Large Scale Speaker Diarization for Long Recordings and Small Collections

Marijn Huijbregts and David van Leeuwen, *Member, IEEE*

Abstract

Performing speaker diarization of very long recordings is a problem for most diarization systems that are based on agglomerative clustering with an HMM topology. Performing collection-wide speaker diarization, where each speaker is identified uniquely across the entire collection, is even a more challenging task. In this paper we propose a method with which it is possible to efficiently perform diarization of long recordings. We have also applied this method successfully to a collection of a total duration of approximately 15 hours. The method consists of first segmenting long recordings into smaller chunks on which diarization is performed. Next, a speaker detection system is used to link the speech clusters from each chunk and to assign a unique label to each speaker in the long recording or in the small collection. We show for three different audio collections that it is possible to perform high quality diarization with this approach. The long meetings from the ICSI corpus are processed 5.5 times faster than the originally needed time and by uniquely labeling each speaker across the entire collection it becomes possible to perform speaker-based information retrieval with high accuracy (mean average precision of 0.57).

Index Terms

speaker diarization, speaker detection, large scale diarization, collection wide diarization, information retrieval

I. INTRODUCTION

The goal of speaker diarization is to automatically segment an audio recording into speaker homogeneous regions, and link these together. With the identity of each speaker and even

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Marijn Huijbregts and David van Leeuwen are with the Centre for Language and Speech Technology, Radboud University Nijmegen, the Netherlands, m.huijbregts@let.ru.nl, d.vanleeuwen@let.ru.nl.

the number of speakers unknown, it is the task of a diarization system to anonymously label each speaker in the recording and answer the question: ‘Who spoke when?’ [1].

Diarization information is useful as pre-processing step for various tasks. In Automatic Speech Recognition (ASR) for example, diarization information can be used for adapting speaker independent acoustic models to each speaker occurring in the data. In the international research project BATS¹ we are interested in using diarization information for enhancing search in multimedia collections so that it becomes possible to search for speech fragments of particular speakers. In order to do this, we do not only need to perform diarization on each recording of the collection, but we also need to determine for each found speaker if he or she is also present in one of the other recordings. We need to perform ‘large scale speaker diarization’: speaker diarization across multiple recordings.

One naive way of approaching this challenge is to concatenate the recordings of the collection into a single, very long recording and perform speaker diarization on it. If possible, this method would ensure that each speaker obtains a unique label for the entire collection. This approach is impractical because of two reasons. First, most diarization systems will not be able to process such very long recordings because of memory or computational scaling issues and second, speakers might be misclassified because diarization systems typically do not compensate for *inter-session* and *intra-speaker variability*. Diarization errors will be made because of varying recording conditions between sessions (different microphones or room acoustics, inter-channel variability) and also because speech characteristics from one speaker typically varies across recordings (because of age or perhaps a cold, intra-speaker variability) [2].

In recent years we have developed a speaker diarization system that performs state-of-the-art at the NIST speaker diarization benchmarks [3], [4]. Our system is based on agglomerative clustering where the number of initial clusters depends on the length of the recording. Unfortunately, because all combinations of two clusters need to be compared at each clustering step, this means that computational efforts increase more than linearly with the length T of the recordings (in fact with $O(T^3)$ [5]). The system is able to process recordings of up to one to two hours, but processing longer recordings becomes very time consuming. Processing entire collections in a single run becomes infeasible this way.

Not many radio or television broadcasts are longer than two hours, but we did encounter

¹<http://www.ibbt.be/en/project/bats-0>

some cases where running our system became impractical because of the length of the recordings. For example, when trying to process the raw recordings of a collection of interviews with Dutch war veterans. These interviews took two hours on average, with some lasting up to four hours.

In this paper we will propose a method that is useful for both diarization of long recordings and also for performing diarization across entire collections of recordings (large scale speaker diarization). In our approach we cut-up long recordings in smaller pieces, referred to as chunks, and perform diarization on each of these chunks. We then need to connect the chunks.

In order to connect the chunks, we try to determine for each speaker from the various chunks, if he/she is -or is not- the same speaker as in one of the other chunks. This is essentially the same problem that is being addressed in the field of speaker detection. We therefore apply a speaker detection system for glueing together the pieces. This approach is convenient because the detection system is developed to recognize speakers even if the recording conditions or the conditions of the speaker vary.

In the experimental section of this paper we will first focus on the performance of the proposed algorithm, but we will also investigate a possible application of large scale speaker diarization of small collections: speaker based information retrieval.

This paper is organized as follows. First, we will discuss related work. In the following section we will discuss our approach for diarization of long recordings and small collections and then describe the diarization and speaker detection systems that we will need for this. In Section V we will discuss the audio collections used in the remainder of this paper. In Section VI we will describe the experiments and in Section VII we will explain what needs to be done to extend our method for bigger, real-life, collections.

II. RELATED WORK

Our goal of linking *all* speech of *all* speakers in a multimedia collection is closely related to two other clustering tasks: speaker tracking [6], [7] and speaker linking [8].

In speaker tracking, the task is to find spoken segments of a particular speaker for which some training material is given. This has both aspects of diarization, as the audio stream needs to be segmented first, and of operation on a collection of recordings. It has as such been performed in both the 1999/2000 editions of the NIST SRE for 2-wire telephone conversations and the French ESTER 2005 evaluation campaign for broadcast news. The task and evaluation

measures were cast in a detection and retrieval framework, respectively, and can be compared to a known item retrieval task in text retrieval. Most speaker tracking systems solve the task by performing speaker diarization followed by speaker detection [9], [10]. Although this problem is similar to our approach for large scale diarization, there are two important differences. First, we try to link the speech of *all* speakers in a collection, not just the speech of a-priori known speakers. And second, in order to do this we not only apply speaker detection on (impure) speaker clusters, we also train the speaker detection models on these clusters. Note that it is possible to combine the speaker tracking and large scale diarization strategies: once a large part of a collection is diarized (across the collection), we could from then on perform speaker tracking on new recordings using the automatically obtained speaker detection models.

