SPHEREDIAR: AN EFFECTIVE SPEAKER DIARIZATION SYSTEM FOR MEETING DATA

Tuomas Kaseva[1], *Aku Rouhe*[1], *Mikko Kurimo*[1]

Aalto University, Department of Signal Processing and Acoustics [1]

ABSTRACT

In this paper, we present *SphereDiar*, a speaker diarization system composed of three novel subsystems: the SphereSpeaker (SS) neural network, designed for speaker embedding extraction, a segmentation method called Homogeneity Based Segmentation (HBS) and a clustering algorithm called Top Two Silhouettes (Top2S). The system is evaluated on a set of over 200 manually transcribed multiparty meetings. The evaluation reveals that the system can be further simplified by omitting the use of HBS. Furthermore, we illustrate that SphereDiar achieves state-of-the-art results with two different meeting data sets.

Index Terms: speaker diarization, speaker embeddings, segmentation, spherical K-means, silhouette coefficients

1. INTRODUCTION

Speaker diarization answers the question "who spoke and when" [1]. In this process, a given audio stream is segmented into speaker turns: time intervals in which one speaker is speaking. It is determined, which of the speaker turns have the same speaker, but the actual identity (e.g. name) of the speakers is not required. Speaker diarization is a necessary subtask in many different speech applications such as creation of speech corpora, speech translation and speech recognition [1, 2, 3].

Speaker diarization is made difficult by the immense variability in speakers and recording conditions, and the unpredictable and overlapping speaker turns of spontaneous discussion [1, 4]. For these reasons, speaker diarization is still far from solved. In this paper, our main contribution is to propose a novel speaker diarization system which we have made available online[1]. The system consists of three main components which operate on three main tasks in speaker diarization: speaker modeling, segmentation and clustering [1].

The objective of speaker modeling is to embed a given speech utterance in a space which is more suitable for speaker discrimination [5]. Traditionally, this transformation has been performed with either Gaussian Mixture model (GMM) or i-vectors [6, 7]. Recently, also deep learning methods, both metric learning based [2, 8, 9, 10] and classification based [11, 12, 13], have been investigated. These methods have focused on creating neural speaker embeddings which have been shown to outperform i-vectors on many occasions [2, 13, 14]. Furthermore, especially classification based methods have shown great promise also in face verification [15, 16]. Motivated by these works, we choose to apply deep learning in our speaker modeling approach. We develop a novel neural network which learns the speaker embeddings through speaker classification. In this process, the network forces the embeddings to be L^2 normalized, or in other words, spherical. In our experiments, we show that this relatively simple operation has a profound positive impact on the speaker diarization task. Consequently, we name the network SphereSpeaker (SS), and our system SphereDiar.

In speaker diarization, segmentation refers to the task in which audio stream is divided into partitions which can be assigned to a single dominant speaker [1]. This procedure consists of speaker change detection (SCD) and overlapping speech detection (OSD) [1]. Whereas hypothesis testing has been the standard approach in the former [1, 9], Hidden Markov models accompanied with GMMs have been used in the latter [1, 17]. However, just as in speaker modeling, deep learning has recently been very successful in both OSD and SCD [9, 18, 19, 20]. Nevertheless, a segmentation approach which combines both OSD and SCD into a single process has not been proposed, although the connection of OSD and SCD has been well documented in literature [17]. In this paper, we develop such an approach, which we call Homogeneity Based Segmentation (HBS), and investigate its importance for our speaker diarization system. HBS uses deep learning and transforms the segmentation into a binary classification task.

The most popular clustering approach in speaker diarization has been agglomerative hierarchical clustering (AHC) [1, 4, 14, 21]. In addition, approaches exploiting Integer Linear Programming (ILP) [22], Information Bottleneck (IB) [23] and supervised learning [24] have been proposed. In our approach, we choose a slightly different clustering method which is based on using spherical K-means algorithm. This algorithm is essentially the same as K-means but uses cosine similarity as a distance distance metric and has L^2 normalized cluster centers [25]. The choice of the algorithm is based on our preliminary experiments for clustering the speaker embeddings created with SS. However, the algorithm requires the number of cluster centers as an input, which is typically

[1] https://github.com/Livefull/SphereDiar

unknown. Hence, in our method, we create multiple spherical K-means clusterings with a different number of clusters and choose the best clustering based on an empirically found and unsupervised criteria. These criteria are based on using silhouette coefficients [26] which, along with spherical K-means were also found to be beneficial for the clustering process. We call this method Top Two Silhouettes.

