does not encourage the neural network in any way to assign a high probability to a particular speaker in the recording – given a single recording and a set of speakers for that recording, the objective function can be minimized by predicting a uniform distribution over all i-vectors for all speakers appearing in that particular recording. However, since we require the recordings to have different speaker distributions, the neural network has to start assigning non-uniform distributions to the i-vectors of a show, in order to better fit the data.

The neural network also doesn't learn to differentiate between the speakers that occur always together in the same recordings.

The training algorithm is summarized in listing 1.

We used a very simple deep neural network as the underlying model (Figure 2) . The network has two fully-connected hidden layers, using the leaky rectified linear units. The number of hidden units was optimized on development data. We found that dropout layers added after the dense layers improve performance of the model.

Once the speaker recognition DNN is trained, it can be used directly for speaker identification, as we proposed in [1]. It can be also used for relabeling all segments found during speaker diarization, resulting in new segment-level annotations for the whole dataset. Since the DNN has intrinsically a dedicated output for the unknown speaker class, we can simply discard segments that are classified to the unknown speaker. The resulting annotations can be used for performing LDA/PLDA based speaker recognition, or training a speaker embedding system for speaker verification.

The segmentations generated using the described method are available for download[1].

Data:
List of target speaker names: $\mathcal{Y} = \{1, 2, ..., C\}$;
A set of N diarized audio recordings with metadata, consisting of

- i-vectors $\{X_1, ..., X_N\}$, $X_n = \{x_{mn}\}$
- Speaker names: $\{Y_1, ..., Y_N\}, Y_n \subset \mathcal{Y}$

Result: Trained model that maps $\mathcal{X} \rightarrow \mathcal{Y}$
Initialize neural network with parameters θ;
Precompute expected average speaker distribution $\bar{\tilde{p}}_n$ for each recording according to eq. 1;
while *training hasn't converged* **do**
  Shuffle training data;
  for $n = 1...N$ **do**
    Compute the neural network predictions $\hat{p}_\theta^m$ for i-vectors x_{mn};
    Compute average predictions $\bar{\hat{p}}_\theta = \frac{1}{M} \sum_{m=1..M} \hat{p}_\theta^m$;
    Update model weights: $\theta \leftarrow SGD(D(\bar{\tilde{p}}_n || \bar{\hat{p}}_\theta))$;
  end
end

Algorithm 1: Algorithm for training a neural network on weakly labeled data.

Table 1. *Statistics of the VoxCeleb dataset. Medians shown with lower and upper quartiles where appropriate.*

# of target speakers	1251
# of videos per target speaker	14 (13 / 22)
Video length (seconds)	268 (171 / 426)
# of diarized speakers per video	3 (2 / 4)

4. EXPERIMENTAL RESULTS

4.1. Data

As described in section 2, VoxCeleb is a dataset that contains YouTube videos corresponding to over 1000 celebrities. The dataset is collected automatically by retrieving top matches from YouTube for queries "<name> interview", for a predefined list of celebrity names. Some statistics of the dataset is given in Table 1.

For speaker identification, the training and testing is performed on the same list of speakers. From each speaker,

[1] https://github.com/alumae/voxceleb_weakly_supervised_segments

one video is reserved for the heldout and one for the test set. The official face-verified segments from the corresponding speaker are used for testing speaker identification performance. Top-1 and top-5 accuracies are reported. This is a closed set speaker identification task, meaning that the segments in the development and test sets are guaranteed to correspond to one of the 1251 speakers. Experiments with speaker identification are performed both on the segment level as the recording level. In the latter case, the we aggregate over all segments of the individual videos and making a single decision for the video, as the annotated segments from a single clip are guaranteed to correspond to the same speaker.

For speaker verification, a different training/testing split is used: all speakers whose name starts with an 'E' are reserved for testing. The videos that correspond to those speakers are not used for training the speaker embedding system. Speaker verification is performed at the segment level using a list of trials provided with the dataset.

4.2. Training data relabeling using weakly supervised training

For both speaker identification and speaker verification experiments we used an identical approach: the raw audio data of the videos in the training set, together with the name of the person-of-interest (POI) that the video corresponds to, was used as the training data. We didn't use the provided segment labels in any way. We found that there were some videos that were retrieved for several POIs. In this case, we used all the names as weak supervision for this video.