Another task that is closely related is speaker linking [8]. For this task, unlike for our problem, the speaker segmentation is assumed to be known, but except for this restriction the task is the same: link together *all* speech of *all* speakers in a multimedia collection. In some way speaker linking occurs in NIST Speaker Recognition Evaluation (SRE) tasks of “2-wire” training, where the common speaker in three conversations must be found for building a proper speaker model. In this task an additional diarization task exists because the recordings are made with both speakers of the conversation on the same channel. However, in that specific task, there are typically three conversations that must be linked, where as in speaker linking and large scale diarization the aim is to be able to process hundreds or thousands of speakers. Even with promising new techniques for diarization of conversations between two speakers [11], linking vast amounts of speakers is an unaddressed problem.

A quite different approach of uniquely labeling each speaker in a collection is proposed in [12] where it was shown that, by making use of the structure of broadcast news speech and automatic speech recognition, it is possible to automatically obtain the names of most speakers. This method can be used for broadcast news collections, but not for any given multimedia collection (such as collections of meetings). It would be very interesting to combine this technique with large scale diarization so that if a name *is* automatically obtained in a news show, the true identity of that speaker is also automatically known for other recordings. Finally, a study that concentrates on diarization of *short* recordings is found in [13]. In this study it is proven for short recordings that, in order to obtain the best results, the number of initial clusters should depend linearly on the amount of speech in the recording. In our studies, we have observed the same for long recordings and even for the recordings of average length in the NIST diarization benchmarks [5].

III. DIARIZATION OF LONG RECORDINGS AND SMALL COLLECTIONS

In this section we will describe our method of performing diarization on long recordings and on small collections. We will first discuss the procedure and then we will describe the diarization and speaker detection systems in-depth. Finally, we will calculate the computational improvement that we gain by applying the procedure.

A. Procedure

The procedure consists of five steps. Figure 1 depicts the procedure. First, speech activity detection (SAD) is performed [14]. Next, the speech found by SAD is cut in evenly sized pieces, *chunks*, of fixed maximum duration. In the third step, diarization is performed within each chunk. Fourth, for each speaker cluster generated by the diarization system, a speaker detection model is trained and the data of all speakers is scored against each speaker model (full scoring matrix). In the final step it is determined, using agglomerative clustering, which speakers from different chunks should be linked together, under some heuristic boundary conditions.

Fig. 1. The five steps of our procedure for diarization of long recordings. First SAD is performed. Next, the recording is cut-up in multiple chunks. Each chunk is diarized and speaker detection models are trained for each found speaker cluster (steps 3 and 4). Finally the scores from the speaker detection models are used to link the speech clusters across chunks.

SAD: Before speaker diarization can be performed, all speech regions need to be found and all non-speech fragments need to be discarded. The speech activity detection step is performed with a system that we originally developed for RT07s in collaboration with ICSI [14]. The SAD component finds all speech regions in two steps: first, using a bootstrapping speech/non-speech detection an initial segmentation is created and models for speech, silence and audible

non-speech are generated. In the second step these models are applied to generate the final speech/non-speech segmentation. The SAD component processes the audio in fragments of ten minutes each and because of this, is able to handle long recordings without scaling issues.

Chunking: After SAD, the recording is cut-up in evenly sized chunks of duration T_c . Note that this does not mean that each chunk is of the exact same length; they only contain the same amount of speech. In the experiments section we will investigate what the minimum value of T_c is so that the diarization system is still able to perform adequately.

Diarization: is done with the system that we developed for the NIST speaker diarization benchmarks. We will describe this system in Section III-B. We use the system directly out-of-the-box without adjusting it in any way in order to show the general applicability of the diarization system.

Speaker enrollment: For each speaker cluster that is found by the diarization system, a speaker detection model is trained (this is called *speaker enrollment*). After enrollment, the data of all speakers is scored against each speaker model. The resulting matrix of scores is used in the final step to link the speakers across chunks. In preliminary experiments we have seen that the absolute scores for each speaker model can vary considerably², perhaps because the enrollment data is not that well balanced as for regular detection tasks. Additionally, conventional normalization methods did not work, because the normalization models were of quite a different domain (see below). We solve this problem by normalizing each score with the score of the target speaker on its own data. Each log-likelihood ratio $S(M_x, D_y)$ where M_x is the model of speaker x and D_y is the data of speaker cluster y , is normalized as follows:

$$S_{\text{norm}}(M_x, D_y) = \frac{S(M_x, D_y)}{S(M_y, D_y)} \quad (1)$$

This means that, provided that each model will score the highest on its own training data, all normalized scores will be smaller than one.

The speaker detection system that we use for this step, developed for the Evalita'09 benchmark, is described in Section III-C. We use this system without adjusting it in any way. Because we do not have appropriate data available to train a new Universal Background Model (UBM), train normalization models and to perform Joint Factor Analysis (JFA), we

²These experiments were conducted during development on Dutch broadcast television recordings for which no reference transcriptions are available.

use the models trained for the Evalita'09 benchmark. This is not ideal for several reasons. First, we have gender specific models, but in this context we do not know the gender of each speaker. For our experiments we decided to simply use one of the two gender models for all speakers (the male models). Second, because the wide band data we are focussing on in this research is quite different from the original Italian telephone data in Evalita'09, the conventional normalization method does not work properly. Therefore, we will apply an alternative normalization method described below.

Linking: In the previous step a matrix is generated with scores of all speaker detection models on the available data of each speaker. This matrix will be used to link the speakers across chunks³. We will apply agglomerative clustering for which we will need a distance measure to determine the distance between each chunk and we will need a stopping criterion in order to prevent over-clustering.