We show that our system achieves state-of-the-art results with a challenging dataset consisting of meeting recordings. Furthermore, we illustrate that these results are obtained even without using HBS and that HBS has overall a little significance for our system. As a consequence, our system can then be simplified considerably by excluding segmentation entirely. This is not only convenient but also an interesting discovery since especially OSD has been a prominent research direction in speaker diarization [1, 4, 17, 20].

2. DATA

The meeting corpus is composed of AMI (Augmented Multi-party Interaction) and ICSI (International Computer Science Institute) corpus, both of which consist of audio recordings of different meetings in various sites [27, 28]. Both corpora provide the recordings in multiple different audio formats from which the 16 kHz *Headset Mix* is used in all of our experiments. In order to create speaker diarization labels for a given meeting, we combine both manually generated and automatic speech recognition (ASR) based transcriptions. Unfortunately, complete ASR based transcriptions were not available for all meetings in AMI and ICSI corpus. The meetings which did not include ASR transcriptions were then excluded. These meetings can be found from Table 1. As a result, the number of remaining meetings is 237 consisting of 163 AMI and 74 ICSI meetings.

Each meeting in the meeting corpus is transformed into a sequence of overlapping frames $S = \{\mathbf{s_1}, ..., \mathbf{s_N}\}$, where frames $\mathbf{s_i}$ have a duration of 2s and are extracted every 0.5s. Before this framing operation, all non-speech segments are removed according to the reference transcriptions.

The choices of frame and overlap duration are based on several factors. Firstly, it is necessary that a frame is long enough so that proper modeling of the speaker corresponding to the frame is possible. Secondly, the frame has to be a short enough so that spontaneous speaker changes would not go unnoticed. As a result, a duration of 2 seconds was chosen, which has also been used in [9, 14].

Relatively large overlapping in turn is beneficial for the clustering procedure as it enables more samples for forming the clusters. However, an increase in overlap duration also results in a increase in computing time as the number of frames in S increases. Preliminary experiments illustrated that an overlap duration of 1.5 seconds would then be a suitable compromise.

Since speaker turns change unpredictably in spontaneous discussion, each two-second frame can include speech from multiple different speakers. That is, in general, each speaker speaks for only some percentage of the frame's duration. For each frame $\mathbf{s}$, we compute a quantity we call *homogeneity percentage* $H_{\%}$. It is the highest percentage of frame time covered by a single speaker. The frame's *speaker label* l is this most prominent speaker. Equivalently,

$$l = \arg\max_i |T_i|, \quad H_{\%} = \frac{\max_i |T_{i\neq -1}|}{|T|} * 100\%, \qquad (1)$$

where $T = \{T_{-1}, T_1, ..., T_{n_s}\}$ is a set of transcription labels of $\mathbf{s}$ with T_{-1} corresponding to samples which include overlapping speech and $T_{i\neq -1}$ depicting samples which are assigned to a speaker i.

Table 1. Removed meetings.

AMI	EN2001a, EN2001e, EN2002c, EN2003a, EN2006a EN2006b, IB4005, IS1003b
ICSI	Bmr012

The speaker corpora comprise of four different partitions, LS_{1000} and LS_{2000} which are collected from Librispeech corpus and VC_{1000} and VC_{2000} which are extracted from the Voxceleb2 dataset [3, 29]. The number of speakers in a partition is given as the subscript. To the best of our knowledge, the speakers in each partition are disjoint from the speakers in the meeting corpus. The speech material of each partition consists of frames of 2s duration which are extracted without overlap. The sampling frequency is the same as with the meeting corpus. In the extraction procedure, we use the webRTC speech activity detection (SAD) system [30], since reference transcriptions are not available. The gender distributions and the frame compositions are depicted in Table 2 and 3.

In order to balance the speaker label distributions with the partitions with the same number of speakers, the maximum number of frames per speaker is limited. The limit for the partitions LS_{2000} and VC_{2000} is assigned as 670 whereas the limit for LS_{1000} and VC_{1000} is 1000. The LS_{1000} partition, however, did not include quite as much speech material as VC_{1000}, so the maximum number of frames per speaker is only 764.