We used a speaker diarization system trained on Estonian radio broadcast data [12] to segment and cluster the VoxCeleb training data, despite the obvious domain mismatch. The diarization system is based on the LIUM SpkDiarization toolkit [13] and uses BIC clustering [14] followed by CLR-like cluster- ing [15] to find the most likely segment-to-speaker mappings.

The speaker-diarized VoxCeleb training data was used to train an i-vector extractor using Kaldi [16]. We use 600-dimensional i-vectors. The underlying 30-dimensional MFCC features are extracted from wide-band speech signal, with the high cutoff frequency of 7600. Speaker-level i-vectors are computed by length-normalizing the utterance i-vectors, averaging the utterance vectors and finally length-normalizing the average.

The DNN for speaker identification has two hidden layers with a dimensionality of 1024. Dropout layers use a dropout proportion of 0.5 at training time. The DNN was trained for 50 epochs using the described weakly supervised method, with a linearly decreasing learning rate on the training partition of the VoxCeleb data.

Finally, the trained DNN was used to classify the i-vectors of the training data. For all i-vectors which were classified to a certain POI (i.e., not to the unknown speaker class), we looked up the source segments for this i-vector and used the result as the new training data annotation. We didn't impose any constraints during labeling, allowing, for example, an i-vector from a clip to labeled as POI A although the clip corresponds to POI B according to the VoxCeleb data. Also, we allow multiple i-vectors from a single clip to be classified as POI A, although at training time we encourage the DNN to only classify a single i-vector to the clip's POI.

Table 2. *Comparison of the VoxCeleb reference annotations and our annotations generated using weakly supervised learning (all times in hours).*

Length of reference annotations	303.6
Length of our annotations	1173.1
Matched annotations	290.2
Missed in our annotations	13.2
Substituted speaker in our annotations	0.2

Once the relabeling using the weakly supervised model in performed, we can compare the reference annotations of the VoxCeleb dataset (generated using face verification) with our annotations. Table 2 gives some insight to the difference between the two annotations, based on the training set for speaker identification. It can be seen that for 95.5% of the time a speaker is annotated in the reference segmentation, it is also annotated with the matched speaker name in our annotations. 4.3% of the reference speaker annotations are missing in our segmentations and for less than 0.1% of the duration of the reference segmentation, the speakers in the two annotations differ. However, our annotations are almost four times longer than reference annotations.

4.3. Speaker identification

We performed speaker identification experiments using five different systems: two i-vector based systems, trained on either the reference annotations or our annotations, and two x-vector based systems, trained on the two different annotations. Both the i-vectors and x-vector based systems use a backend with LDA/PLDA scoring. We also measured the accuracy of the DNN trained with weakly supervised training that was used for generating our annotations. As in the original VoxCeleb paper, both top-1 and top-5 error rates are given.

For training the x-vector systems, we used a recipe that closely follows the Kaldi x-vector recipe for the VoxCeleb corpus[2]. However, we only use the VoxCeleb1 corpus and don't include the newer VoxCeleb2 [17] corpus that was just being released at the time of writing this paper. We augment [18] the original audio with reverberation, background environmental noise, music and babble noise (all three from the MUSAN corpus [19]), and then take a random subset of

[2]https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb/v2

Table 3. *Top-1 and top-5 segment-level speaker identification error rates (in percentages), using different systems.*

System	Heldout		Test	
	Top-1	Top-5	Top-1	Top-5
DNN, weak sup.	17.86	7.17	18.52	7.60
Ref. annot., i-vec PLDA	12.53	4.28	11.21	3.96
Ref. annot., x-vec PLDA	17.91	7.59	16.99	7.49
Our annot., i-vec PLDA	10.70	4.28	10.58	4.38
Our annot., x-vec PLDA	**8.91**	**3.18**	**8.50**	**3.31**

Table 4. *Top-1 and top-5 recording-level speaker identification error rates (in percentages), using different systems.*

System	Heldout		Test	
	Top-1	Top-5	Top-1	Top-5
DNN, weak sup. [1]	-	-	5.40	1.90
DNN, weak sup.	2.64	1.07	2.08	0.75
Ref. annot., i-vec PLDA	1.24	**0.58**	**0.91**	**0.58**
Ref. annot., x-vec PLDA	1.82	0.91	1.25	0.75
Our annot. i-vec PLDA	**1.16**	0.83	1.00	**0.58**
Our annot., x-vec PLDA	1.24	0.66	1.16	**0.58**

the augmented data so that the amount of augmented data is roughly equal to double the amount of the original data. The architecture of the x-vector system was identical to that in the Kaldi recipe.