Because the previous step provided scores for each speaker/data combination but no distances, we first translate the scores into distances. We define a distance D by subtracting each score from the maximum score (which is one):

$$D(M_x, D_y) = 1 - S_{\text{norm}}(M_x, D_y) \quad (2)$$

For further normalization of this distance measure we use the following heuristics. We assume that after diarization, each individual chunk contains unique speakers. We further assume that the diarization system will not output more than one cluster for each speaker. We can use this assumption to our advantage in two ways. First, linking clusters is only needed across chunks. Speakers from the same chunk can be excluded from linking. This saves computational effort and prevents the system from making unnecessary mistakes. Second, if all speakers within a chunk are unique, we can use the other speakers in the chunk to normalize the score of each speaker. We normalize the score by the minimum distance of the target speaker (M_x) with the other speakers (S) from the chunk:

$$D_{\min}(M_x) = \min_{S \neq x} D(M_x, D_S)$$

$$D_{\text{norm}}(M_x, D_y) = \frac{D(M_x, D_y)}{D_{\min}(M_x)} \quad (3)$$

³Although this is essentially clustering, to avoid confusion with the term clustering for diarization, we will refer to this process as chunk *linking*

We now have defined the distance of each model to each data set. The distance of each model to its own training data will be zero and because of the normalization in Equation 3, the distance of each model to speaker data of which we assume is from another speaker will minimally be one. After this normalization step we now define the distance between speakers (speaker models) as follows:

$$D(M_x, M_y) = \sqrt{D_{\text{norm}}^2(M_x, D_y) + D_{\text{norm}}^2(M_y, D_x)} \quad (4)$$

We have defined the minimal distance of each model with data of a (known) different speaker as one, and therefore we set the stopping criterion at distance threshold $\sqrt{2}$. In the experimental section (Section VI) we will investigate the usefulness of both the normalization method and the stopping criterion.

Small collections: The above procedure can easily be extended from single long recordings to an entire collection. In the case of collections, the first three steps are performed for each recording and the final two steps (speaker detection and linking) are performed on all chunks across all recordings. Of course the matrix of speaker detection scores will be considerably bigger than for single recordings, but for relatively small collections this is no problem. By chunking, the diarization time per chunk is $\mathcal{O}(T_c^3)$ and for all chunks (the amount of which is T/T_c) the diarization time scales as $\mathcal{O}(TT_c^2)$, where T is the total duration of the recording(s). The speaker detection scoring scales with $\mathcal{O}(T^2)$, assuming that the number of speakers in a chunk scales with chunk-duration T_c .

B. Diarization system

For this research we will use the diarization system that we have developed for the NIST speaker diarization benchmarks. NIST has organized such benchmarks for the meeting domain since 2004 [15]. At each benchmark, diarization systems are evaluated for a number of audio recording conditions. The primary evaluation condition allows the use of audio recorded from multiple distant microphones. We will use our multiple distant microphone system for this paper⁴. This speaker diarization system is based on the algorithm originally described in [16]. It consists of three main components: feature extraction, speech activity detection

⁴Note that for comparison with the experiments in this paper where only single sources are available, we do not make use of the delay feature stream that is normally part of our system

(see Section III-A) and speaker diarization. An extensive description of these components can be found in [3], [4].

The diarization component is based on Hidden Markov Models (HMM) with Gaussian Mixture Models (GMM) as probability density functions. For feature extraction, the first nineteen Mel Frequency Cepstral Coefficients (MFCC) are extracted using standard speech processing procedures [5]. Each speaker is modeled by a string of states that share a single GMM. Initially a high number of models is placed in parallel in the HMM and by using agglomerative clustering, the number of models is automatically reduced until the optimal number of speakers is reached. The final speaker segmentation is obtained by performing a Viterbi search on all audio that contains speech. All audio that is processed by the same model during this alignment is grouped together as speech from one speaker. By using a string of 250 states to model each speaker (instead of a single state), a minimum duration of each speech segment is guaranteed. The merging and stopping criteria that are needed for agglomerative clustering are based on the Bayesian Information Criterion (BIC) [17].

The number of models (GMM strings) that the diarization system is initialized with, depends on the amount of speech in the recording. For a recording that contains S speech frames (of 10 ms each), the initial number of clusters is:

$$N = \min\left(N_{\max}, \left\lceil \frac{S}{S_{\text{init}}} \right\rceil\right) \quad (5)$$

where $S_{\text{init}} = 4000$ and, to prevent an unreasonably high computational load, N is not allowed to be larger than $N_{\max} = 40$. Note that this is already the case for recordings containing more than half an hour of speech.

The initial number of clusters is set dependent on the amount of speech in the recording because if the initial clusters are trained with too much data, they may become too similar to each other and clustering will fail [5].

The N models are initialized by linearly segmenting the available speech in N consecutive segments, and performing several Viterbi decoding / resegmentation iterations.

C. Speaker detection system

Speaker detection is the task of determining whether or not two segments of speech originate from the same speaker. Often one of the segments, the training segment, is relatively

long (up to several minutes of speech) while the other segment, the test segment, typically is a lot shorter (down to a few seconds only).

Over the years several benchmarks have been organized. Since 1996 NIST organizes speaker detection benchmarks annually and in 2009, the Evalita'09 speaker detection benchmark, a benchmark for Italian telephone speech was conducted. We have participated the Evalita'09 benchmark with a Joint Factor Analysis (JFA) based system. We will describe this system briefly below. An in-depth description is provided in [18].

Our Joint Factor Analysis (JFA) based speaker detection system consists of Universal Background Model (UBM) training and JFA itself. For UBM training we used the Shout toolkit [5] and we used the JFA cookbook⁵ developed by Ondrej Glembek and others at Brno University of Technology which is based on [19] for the joint factor analysis, developed during a JHU workshop in 2008. The system is gender specific, but because in this case we do not know the gender of each speaker, we simply use the male speaker models for all speakers. For SAD we use the method described earlier in Section III-A.

UBM training: is carried out using features with 12 Mel-Frequency Cepstral Coefficients (MFCC), energy and the deltas of these thirteen coefficients. A UBM is trained using all available male Evalita'09 UBM data. The UBM (a GMM) is incrementally trained up to 1024 Gaussians, doubling the number of Gaussians at each iteration. After training the UBMs, the zeroth and first order sufficient statistics are calculated for all Evalita'09 audio segments. These statistics are used by the JFA component for training of the eigenchannels, eigenvoices and residuals.

Training the eigenvoices, eigenchannels and residuals: Similar to UBM generation, JFA is performed for this research for male speech data only. The first step in JFA is to determine the eigenvoices, eigenchannels and residual matrices (respectively denoted with v , u and d). For training of the eigenvoices and the residuals, the Evalita'09 UBM set is split into two. The first 25 speakers (with identities 1 to 25) are used for training the eigenvoices and the remaining 5 speakers are used to train the residuals. The eigenchannels are trained on the entire Evalita'09 UBM data set. We used 50 eigenchannel factors and 20 eigenvoice factors. Because of the relatively small data set, using more than 20 eigenvoice factors did not improve the system results on the Evalita'09 development trial set.