3. SPHEREDIAR

The block diagram of SphereDiar speaker diarization system is presented in Figure 1. In this Figure, input S depicts a sequence of $2s$ frames sampled with 16 kHz frequency and the output L the corresponding speaker label sequence. Note that we do not provide SAD in this system. This is by no

Table 2. Gender distribution in speaker corpora.

	Number of females	Number of males	Total
LS_{1000}	500	500	1000
LS_{2000}	987	1013	2000
VC_{1000}	500	500	1000
VC_{2000}	731	1269	2000

Table 3. Frame compositions in speaker corpora.

	Minimum number of frames per speaker	Maximum number of frames per speaker	Total number of frames
LS_{1000}	382	764	654 297
LS_{2000}	341	670	1 204 967
VC_{1000}	838	1000	995 443
VC_{2000}	577	670	1 337 601

means a trivial exclusion since SAD is an essential component in any speaker diarization system [1]. However, when diarization systems are developed, reference SAD labels are often used in order to focus on the actual speaker diarization [17, 20, 21, 31]. This is also the case with the speaker diarization systems against which we compare our system in section 4.

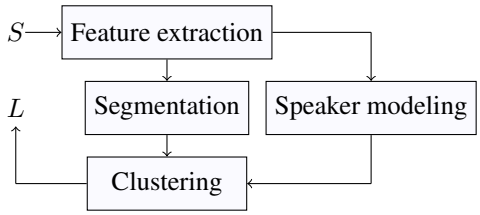

Fig. 1. Block diagram of SphereDiar.

Feature extraction. In the beginning of the diarization procedure, each frame $\mathbf{s}$ in S is converted to $\mathbf{x} \in \mathbb{R}^{201\times59}$, which consists of a sequence of 19 Mel-Frequency Cepstral Coefficients (MFCC), their first and second derivatives, and the first and second derivatives of energy just as in [32]. MFCCs are extracted every 10ms with a 25ms window duration using Librosa [33] and normalized with zero mean and unit variance.

SphereSpeaker. In speaker modeling, each feature sequence $\mathbf{x}$ is projected into a speaker embedding $\mathbf{f}(\mathbf{x})$. The projection is attained by using the neural network depicted in Figure 2 and Table 4. This network is initially designed to predict a class, or in our setting, a speaker identity for $\mathbf{x}$. Consequently, the final layer has a softmax activation function which assures that the output is an N_s dimensional probability distribution, where N_s is the number of classes. The speaker embedding $\mathbf{f}$ is produced in this classification process as the output of the last hidden layer. As a result, the final layer is only used during the training.

The network consists of two main components: a cascade of three bidirectional Long Short-Term Memory (LSTM) neural networks with skip connections which adheres to the architecture of [32] and an embedding layer. In this layer, we assign two conditions on the embedding: $\mathbf{f} \in \mathbb{R}^{1000}$ and $\|\mathbf{f}\|_2 = 1$. The use of L^2 normalization is influenced by the work in [12, 16] whereas the embedding dimension and the overall configuration of the embedding layer are based on our preliminary experiments. The importance of the normalization operation inside the network will be emphasized further in the experiments section where we compare SS with SS*. The latter is otherwise the same network as SS, but does not include L^2 normalization layer. Instead, the speaker embeddings extracted with this network are L^2 normalized externally.

Table 4. Output dimensions of each layer in SphereSpeaker and HBS neural networks.

SphereSpeaker neural network	Output dimensions
Bidirectional LSTM_1	201×500
Bidirectional LSTM_2	201×500
Bidirectional LSTM_3	201×500
Concatenation	201×1500
Embedding layer	1000
Fully connected layer (softmax)	N_s
HBS neural network	**Output dimensions**
Bidirectional LSTM	201×600
Attention layer	201×600
Average pooling layer	600
Fully connected layer (sigmoid)	1

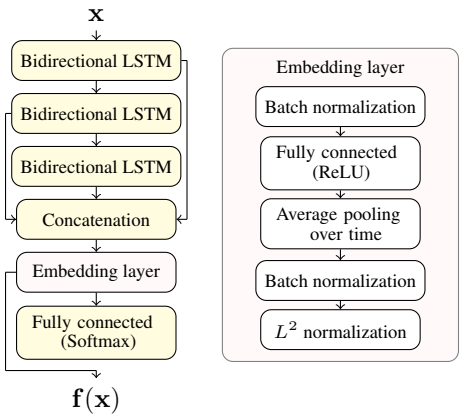

Fig. 2. SphereSpeaker neural network.