Table 3 lists the error rates for segment-level scoring. That is, speaker identification was performed independently on the individual reference segments of the evaluation and test sets. As can be seen, the i-vector and x-vector based systems trained with the segments generated using the proposed method have lower error rates than the systems trained with reference annotations. Furthermore, the x-vector based embeddings (as opposed to i-vectors) result in higher accuracy when trained with our annotations, but deteriorate the accuracy when using the reference training segments.

The direct classification accuracy of the weakly supervised DNN is relatively low. This can be explained by the fact that the DNN is trained using speaker-averaged i-vectors, not i-vectors that correspond to individual segments, and has thus difficulties when dealing with segment-level data that has more variety.

Table 4 lists the speaker identification error rates when scoring is performed on the recording level. Here, the embeddings of segments from individual recordings were averaged and length-normalized. LDA and PLDA transforms were also estimated with respect to the averaged embeddings of the speakers in individual recordings. In this experiment, all systems using LDA/PLDA scoring achieve similar error rates. The differences between the error rates are indeed very small: the heldout and test set both contain 1251 speakers, and the difference between 0.91% and 1.00% error rate reduces to identifying a single speaker correctly or incorrectly. Only using DNN trained in weakly supervised manner directly results in roughly twice more errors than the LDA/PLDA based systems. The first line in the table corresponds to the system described in our earlier work [1]. The main difference of the DNN used in this work with regard to this system is that in [1] we used narrow-band acoustic features and slightly less optimized i-vector training procedure.

4.4. Speaker verification

In speaker verification experiments, we compared four different embeddings: i-vectors and x-vectors trained on either the reference VoxCeleb segments or segments generated using the DNN trained using weak supervision. Note that the training set for speaker identification is different from speaker verification, thus the systems compared here are different from the ones used in the identification experiments, although they share their most important characteristics. The main difference is that the i-vector in the identification experiments are 600-dimensional and in the verification experiments 400-dimensional. This is because we wanted to replicate the Kaldi reference implementation as close as possible.

Table 5 report results in terms of equal error-rate (EER) and the minimum of the normalized detection cost function (DCF) at $P_{Target} = 10^{-2}$ and $P_{Target} = 10^{-2}$. The first row lists speaker verification results reported in the original VoxCeleb paper [2]. Our Kaldi based implementation of this system (second row) gives better results, although the most important hyperparameters should be identical between the systems. Thus, we suspect that difference in performance comes from feature extraction or the details of i-vector extraction. In this experiment, x-vector embeddings trained on our segmentations show advantage, resulting in 17% relative reduction in EER compared to the baseline system (i-vector embeddings trained on reference segmentations). The DET curves corresponding to the four different systems are plotted in Figure 3.

The last two rows of Table 5 list speaker verification results when using the newer and much larger VoxCeleb2 corpus for training the embeddings [3]. The x-vectors trained on the VoxCeleb2 dataset provide a large gain compared to the x-vectors trained on VoxCeleb1 only. At the time of writing this paper, VoxCeleb2 was just being released and we didn't have enough time to conduct experiments on it, mostly because our method needs full audio of the YouTube videos in the dataset, while only the reference audio segments were provided with the official VoxCeleb2 release. Therefore, we had to down-

[3]The result are copied from the recipe available at https://github.com/kaldi-asr/kaldi/blob/master/egs/voxceleb/v2/run.sh

Table 5. *Speaker verification results.*

Training segments	Embeddings	EER	DCF10^{-2}	DCF10^{-3}
Ref	i-vec [2]	8.80%	0.730	-
Ref.	i-vec	5.49%	0.499	0.687
Ref.	x-vec	5.74%	0.511	0.733
Our	i-vec	5.59%	0.466	**0.617**
Our	x-vec	**4.57%**	**0.421**	0.621
Including VoxCeleb2				
Ref.	i-vec	5.33%	0.493	0.617
Ref.	x-vec	3.13%	0.326	0.500