⁵<http://speech.fit.vutbr.cz/en/software/joint-factor-analysis-matlab-demo>

IV. COMPUTATIONAL IMPROVEMENT

The computational improvements of the algorithm described in the previous section make it possible to perform diarization of long recordings. Next we will illustrate the difference between the original algorithm and our new approach by calculating the number of merging steps, the most computationally intensive step, that both algorithms require. In Section VI-C we will measure the actual processing time needed for both the original system and the proposed system.

At each clustering iteration, the diarization system needs to train a model for each combination of two speaker models, calculate BIC scores for each of these combinations, choose the models to merge and refine all the models in the system by performing a number of re-alignment and re-training iterations. By far most time is spent on the first two steps: for each combination of two existing models train a new model and perform BIC computation. The number of times that such a new model needs to be trained and BIC needs to be computed depends on the initial number of clusters i and on the final number of clusters f :

$$N(i, f) = \sum_{k=f}^i \binom{k}{2} \quad (6)$$

For a recording that contains 25 minutes of speech, the diarization system will start with 38 initial clusters (using eq. 5). If the recording contains 4 speakers (the final number of clusters), the system will need to train a new model and perform BIC 9135 times. The system only needs to perform the merge and model-refining steps 34 times (number of clustering iterations). This explains why most computational efforts are taken by training models and performing BIC.

If we apply our procedure for diarization of long recordings, a recording with 25 minutes of speech would be cut-up in three chunks of 8.7 minutes of speech each. Each chunk would start with 13 initial clusters. If each chunk merges the initial models until 4 models are left, the system needs to perform $N(13, 4) = 360$ train/BIC steps (eq. 6). This means that only $3 \times 360 = 1080$ calculations are needed in total. This is 8.45 times less than when using the original diarization algorithm.

We made this calculation to give an impression of the computational improvement that can be obtained by using the proposed procedure. In order to demonstrate the real, total computational improvement, we need to take into account the entire system including the

speaker detection system. In Section VI-C we will measure the actual processing time needed for both the original system and the proposed system.

V. THE AUDIO COLLECTIONS

We have carried out large scale diarization experiments on three different audio collections. We have selected the collections on the criteria that 1) they contain recordings that are longer than typical NIST rich transcription diarization evaluation data, 2) they contain re-occurring speakers across different recordings, and 3) speaker metadata is available at the turn level, and forced-aligned speech/non-speech data is available can be computed.

For some of the collections described here, it is still possible to perform diarization with the conventional method because the individual recordings are longer than normally, but not too long for the system to process at once. It is therefore hard to define a specific maximum recording length for which normal diarization can still be performed and a minimum length for which large-scale-diarization should be performed. In the experiments section we will show that even if a recording can be processed with conventional diarization, the large-scale-diarization method is computationally less expensive.

A. Interviews with school teachers

We use a collection of interviews with school teachers from the Spoken Dutch Corpus (component B, CGN) [20] for our first set of experiments. The collection consists of 80 recordings in Dutch and 80 recordings in Flemish. The average speech length (audio length without silence) is 13 minutes for the Dutch recordings and 18 minutes for the Flemish recordings. The longest recordings are over 30 minutes long while the shortest recording is 11 minutes long.

The recordings in the collection are longer than most meeting excerpts in the NIST rich transcription diarization benchmarks for which the diarization system is developed and tuned. Although none of the recordings are too long to be diarized at once, it is possible to follow our approach and cut the recordings into shorter chunks for diarization and investigate the efficiency of the approach.

The interviewer is the same person in all interviews. This makes it possible to test if we can track the interviewer across the collection with the diarization approach for small collections (see Section III-A).

The recordings are manually annotated at the utterance level. However, for accurate diarization performance measurement, we need to know exactly when each word is pronounced. We have generated such an accurate word-level annotation by performing forced alignment of the manual annotations on the audio using our speech recognizer for Dutch [5], [21].

B. Debates

Another collection in CGN is a collection of debates with politicians (component-g). We have selected the 15 longest recordings of this component. Recording durations range from 18–30 minutes. Similar to the interview collection, we have generated the diarization reference by running forced alignments of the manual annotations on the audio.

C. The ICSI meeting corpus subset

The ICSI meeting corpus [22] consists of 75 recordings of meetings. Each meeting is recorded with both table-top microphones and close-talking microphones. We will use the table-top microphones, beam-formed into a single channel.

SRI has provided us with forced-aligned annotations of the meetings. Unfortunately, because these alignments were generated for the purpose of training phone models, the less useful utterances (e.g., read digit sequences) were not processed and hence these segments were not available in the forced-aligned reference. We have selected the 15 recordings of which most of the utterances were still present in the reference, but the incomplete annotation will likely cause an ‘artificial’ higher false alarm rate in the SAD error. Fortunately, this increase of error will be the same for all our experiments on this set and it does not affect the part of the diarization error rate due to misclassification of speakers so that we can still use the set to measure the efficiency of our approach.

The ICSI meetings are much longer than the debate recordings, durations are in the range 44–88 minutes. On average six people participate in each meeting (with a maximum of nine people and a minimum of four people). On average each speaker produces 293 utterances of 1.6 seconds. Most people participate in multiple meetings; in total the collection contains 25 unique speakers. 14 Of these speakers occur in three or less than three meetings. One speaker occurs in 8 meetings. On average, each speaker occurs in 3.5 meetings.

The fifteen meetings were part of recurring meetings on three topics. Five meetings are about the ‘Meeting Recorder Project’, two meetings are about ‘Robustness’ and eight meetings are titled: ‘Even Deeper Understanding’. This collection will be a very nice test

bed for testing the diarization procedure for small collections. Speakers occur in multiple recordings in various combinations. Because not all recordings contain the same speakers, it will be very important that the stopping threshold is set correctly.

VI. EXPERIMENTS

A number of experiments have been performed to test the efficiency of the approach. First, the minimal feasible chunk length is determined on the collection of interviews with school teachers. Next, the proposed method for linking the chunks is evaluated. In the last set of experiments, the interview collection and the meetings collection are linked across recordings.