Homogeneity Based Segmentation. Segmentation is performed as a binary classification where the formulation of classes is based on the concept of homogeneity percentage. In this approach, our aim is to label each $\mathbf{x}$ as 0, if $H_{\%}$ of the

corresponding frame $\mathbf{s}$ of $\mathbf{x}$ exceeds a given threshold $H_{\theta\%}$ and otherwise as 1. As a result, we call this method Homogeneity Based Segmentation. Ideally, due to the definition of the homogeneity percentage, class 1 consists of frames which include speaker change boundaries and overlapping speech whereas class 0 comprises of frames which can be assigned to a single dominant speaker. Nevertheless, a gray area between classes does exist when homogeneity percentages are close to the threshold $H_{\theta\%}$. Moreover, there is no optimal threshold: we set $H_{\theta\%} = 65\%$ in our experiments as we consider it to be a suitable compromise. The ultimate goal of HBS is to exclude the frames assigned to class 1 from the clustering procedure. The main feature of the HBS is that in theory, it allows performing both OSD and SCD in a single process. Such simplification is yet to be proposed or experimented in speaker diarization.

The class labels are predicted with the neural network illustrated in Figure 3 and in Table 4. The two main components of this network are the bidirectional LSTM neural network which is motivated by the works in [9, 18, 19] and the attention layer which is based on the implementation in [34]. With the former, we use both regular and recurrent dropout and assign both dropouts as 0.2. All other layers are chosen based on our preliminary experiments. The class $h(\mathbf{x}) \in \{0, 1\}$ of each $\mathbf{x}$ is determined based on rounding the output of the network $\hat{h}(\mathbf{x}) \in [0, 1]$ to the nearest integer.

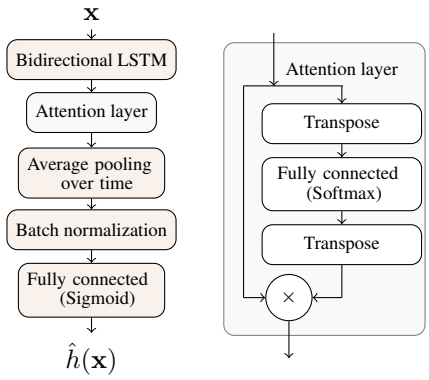

Fig. 3. HBS neural network.

Top Two Silhouettes. After the speaker modeling and segmentation we have obtained a sequence of speaker embeddings F and a sequence of HBS labels H. As a final step, we assign each $\mathbf{f}$ in F with a speaker label. In our approach, this assignment is determined by clustering E, a subset of F consisting of embeddings $\mathbf{f_i}$ with the HBS label $h_i = 0$. The clustering is performed with a novel algorithm which can be divided into two steps: the proposal generation and the optimal proposal determination.

In the first step, E is fitted with multiple different spherical K-means configurations with K ranging from 2 to N_{max}. Here, N_{max} refers to an initial guess of a maximum number of speakers in E. Each configuration is run with R different initializations from which the final configuration is determined based on the run which yielded the highest silhouette score. This score is the average of silhouette coefficients which are computed for each speaker embedding. In this computation, cosine similarity is used as a distance metric. More details of the calculation of the coefficients can be found in [26]. The proposals P_i are then created based on these final configurations.

In the second step, the optimal proposal P_{opt} is chosen. First, the proposals corresponding to the two largest silhouette scores, P_{top-1} and P_{top-2} are recovered. If (i) P_{top-1} has more clusters, or (ii) the silhouette score of P_{top-2} is below a threshold δ, then $P_{opt} = P_{top-1}$. This is a heuristic rule which we have found experimentally and can be interpreted as a further confidence that P_{top-1} is the optimal proposal.

Otherwise, if both (i) and (ii) are unsatisfied, the algorithm deduces that P_{top-2} could also be chosen. As P_{top-2} has then more clusters than P_{top-1}, the algorithm investigates if any of the clusters in P_{top-1} might contain inner clusters. This investigation is performed in a similar fashion as in the first step but for each each cluster in P_{top-1}. The assignment $P_{opt} = P_{top-2}$ is then chosen if for any initialization or cluster, both maximum silhouette value is above δ and a corresponding $K \in \{2, 3\}$. In this condition, the maximum number of inner clusters is restricted to 3 since a higher number would be highly improbable. However, if this condition is not satisfied the algorithm again chooses $P_{opt} = P_{top-1}$.

