INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Using a GPU, Online Diarization = Offline Diarization

Gerald Friedland

TR-12-004

January 2012

Abstract

This article presents a low-latency, online speaker diarization system ("who is speaking now?") based on the repeated execution of a GPU-optimized, highly efficient offline diarization system ("who spoke when"). The system fulfills all requirements of the diarization task, i.e., it does not require any a priori information about the input, including specific speaker models. In contrast to earlier attempts at online diarization, the system achieves similar accuracy to the underlying offline system and does not require explicit detection of new speakers. Using GPUs, online diarization has become a side-effect of offline diarization, obsoleting the requirement for specialized online diarization systems.

This research is partly supported by Microsoft (Award # 024263) and Intel (Award # 024894) funding, by matching funding by U.C. Discovery (Award # DIG07-10227), and a CISCO URP grant. I want to thank the following persons for their support in writing this article: Adam Janin, Luke Gottlieb, Carlos Vaquero, Henry Cook, Mary Knox, and Nelson Morgan.

1. INTRODUCTION

Traditionally, the task of speaker diarization is to segment an audio signal into speaker-homogeneous regions addressing the question “who spoke when?” without any prior knowledge of the number of speakers, specific speaker models, text, language, or amount of speech present in the recording [14]. Diarization has mainly been addressed as an offline task. In other words, conventional systems make use of all available data in the recording before making a decision about how many speakers are present and when each of them is speaking. While offline processing offers the possibility to make use of long-term assumptions and optimize globally over the entire recording, there are many applications, including dialog systems and videoconferencing, which require online processing or, informally, “who is speaking now?”. For example, a robot that interacts with several people might perform online diarization to turn its head to the actual speaker to make its response seem more natural. The major difficulty of online processing is that decisions are based on much less data. For example, at a given point in time, a speaker might enter the conversation who had not yet been registered by the system. Overcoming this problem using speaker identification with pre-trained speaker-specific models (as in [17]) would not be considered a diarization system, as diarization requires no speaker- or recording-specific a-priori training.

This article presents a novel approach to online speaker diarization, where online diarization is produced as a side-effect of offline diarization. The approach uses an offline diarization system that was highly optimized using GPU technology, namely the NVidia CUDA framework, to process audio recordings at about $0.004\times$ realtime. At this speed, the diarization system can be treated as a filter that progressivly re-runs over the existing audio recording every couple of seconds to take advantage of all available audio information up to the current time. The highly parallel system is able to output a new decision for the current speaker every 2.5 seconds for audio chunks up to the length of about 10 minutes. The models trained are held in a pool that is updated periodically and used as an initialization for the next 2.5 second run. Intuitively, the farther one progresses into the meeting, the higher the accuracy of the system since more data is available to generate better models. The experiments show that for a subset of meetings from the AMI corpus, the accuracy of the online system that outputs decisions every couple seconds (after a short initialization phase) is about the same as the accuracy of the offline system, in which the entire meeting in processed before any decision is output.

2. RELATED WORK

Even though an experimental low-latency task was introduced in the RT’09 evaluations, speaker diarization research so far has mostly focused on improving offline diarization performance. The systems presented in [17], [7], and [15] are online speaker identification systems since speaker-specific models were used. In [12], a framework based on multimodal information over Dynamic Bayesian Networks was proposed with the goal of creating an online speaker diarization system. Initial experiments using the framework were encouraging, but the experimental setup was very controlled and consisted of a small dataset. More elaborate approaches include the one presented in [9] that uses a bootstrapping approach using a UBM, which was later refined in [10]. The latter is able to detect new speakers in a recording without any prior knowledge of the speakers using audio from a single microphone. However, the system relies heavily on the accurate detection of new speakers and on speaker models that are trained according to online decisions. This strategy, however, leads to error accumulation. Our previous work [16] solves this problem through the use of hybrid online/offline processing, making use of all available information to train speaker models and not relying completely on online decisions, thus avoiding error propagation. However, the accuracy of the online system described in [16] did not reach the accuracy of the underlying offline system. Most importantly, all the previous attempts were specialized approaches to online diarization, while the approach presented here presents online diarization as a side-effect of repeated execution of offline diarization, mapping the problem of online diarization back to a offline diarization.

3. OFFLINE DIARIZATION

The underlying offline speaker diarization system is a state-of-the-art diarization engine [3] that performed very well in the 2007 and 2009 NIST Rich Transcription evaluations[1]. It is based on 19-dimensional, gaussianized, Mel-Frequency Cepstral Coefficients (MFCCs). A frame period of 10 ms with an analysis window of 30 ms is used in the feature extraction. The speech/non-speech segmentation utilized in [3] is used and described in [6]. It is an HMM/GMM approach originally trained on broadcast news data that generalizes well to meetings. In the segmentation and clustering stage of speaker diarization, an initial segmentation is generated by uniformly partitioning the audio track into k segments of the same length. k is chosen to be much larger than the assumed number of speakers in the audio track. For meeting recordings of about 30 minute length, previous work [8] found experimentally that $k = 16$ is a good value.