A. Minimal chunk length

In the first set of experiments we will segment the speech in each recording in chunks of varying minimum length. We will perform speaker diarization on each chunk and evaluate the chunks as if they were separate recordings. In order to do so, we also segment the reference segmentation files to match with the chunks.

These experiments will give insight of the diarization module performance on speech chunks of varying length. The overall diarization error rate also tells us how well our method of diarization for long recordings would work if the speaker detection step would perform flawlessly.

TABLE I

DIARIZATION ERROR RATES OF THE DUTCH (NL) AND FLEMISH (VL) INTERVIEWS WITH TEACHERS WHEN USING CHUNKS OF VARYING LENGTH. THE FIRST TWO COLUMNS CONTAIN THE NUMBER OF RECORDINGS THAT ARE SEGMENTED INTO TWO CHUNKS OR MORE. WITH ‘ONE CHUNK’, THE DATA IS NOT DIVIDED INTO MULTIPLE CHUNKS. WITH ‘ONE SPEAKER’ ALL SPEECH IS ASSIGNED TO A SINGLE SPEAKER.

Chunk length (minutes)	NL #chunked	VL #chunked	NL %DER	VL %DER
2.5	80	80	8.36	12.31
5.0	76	80	5.41	9.82
7.5	6	52	4.74	8.94
10.0	0	15	4.76	9.00
12.5	0	4	4.76	9.86
one chunk	0	0	4.76	10.15
one speaker	n.a.	n.a.	12.11	18.93

Table I contains the experimental results for chunks with a minimum length of 2.5 up to 12.5 minutes and the results when performing diarization on the entire recording at once ('one chunk', the best we can do) and when all speech is attributed to a single speaker ('one speaker', the worst we can do). We expected that the Diarization Error Rate (DER) would be highest for the smallest chunk and decrease when increasing the chunk size. The table shows that the DER is near constant for chunks with 7.5 minutes of speech or more and deteriorates for smaller chunk sizes. Therefore, we will set the chunk size to 7.5 minutes for the remainder of the experiments.

The chunks in Table I were scored as if they are individual recordings. This is exactly the same as linking the chunks perfectly and then score each entire recording. The results from the table can therefore be seen as cheating experiments for the chunk linking method. For a chunk size of 7.5 minutes speech, the results after chunk linking will be 4.74 %DER (NL) and 8.94 %DER (VL) at the very best. These cheating experiments show that, although it was not our goal, the diarization results can be improved by applying the chunking approach as long as chunk linking is done flawlessly.

B. Chunk linking

In this set of experiments we will run the entire system, including automatically linking the speaker clusters across chunks using our speaker detection system. The results for the interview collection is listed in Table II. The results of the debate and meeting collections are shown in Table III. For the meetings and debates, a chunk size of 7.5 minutes of speech was used.

TABLE II
DIARIZATION ERROR RATES OF THE DUTCH (NL) AND FLEMISH (VL) INTERVIEWS WITH TEACHERS WHEN USING CHUNKS OF VARYING LENGTH THAT ARE CONCATENATED AUTOMATICALLY WITH USE OF THE SPEAKER DETECTION MODULE.

Chunk length (minutes)	NL %DER	VL %DER
2.5	10.04	18.06
5.0	5.71	10.44
7.5	4.74	9.06
10.0	4.76	9.23
12.5	4.76	9.92
one chunk	4.76	10.15
one speaker	12.11	18.93

As can be seen in Table II, the DER for chunks of 7.5 minutes or more are nearly as good as the DER in Table I with ideal chunk linking. For some of the chunk lengths, the result is even exactly the same as for the optimum. These results are promising, but it should be noted that in this particular case, the chunk linking step is not very difficult. There are only two speakers that should be linked and one of these speakers, the interviewer, talks considerably less than the second person. Linking all speech of the interviewee will be enough to get good results. The ‘one speaker’ line in Table II indicates the difficulty of this task: even if everything goes wrong during linking, the DER will not be above 12.11 % (NL) and 18.93 % (VL).

TABLE III

DIARIZATION ERROR RATES OF THE DEBATE COLLECTION AND THE MEETING COLLECTION. THE FIRST ROW CONTAINS THE SYSTEM OUTPUT. THE SECOND ROW CONTAINS THE OUTPUT WHEN THE STOPPING CRITERION WOULD PERFORM PERFECTLY AND THE THIRD ROW CONTAINS THE OUTPUT WHEN THE CHUNKS WOULD BE LINKED PERFECTLY.

Condition	%DER	
	Debates	Meetings
7.5	12.60	16.14
7.5, perfect stop	12.05	15.88
7.5, perfect linking	11.82	15.68
one chunk	9.84	14.56
one speaker	37.00	60.37

The chunk linking task is a lot more challenging for the debates and meetings. The chunks contain various numbers of speakers and not all speakers should be linked (not all participants are talking in each chunk). Table III lists the results of the chunk linking experiments on these two collections. The system performs less well on the meetings than on the debates. It must be noted though that because some speech is missing from the reference transcripts of the meeting collection (see Section V-C), the SAD error is unusually high: the debates have a SAD error of 1.2 % and the meetings an error of 9.6 %. Fortunately, because the SAD error is the same for all experiments in Table III, it is still fair to compare the experiments.

The first three rows of Table III show that the deterioration in diarization error rate due to chunk linking is limited but present. The stopping threshold introduces errors of 0.55 % and 0.26 % absolute⁶. The linking itself introduces another error of 0.23 % and 0.20 % absolute.

⁶Note that for the ‘stop cheating’ experiment, we used the optimal stopping point for each individual meeting. Such an optimum can not be reached by a single threshold for all recordings.

Considering that the speaker detection system used for determining the distances for chunk linking is developed for Italian telephone speech which is quite different from these two collections, these small error margins are acceptable.

A bit more disappointing is the deterioration in DER due to chunk-based diarization. For the interviews, chunk-based diarization even improved the results a little, but for these two collections, DER increases with 1.98 % (debates) and 1.12 % (meetings) absolute.