Algorithm 1: Top Two Silhouettes
Input: Set of speaker embeddings E, a number of initializations R, a maximum number of speakers N_{max} and a threshold δ
Output: Proposal $P = \{L, C\}$.
Steps:

1. Initialize $K = \{2, ..., N_{max}\}$ and $s = \{0, ..., 0\}, |s| = |K|$
2. **for** $r = 1$ to R **do**
   for $i = 1$ to $|K|$ **do**
   $\phi(K_i, E) \rightarrow L_i \rightarrow \upsilon(L_i, E) \rightarrow \hat{s}_i$
   if $(\hat{s}_i > s_i) \rightarrow s_i = \hat{s}_i$.
3. Find largest and second largest silhouette scores s_{top-1} and s_{top-2}, respectively.
   If not top-2 > top-1 $\wedge s_{top-2} > \delta$)
   $\rightarrow$ **return** L_{top-1}, C_{top-1}
4. Repeat step 2 for each $E_j \in E = \{\mathbf{f_i} \mid l_i = k \in L_{top-1}\}$ In the process, for any j, r:
   If $(\max_i \upsilon(L_{ij}, E_j) > \delta \wedge K_i \in \{2, 3\})$
   $\rightarrow$ **return** L_{top-2}, C_{top-2}
5. **return** L_{top-1}, C_{top-1}.

The labels L for F are then generated using associated cluster centers C_{opt} of P_{opt}. As the proposals corresponding to the two largest silhouette scores are central to the algorithm, we have named it Top Two Silhouettes. In the experiments section, we demonstrate the validity of this algorithm by comparing it with Top Silhouette (TopS), which is essentially the same as Top2S but always assigns $P_{opt} = P_{top-1}$.

Top Two Silhouettes is described more formally in Algorithm 1. In this description, spherical K-means is denoted with ϕ and the calculation of the silhouette score with a variable v. Moreover, instead of two steps, the description consists of three main steps consisting of the calculation of the silhouette scores, the evaluation of the conditions (i) and (ii) and the possible inner cluster search.

4. EXPERIMENTS

4.1. Experimental setup

Evaluation metric. All experiments are conducted using the same evaluation metric called diarization error rate (DER) [1]. In general, DER consists of SAD related errors (false alarm and false rejection) and speaker errors which, in our case, can be interpreted as a clustering errors between reference and predicted speaker labels [1]. However, since we have performed SAD on all meetings in the meeting corpus as a preprocessing step, the computation of DER simplifies to a calculation of the speaker error which we compute with Hungarian algorithm [35]. Furthermore, when calculating DER, we consider only labels corresponding to the frames which have $H_{\%}$ above the threshold $H_{\theta\%} = 65\%$ unless explicitly mentioned otherwise.

Neural network training and evaluation. We train 10 models in total: 8 for speaker modeling and 2 to be used for segmentation. The first eight models are trained using SS and SS* and four different training and evaluation set splits. The splits are generated from each partition in the speaker corpora by choosing randomly 45 frames from each speaker for testing and leaving the rest for training.

The last two models both use HBS neural network, but are trained solely using the meeting corpus with two different evaluation sets: AMI_{eval} which is a same as in [21] or $ICSI_{eval}$ consisting of 9 ICSI meetings [1]. In both cases, all other meetings in the meeting corpus are reserved for training. Moreover, only frames which have $H_{\%} = 100\%$ (labeled as 0) and frames with $H_{\%} \leq 65\%$ (labeled as 1) are used in training and evaluation. This choice is based on ensuring proper discrimination between classes that we found beneficial in our preliminary experiments.

All 10 models are trained using Keras deep learning library [36]with batch size 256, for 45 epochs. We use the cross entropy as a loss function and using Adam [37] optimizer. When training the last two models, we also weight class 1 twice as much as class 0 in order to balance the class distributions.

Clustering parameters. We assign $R = 50$ and $N_{max} = 11$ in all experiments. We choose to set R this high since spherical K-means has a tendency to converge to a local maximum [25]. The value of N_{max} is selected to exceed the highest possible participant number, 9, of the meetings in the meeting corpus. In addition, we set $\delta = 0.1$, which we attained by conducting a grid search on a clustering development set $Clust_{\text{dev}}$ of 12 meetings extracted from the meeting corpus[1]. This set is disjoint with both AMI_{eval} and $ICSI_{eval}$. In the grid search, we evaluated each threshold using DER, did not use HBS and performed speaker modeling with SphereSpeaker trained with VC_{1000}.