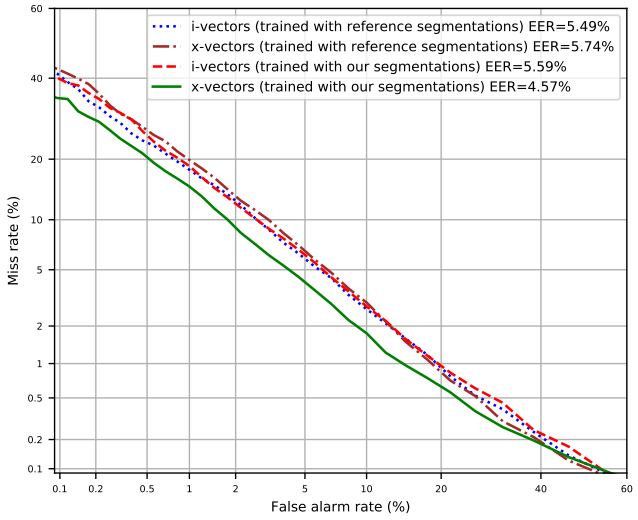

Fig. 3. *DET curves of speaker verification systems using different speaker embeddings.*

load all 1 128 246 videos of the VoxCeleb2 dataset that took around 30 days due to YouTube throttling limits.

4.5. Analysis

Experimental results showed that the annotations generated for the VoxCeleb database using weakly supervised training with recording-level speaker labels yield equal or better results than the official annotations that make use of facial tracking and face verification. The i-vector based identification and verification systems built from the two different annotations performed at a similar level. The x-vector based systems clearly benefited from four times more training data that our annotations provide and resulted in better performance for both identification and verification. This is not surprising, as x-vectors have been shown to require more data than i-vectors for good performance [4].

5. CONCLUSION

In this paper, we presented some improvements to the recently proposed method for training speaker identification models using coarse-grained speaker labels, given at the recording level. We showed that instead of using the trained speaker identification DNN directly, it is beneficial to instead use it for reannotate the whole training data and then use conventional LDA/PLDA based speaker recognition recipes. Experiments on the VoxCeleb dataset showed that speaker segmentation annotations generated using the proposed method, using only the information about speaker-recording correspondence, can yield better results than official segmentations of the VoxCeleb dataset that are generated using face tracking and face verification. The annotations that are generated using weak supervision are almost four times longer than the reference segmentations and thus result in superior x-vector models for both speaker identification and speaker verification.

As future work, we want to apply the same method to the recently proposed VoxCeleb2 database [17]. Furthermore, as our method does not require any face tracking and verification, it is fairly easy to generate entirely new datasets together with speaker annotations, e.g. in order to cover a required set of speakers.

6. REFERENCES

[1] Martin Karu and Tanel Alumäe, "Weakly supervised training of speaker identification models," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018.

[2] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," *Interspeech*, pp. 2616–2620, 2017.

[3] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018.

[5] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *International Conference on Computer Vision*, 2007.

[6] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[7] Ondrej Novotnỳ, Karel Veselỳ, Ondrej Glembek, Oldrich Plchot, Ladislav Mošner, and Pavel Matejka, "BUT system for DIHARD speech diarization challenge 2018," in *Interspeech*, 2018.

[8] Sarthak Yadav and Atul Rai, "Learning discriminative features for speaker identification and verification," *Interspeech*, 2018.

[9] Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al., "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," *Interspeech*, 2018.

[10] Hannah Muckenhirn, Mathew Magimai Doss, and Sébastien Marcell, "Towards directly modeling raw speech signal for speaker verification using CNNs," in *ICASSP*, 2018.

[11] Anna Silnova, Niko Brümmer, Daniel Garcia-Romero, David Snyder, and Lukáš Burget, "Fast variational Bayes for heavy-tailed PLDA applied to i-vectors and x-vectors," arXiv:1803.09153, 2018.

[12] Tanel Alumäe, "Recent improvements in Estonian LVCSR," in *SLTU*, 2014.

[13] Sylvain Meignier and Teva Merlin, "LIUM SpkDiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

[14] Scott Chen and Ponani Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[15] Douglas A Reynolds, Elliot Singer, Beth A Carlson, Gerald C O'Leary, Jack J McLaughlin, and Marc A Zissman, "Blind clustering of speech utterances based on speaker and language characteristics," in *SLP*, 1998.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[17] J. Son Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Interspeech*, 2018.

[18] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017.

[19] David Synder, Guoguo Chen, and Daniel Povey, "MUSAN: A music, speech, and noise corpus," arXiv:1510.08484, 2015.