C. Computational efforts

We have measured the processor time needed by the system to finish diarization of the fifteen ICSI meetings collection. All meetings were diarized on a 2.27GHz Intel Xeon processor. When diarization is done in the original manner without chunking it takes 1936 CPU minutes to finish the entire collection. Diarization of the collection when chunking is applied took 355 CPU minutes in total. It took 230 minutes to perform diarization on each chunk, 123 minutes to enroll the speaker detection models, 2 minutes to calculate the detection scores and 1.29 seconds to perform the actual chunk linking. The collection of ICSI meetings consists of 917 minutes of audio. This means that diarization using the original algorithm was done in 2.1 times real-time while diarization in chunks is done in 0.39 times real-time (5.5 times faster).

Based on our prediction of the reduction of needed training and BIC steps (Section III), we expected the speed difference to be even bigger than this. In Section III, we calculated the difference in number of training and BIC steps for a recording containing 25 minutes of speech. The ICSI recordings contain much more speech and therefore the initial number of clusters is fixed to 40 for all the meetings. Also note that our code for speaker detection enrollment is not yet optimized for speed. We expect to be able to decrease the time needed for enrollment considerably.

D. Diarization of the entire collection of meetings

Our baseline approach for ‘large scale diarization’, diarization of an entire collection, is to generate chunks with speaker models for the individual recordings and then link all these chunks at once as if it is one long recording. We have applied this technique to the collection of meetings and evaluated the output of the system in two ways: as a diarization experiment and as an information retrieval experiment. In this section we will discuss the diarization

experiment and we will analyze our proposed method for normalizing the speaker detection scores. In the following section we will discuss the information retrieval experiment.

Diarization experiment: In this experiment we perform large scale diarization on the fifteen ICSI meetings (see Section V-C) where each speaker is uniquely identified across all meetings. For calculating the diarization error rate, each reference speaker is mapped to one single speaker cluster that spans across all recordings. Table IV contains the results of this experiment. The DER measured across meetings (25.16 %) is considerably higher than the overall DER calculated on the separate meetings (16.14 %), cf Table III. We also investigated the effect of collection wide linking restrictions to the diarization performance of individual meetings, this led to a slightly increased DER of 17.79 %. Clearly, the task of tracking speakers throughout the entire collection is more difficult than only tracking them within each recording.

TABLE IV
DIARIZATION OF THE ENTIRE COLLECTION OF MEETINGS.

Experiment	Result (%DER)
scored entire collection, chunks entire collection	25.16
scored per meeting, chunks entire collection	17.79
scored per meeting, chunks per meeting	16.14

Analysis of the proposed normalization method: The experimental results of our large scale diarization system discussed in this section look promising, but it must be noted that the system is not yet successful for all types of collections because of the normalization method that we used. To normalize the speaker detection scores of each cluster we use the other (non-target) speakers from the same chunk (see Section III-A). This works best if the other speakers are fairly similar to the target speaker. Otherwise the stopping criterion for speaker linking might be set too leniently, causing the linking component to over-cluster⁷. If a chunk only contains one or two speakers, chances that the speakers are similar will be slim and thus the chance of over-clustering will be big. We observed this when applying the system on the interviews with school teachers. In this collection, each chunk typically contains one or two (sometimes three because of diarization errors) speakers and without

⁷Note that this is not the speaker diarization stopping criterion, but the criterion for stopping linking clusters of the various chunks.

restricting the linking component, it over-clusters (it links too many clusters) in most cases. The overall DER measured on the entire collection of school teacher interviews (comparable with the first line of Table IV) is 97 % for Dutch and 71 % for Flemish. The over-clustering did not affect the performance measured on the individual recordings (similar to the second line of Table IV). These diarization error rates were identical to the originally obtained rates.

E. Information retrieval

With the output of the collection wide diarization experiment described above, it is possible to perform a query-by-example information retrieval experiment. In the query-by-example type of information retrieval, an example ‘document’ is used as a query to find all other documents that are similar to the query document. This approach is often taken for example in image retrieval where an example picture is used to find other similar images.

The retrieval task: In order to perform a query-by-example information retrieval experiment, we need to define the ‘documents’ (the images in the image retrieval example), we need to define queries and we need to define when a document is considered relevant to a query.

The documents, the items that are presented to the user as query result, should be fragments of the meetings that are short enough for a user to listen to (not the entire meeting) and long enough to obtain information from. For a fair experiment, the length and the contents of the documents should not depend on our system output in any way. Also, because for real-life systems no reference speaker transcripts will be available, we do not want to use the reference transcriptions to create the documents. Instead, the collection is divided into documents of two minutes long each without taking into account how many speakers there are in a document.

A set of queries is generated by randomly picking timestamps at which someone is talking according to the reference transcripts. The system should retrieve all documents that are relevant to these queries. We consider a document relevant when the query speaker is talking in the document; he or she does not need to be talking the most.

The ICSI collection consists of a fixed set of documents. In our experiment, the system will need to rank these documents in order of relevance. Once this is done we will evaluate the system using two commonly used information retrieval metrics: Mean Average Precision (MAP) and the average precision-at-ten.

Precision is the fraction of retrieved documents that is relevant to the query. The precision-at-ten is defined as the fraction of relevant documents in the top 10 ranked documents. The

average precision-at-ten is the precision-at-ten averaged over all queries.

The precision can be calculated at each point in the ranked list. For example, precision-at-twenty is the precision of the twenty top ranked documents. The Average Precision (AP) is obtained by calculating the average at all points in the ranked list that contain a relevant document:

$$AP = \frac{\sum_{r=1}^N P(r) \times rel(r)}{N_r} \quad (7)$$

where N_r is the total number of relevant documents. The Mean Average Precision (MAP) is the mean of the AP for all queries.

The document size was chosen somewhat arbitrarily. For very short documents (less than a second each), average precision is essentially the same as one minus the diarization error rate and very long documents would be impractical for the user to skim through. Two minute long documents could already be considered to be too long to skim through and therefore we also evaluated the system using documents of only 12 seconds (one tenth of two minutes). When using the two minute long documents, the collection consists of 453 documents. Using the twelve second documents, the collection consists of 4530 documents.