4.2. Results

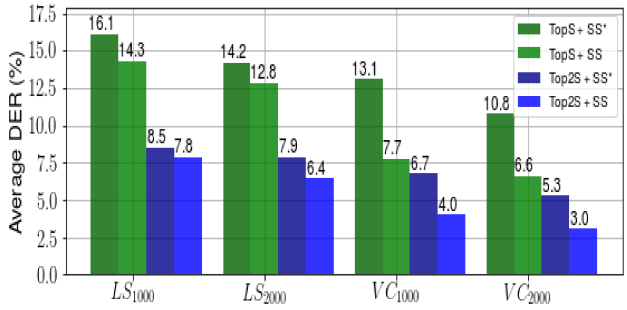

Fig. 4. Average DER over 225 meetings from the meeting corpus with different SphereDiar configurations which omit HBS.

In Figure 4, we visualize speaker diarization results with 225 meetings from the meeting corpus that are disjoint with $Clust_{\text{dev}}$. These results are obtained using all possible SphereDiar configurations introduced in this paper but without using HBS ($h_i = 0, \forall i$) as most of the meetings have been used in HBS training. The results illustrate that SS outperforms SS*, especially when these neural networks are trained with Voxceleb2 partitions, and that Top2S performs markedly better than TopS. Moreover, the results show that both the increase in the number of training speakers and the use of Voxceleb2 partitions over Librispeech partitions are preferable in speaker modeling training. The best configuration is attained by combining SS trained with VC_{2000} and Top2S and it achieves 3% average DER over the 225 meetings.

The results in Table 5 show that our HBS system fails to benefit the speaker diarization task. In the experiments which were briefly discussed with neural network training and evaluation, the HBS system achieved mean average precision of 0.953 with AMI_{eval} and 0.935 with $ICSI_{eval}$. Clearly, these scores were not high enough to make HBS beneficial for speaker diarization.

Table 5. Average DER (%) over different evaluation sets and HBS setups with the best SphereDiar configuration.

Segmentation	AMI_{eval}	$ICSI_{eval}$	225 meetings
-	2.4	2.9	3.0
HBS	3.5	4.8	-
Optimal HBS	2.0	2.5	2.8

However, the results also depict that even when using optimal HBS, which assigns h_i based on reference HBS labels, no significant improvement for the task is attained. This remark is especially distinct when the evaluation set consists of all of the 225 meetings. Interestingly, these results imply that our system is neither too dependent on OSD or SCD which have been previously shown to be important factors in speaker diarization [1, 4]. We hypothesize that this outcome is due to two reasons: a good generalization ability of the speaker embeddings and relatively low significance of HBS for the Top2S algorithm. Especially the latter can be emphasized, since HBS labels are only utilized to exclude the embeddings from the clustering procedure but not in any other manner. For example, the labels could have also been used in the initialization of spherical K-means. Nevertheless, based on the results in Table 5 we can deduce that SphereDiar achieves good results even without OSD or SCD.

Table 6. Average DER (%) comparison.

Test set	Previous best	Ours ($H_{\theta\%} = 55\%$)
AMI_{eval}	4.8 [21]	**3.6**
ICSI subset	13.1 [38]	**4.5**

In Table 6, a comparison between the best SphereDiar configuration and two other speaker diarization systems which have obtained top scores on AMI and ICSI subsets in the literature is provided. These systems include a state-of-the-art i-vector based speaker diarization system [21] and the ICSI RT07s speaker diarization system, which uses both MFCCs and deep learning based features [38, 39]. The average DER for both systems has been calculated from the segments which do not include overlapping speech and by using a forgiveness collar around speaker change boundaries [21, 38]. With [38], this collar is ± 0.25 seconds, whereas [21] uses the collar of ± 0.5 seconds. As was mentioned, the DER scores for both systems have been attained using reference SAD labels.

The computation of DER for SphereDiar is based on using the frames which have homogeneity percentages above the threshold $H_{\theta\%} = 55\%$. Due to the formulation of the percentage, this means that virtually all overlapping speech is removed from the DER calculation. Furthermore, decreasing the $H_{\theta\%}$ from 65%, which was used previously, to 55%, can be interpreted as shrinking the collar around speaker change boundaries. This decrease allows the average DER comparison to be as fair as possible since any further decrease in the value of $H_{\theta\%}$ results in severe difficulties of labeling the frames accurately. Consider, for instance, the example given in subsection 2.1.5. If a frame would have $H_{\%} = 50\%$, and would contain two speakers without any overlapping speech. Then, the speaker label of this frame could not be determined.