The system approach: The retrieval result is generated from the collection wide diarization segmentation as follows. First, the ID of the speaker that is talking at the query timestamp is determined. Next, all documents are listed in which the speaker is present, sorted on speaking time. The retrieval result is evaluated using the reference speaker segmentation in a similar fashion. Using the reference segmentation, for each query it is determined who was talking and the list of documents is created in which that speaker is talking as well. With this reference list of relevant documents, the average precision and the precision-at-ten of each retrieval result is calculated. The mean average precision and the average precision-at-ten is calculated for the entire set of queries.

The results: Because the queries are generated randomly, speakers that talk a lot in the collection will be queried for more often than others and therefore the MAP will be biased towards these speakers. It depends on the application if this is representative. We also calculated MAP for all queries of each individual speaker and then calculated the total average MAP, a speaker time normalized mean average precision.

The results of the information retrieval experiments are listed in Table V. In total 10.000 queries were generated and tested. The collection contains 25 speakers. In Figure 2 the MAP

TABLE V
DIARIZATION OF THE ENTIRE COLLECTION OF MEETINGS.

Experiment	MAP	Precision-at-ten
2min, no normalization	0.57	0.74
2min, speaker normalization	0.37	0.49
12s, no normalization	0.49	0.70
12s, speaker normalization	0.31	0.49

of each individual speaker is plotted. As can be seen in this graph, with the exception of two speakers, all speakers that occur in less than 50 documents score under 0.3. Only two speakers that occur in more than 50 documents score under 0.3. We have calculated the total speaking time of each speaker and, except for the speaker with MAP 0.93, the speakers that occur in less than 50 documents also speak significantly less than the other speakers. Speakers that talk little and infrequently are often missed by the diarization system (indirectly because this is not penalized in the evaluation measure, the DER), this explains why these speakers score low on average precision.

Fig. 2. The mean average precision of each speaker in the ICSI meeting collection (every speaker is assigned multiple queries) plotted against the number of relevant documents for each speaker.

VII. DISCUSSION AND FUTURE WORK

In this section, we will discuss the results of the diarization system for long recordings and of the system for diarizing entire collections. We will conclude this paper with a discussion of possible future work.

A. *Diarization for long recordings*

State-of-the-art speaker diarization systems based on the combination of GMMs and agglomerative clustering are not able to process very long recordings⁸. In this paper we have proposed a solution for this scaling problem. The proposed system makes use of both speaker diarization and speaker detection technology. The experiments show that the system can diarize long recordings almost as accurately as the original diarization system, but with less computational efforts. We have analyzed the effect of the chunk length and the effect of errors made during the linking process on the diarization accuracy. An optimum was found for chunks with 7.5 minutes of speech and we showed that the average error due to linking is less than 1% absolute.

Even though the speaker detection component was originally developed for entirely different data, it was able to link the speakers from the various chunks very well. We used the system out-of-the-box except for score-normalization, because the original normalization method (with speaker models trained on the Evalita'09 data) did not lead to adequate results. Instead we successfully made use of the assumption that the diarization system does a reasonable job and normalized the scores of each speaker model on the other speakers appearing in the same chunk. This normalization method provided us with a natural threshold for the stopping criterion.

B. *Diarization for small collections*

Our approach for diarization of long recordings is also useful for diarization of small collections. We tested the approach on a collection of fifteen ICSI meetings with encouraging results. The diarization error rate increases when measuring across recordings, but this is — at least in part— due to the more difficult task. We inspected the system output and found that the system has over-clustered two out of fifteen times. It merged four speakers (twice two speakers) that it should not have merged. This problem could not be solved by tuning the stopping threshold. We might need a more accurate detection component to solve this problem.

We showed that our proposed method for normalization of the speaker detection scores works for the ICSI collection, but that it is not able to normalize the scores properly for

⁸The maximum processable length depends on the efficiency of the system. Ours can handle recordings up to two hours

recordings with only two speakers such as the interview recordings. Because of this, the stopping criterion for linking clusters across chunks fails for the interview collection.

In our speaker information retrieval experiment we showed that the quality of our large scale diarization system is high enough to perform retrieval (MAP between 0.31 and 0.57). We showed that speakers that occur regularly in meetings and that speak often, are easier to retrieve. It will be interesting to investigate if these results can also be obtained for larger real-life collections.

C. Future work

We enrolled the speaker detection models on all the data of each speaker cluster (generated by the diarization component). We also scored each model on all the data of the other speaker clusters. The fact that none of these speaker clusters is perfect—clusters will contain some speech of other speakers—will influence the quality of the detection models. In future work we will investigate if it is helpful to use only the segments with the highest diarization likelihoods to train and score the speaker detection models.

The speaker detection component itself can also be improved by training its background models on more representative data and on data of both genders (because we did not have gender independent models, we used the male models). Finding well balanced representative data is very hard, but once we have diarized a big collection (such as the entire CGN collection) with our current system, we might be able to train background models on the data that were automatically obtained.

In our current approach we enroll speaker detection models only once for each speaker. We intend to try methods similar to the diarization approach where the speaker models are retrained after each linking iteration. This might be especially helpful for diarization of collections where speakers are recorded in multiple sessions. The speaker detection system is developed to handle channel variability in order to be able to detect the same speaker across different sessions, but by re-training the speaker model over multiple sessions, obviously more robust speakers will be obtained.

In this paper we have shown that our approach is useful for diarizing small collections, although normalization of the speaker detection scores remains a problem. It is possible to use the non-target speakers from the same chunk for normalization, but this only works when the chunks contain enough (similar) speakers. Finding representative speech data to train

normalization speaker models will be one of the main challenges when processing bigger collections.

We expect to run into more challenges when we upscale to bigger collections. Training all the detection models, scoring each model on all data and performing the linking itself will have its own computational challenges for big collections, but thus far the development in speaker detection is towards more efficient modeling and scoring, such as ‘dot scoring’ [23] and the i-vector approach [24], which can make use of very efficient matrix-multiplication optimizations of software libraries on modern computing hardware for computing the score matrix. Working with incrementally growing data collection will require additional approaches to maintain scalability of computation and performance. Creating chunks of collections and in turn merging these chunks (of chunks) is one such an approach.

In future work we will start investigating methods to automatically or semi-automatically, based on ASR content or meta-data, label each speaker cluster with the true identity of the speaker so that it really becomes possible to track people and to search collections for fragments of known speakers.