The results illustrate that our system is able to outperform the systems in [21, 38]. Our result is particularly better when comparing to [38] but we admit that our system has been trained with Voxceleb2 which was not available at the time for [38]. However, the system in [21] has been trained with a very similar data as ours, using Voxceleb [2] and other relevant datasets, but our result is still better. Moreover, as we do not use HBS in the comparison, our domain adaptation is only based on 12 meetings in $Clust_{dev}$. This is significantly less than used in either [21] or [38] and further emphasizes the generality of our system.

5. CONCLUSIONS

This paper proposed a novel speaker diarization system SphereDiar. The system includes two neural networks and one clustering algoritm: SphereSpeaker neural network for speaker embedding extraction, HBS neural network for segmentation and Top Two Silhouettes for clustering. In our experiments, we focused on evaluating the system with 225 meetings and illustrated that the system could be simplified by excluding HBS. Using the best system configuration, we achieved average DER over the meetings as 3%. We compared our system with two state-of-the-art speaker diarization systems and showed that the results obtained with our system were better.

Nevertheless, the system still suffers from deficiencies. Firstly, the dimension of the speaker embeddings is relatively large which slows clustering. Secondly, Top2S does not yet have any proper theoretical foundation. Furthermore, this algorithm is also not very suitable for situations were only few frames for each speaker can be attained. Finally, we have not presented any methods for SAD. In future work, we would like to address each of these shortcomings.

6. ACKNOWLEDGEMENTS

We would like to thank Anja Virkkunen and Stig-Arne Grönroos for their helpful comments. This work was supported by the European Unions Horizon 2020 research and innovation programme via the project MeMAD (GA780069). Computational resources were provided by the Aalto Science-IT project.

7. REFERENCES

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[2] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[3] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[4] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. INTERSPEECH*, 2018, pp. 2808–2812.

[5] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[6] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, 2006.

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

[8] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[9] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[10] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[11] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification." in *ICASSP*, vol. 14. Citeseer, 2014, pp. 4052–4056.

[12] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *Submitted to ICASSP*, 2018.

[14] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 1.

[16] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[17] S. H. Yella and H. Bourlard, "Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1688–1700, 2014.

[18] R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Interspeech 2017*. ISCA, 2017.

[19] G. Hagerer, V. Pandit, F. Eyben, and B. Schuller, "Enhancing LSTM RNN-based speech overlap detection by artificially mixed data," in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.

[20] J. T. Geiger, F. Eyben, B. Schuller, and G. Rigoll, "Detecting overlapping speech with long short-term memory recurrent neural networks," in *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*, 2013.

[21] M. Maciejewski, D. Snyder, V. Manohar, N. Dehak, and S. Khudanpur, "Characterizing performance of speaker diarization systems on far-field speech using standard methods," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5244–5248.

[22] M. Rouvier and S. Meignier, "A global optimization framework for speaker diarization," in *Odyssey 2012*, 2012.

[23] D. Vijayasenan, F. Valente, and H. Bourlard, "Agglomerative information bottleneck for speaker diarization of meetings data," in *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2007, pp. 250–255.

[24] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," *CoRR*, vol. abs/1810.04719, 2018.

[25] S. Zhong, "Efficient online spherical K-means clustering," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 5. IEEE, 2005, pp. 3180–3185.

[26] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[27] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005, p. 100.

[28] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The ICSI meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–I.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[30] A. Johnston, J. Yoakum, and K. Singh, "Taking on WebRTC in an enterprise," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 48–54, 2013.

[31] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 402–406.

[32] G. Wisniewksi, H. Bredin, G. Gelly, and C. Barras, "Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization," in *Proc. Interspeech*, 2017.

[33] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[34] P. Rémy, "Keras Attention Mechanism," https://github.com/philipperemy/keras-attention-mechanism, 2017.

[35] O. Galibert, "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech." in *INTERSPEECH*, 2013, pp. 1131–1134.

[36] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12, 2014.

[38] S. H. Yella and A. Stolcke, "A comparison of neural network feature transforms for speaker diarization," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[39] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," in *Multimodal Technologies for Perception of Humans*. Springer, 2007, pp. 509–519.