ACKNOWLEDGMENTS

This paper is based on research carried out for the BATS project within the research program IM-Pact funded by IBBT and ICTRegie. We would like to thank Ondrej Glembek for the JFA cookbook, Xavier Anguera [25] for the beamforming software and Andreas Stolcke for providing forced aligned transcriptions of the ICSI meeting corpus.

REFERENCES

- [1] D. Reynolds and P. Torres-Carrasquillo, “Approaches and applications of audio diarization,” in *Proceedings of Acoustics, Speech, and Signal Processing*, (Philadelphia, PA), pp. 953–956, March 2005.
- [2] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds, “The NIST speaker recognition evaluation—Overview, methodology, systems, results, perspective,” *Speech Communication*, vol. 31, pp. 225–254, 2000.
- [3] C. Wooters and M. Huijbregts, “The ICSI RT07s speaker diarization system,” in *Multimodal Technologies for Perception of Humans*, vol. 4625 of *Lecture Notes in Computer Science*, (Berlin), Springer Verlag, 2008.
- [4] D. van Leeuwen and M. Huijbregts, “The AMI speaker diarization system for NIST RT06s meeting data,” in *Machine Learning for Multimodal Interaction (MLMI)*, vol. 4299 of *Lecture Notes in Computer Science*, (Berlin), pp. 371–384, Springer Verlag, October 2007.
- [5] M. Huijbregts, *Segmentation, diarization and speech transcription : surprise data unraveled*. PhD thesis, Enschede, November 2008.
- [6] A. Martin and M. Przybocki, “The NIST 1999 speaker recognition evaluation—an overview,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 1 – 18, 2000.

- [7] R. B. Dunn, D. A. Reynolds, and T. F. Quatieri, "Approaches to speaker detection and tracking in conversational speech," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 93 – 112, 2000.
- [8] D. van Leeuwen, "Speaker linking in large data sets," in *Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop*, (Brno), June 2010.
- [9] L. J. Rodríguez, M. Peñagarikano, and G. Bordel, "A simple but effective approach to speaker tracking in broadcast news," vol. 4478, (Girona, Spain), pp. 48–55, 2007.
- [10] J. Zibert, B. Vesnicer, and F. Mihelic, "A system for speaker detection and tracking in audio broadcast news," *Informatica (Slovenia)*, vol. 32, no. 1, pp. 51–61, 2008.
- [11] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Special Topics in Signal Processing*, December 2010.
- [12] L. Canseco-Rodriguez, L. Lamel, and J.-L. Gauvain, "Speaker diarization from speech transcripts," in *Proceedings of the International Conference on Spoken Language Processing*, 2004.
- [13] D. Imseng and G. Friedland, "Robust speaker diarization for short speech recordings," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, (Merano , Italy), December 2009.
- [14] M. Huijbregts, C. Wooters, and R. Ordelman, "Filtering the unknown: Speech activity detection in heterogeneous video collections," in *proceedings of Interspeech*, (Antwerp, Belgium), August 2007.
- [15] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The rich transcription 2007 meeting recognition evaluation," in *Multimodal Technologies for Perception of Humans*, Lecture Notes in Computer Science, (Berlin), Springer Verlag, 2008.
- [16] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan, "Unknown-multiple speaker clustering using HMM," in *proceedings of the International Conference on Spoken Language Processing*, (Denver, Colorado, USA), 2002.
- [17] G. Schwartz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [18] M. Huijbregts and D. van Leeuwen, "The RU submission to the "evalita'09 application" track speaker recognition evaluation," in *the 11th Conference of the Italian Association for Artificial Intelligence*, (Reggio Emilia, Italy), December 2009.
- [19] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, July 2008.
- [20] N. Oostdijk, "The spoken dutch corpus. outline and first evaluation.," in *Proceedings of the Second International Conference on Language Resources and Evaluation*, 2000.
- [21] D. van Leeuwen, J. Kessens, E. Sanders, and H. van den Heuvel, "Results of the n-best 2008 dutch speech recognition evaluation," in *proceedings of Interspeech*, (Brighton, UK), September 2009.
- [22] A. J. et al, "The ICSI meeting project: Resources and research," in *NIST 2004 Meeting Recognition Workshop at the International Conference on Acoustics, Speech and Signal Processing*, (Montreal), May 2004.
- [23] O. Glembek, L. Burget, N. Dehak, N. Brümmer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, (Taipei, Taiwan), April 2009.
- [24] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proceedings of Interspeech 2009*, (Brighton, UK), September 2009.
- [25] X. Anguera, *Robust Speaker Diarization for Meetings*. PhD thesis, Universitat Politècnica De Catalunya, 2006.

Marijn Huijbregts is a postdoc researcher at Radboud University Nijmegen. He received his MS (Ir.) from University of Twente in 2001 after which he worked in industry on the development of Bluetooth headsets and car-kits. After three years Marijn started his Ph.D. research on automatic speech recognition for multimedia retrieval. The main focus of this study was on robust speaker diarization for which he visited the International Computer Science Institute in Berkeley, CA. Marijn received his Ph.D. degree from University of Twente in 2008. In Nijmegen, Marijn is currently performing research on speaker retrieval for large broadcast television archives within the research program IM-Pact funded by IBBT and ICTRegie.

David A. van Leeuwen (Ir.[MS]1984Delft University of Technology, Dr. [PhD] 1993 University of Leiden) was with TNO Human Factors since 1994 and is with Radboud University Nijmegen since 2008. He has worked in various areas in speech technology, with special interests in evaluation of systems. Examples of evaluations he organized are the EU FP5 SQALE project (1995), evaluating large vocabulary speech recognition systems in four languages, the NFI-TNO Forensic Speaker Recognition Evaluation with 11 international participants in 2003, and the N-Best evaluation of Dutch speech recognition systems in 2008. He has also successfully participated in various NIST Rich Transcription, Speaker and Language Recognition Evaluations. In 2008 he was appointed professor at Radboud University and he co-authored the EU FP7 Marie Curie ITN project Bayesian Biometrics for Forensics (BBfor2), which he is currently leading. In recent years he has been focussing on speaker diarization and automatic speaker and language recognition, with special interest in highly accurate and efficient systems, forensic application scenarios and calibration.