The procedure for segmenting the audio data takes the following steps:

1. Train a set of GMMs for each initial cluster.
2. Re-segmentation: Run a Viterbi decoder using the current GMMs to segment the audio track.
3. Re-training: Retrain the GMMs using the current segmentation as input.
4. Select the closest pair of clusters and merge them. At each iteration, the algorithm checks all possible pairs of

[1] NIST regulations prevent us from presenting a ranking number.

clusters to see if there is an improvement in BIC scores when the clusters are merged and the two models replaced by a new GMM trained on the merged cluster pair. The clusters from the pair with the largest improvement in BIC scores, if any, are merged and the new GMM is used. The algorithm then repeats from the re-segmentation step until there are no remaining pairs that will lead to an improved BIC score when merged.

The result of the algorithm consists of a segmentation of the audio track with n clusters and with one GMM for each cluster, where n is assumed to be the number of speakers. A more detailed description can be found in [3].

4. DIARIZATION ON A GPU

A prior analysis of the above described diarization system [2] showed that there are two main computational bottlenecks to offline diarization: The training of the Gaussian Mixture Models, mostly during the merging phase that requires $\binom{n}{2}$ comparisons to determine the cluster pair to merge [5], and the Viterbi alignment. In prior work [4], it was shown that for the system used here (described in Section 3 and [3]), Viterbi alignment can be replaced by a local majority vote without a significant change in accuracy. As in [4], Viterbi alignment was therefore replaced by a majority vote on the emitted log-likelihoods per chunk of 250 frames to increase efficiency, i.e., a speaker decision is made every 2.5 seconds. The training of the Gaussian Mixture Models is performed on the GPU by adopting the expectation-maximization-algorithm described in [13]. The main idea is to reduce the expectation and maximization steps to a set of kernel functions; the algorithm is outlined as follows:

1. Copy the input data to the GPU
2. Initialize the GMMs and copy to the GPU
3. Launch expectation kernels; aggregate log-likelihood values from each GPU
4. Launch maximization kernels; aggregate parameters from each GPU
5. Repeat steps 3 and 4 a fixed number of times (currently 5).
6. Copy membership values to host

The details of the data organization on the GPU are described in detail in [13]. This training algorithm is used both in the re-training step as well as in the select and merge step. The remaining computation (including calculating BIC) is currently performed on the CPU. Note that this realization may not be optimally efficient. However, the current speed is about $0.004\times$ realtime and therefore diarizes a 600 second chunk of audio in about 2.4 second using an NVidia GeForce GTX280 GPU. The GMM parameters are initialized at random for the first run and are based on previous models for subsequent runs thereafter, as described in the following section.

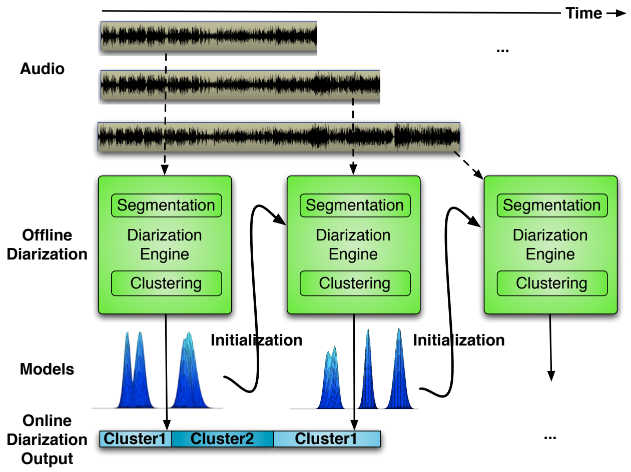

Fig. 1. *Overview diagram of the online diarization approach as described in this paper: Repeated execution of a very fast offline system is used to diarize increasing chunks of audio data while they pass along models to initialize each other. Using a GPU, online diarization is virtually a side-effect of offline diarization.*

5. FROM OFFLINE TO ONLINE

With the offline diarization system running about 250 times faster than realtime, online diarization can be performed almost trivially by applying the offline diarization many times. Online diarization is performed by running the offline system whenever the length of the audio recording has increased by 2.5 seconds until the recording is about 600 seconds. At this point the runtime of the diarization system is about 2.4 seconds and since no further latency is to be introduced, the offline system is then used to diarize only the last 600 seconds of the audio recording. An initialization phase of 150 seconds is introduced, as this enables running the diarization system at full accuracy, as explained in [8]. So the first offline diarization system is run after 150 seconds of recorded audio, the second system is then run after 152.5 seconds, the third after 155, etc. Figure 1 illustrates the idea of the online diarization approach.

The output of each run of the offline diarization system consists of a segmentation assigning speech frames to clusters and speaker models trained for each cluster. The models are used to initialize the subsequent diarization system that is run on the existing audio recording (which only differs from the previous audio recording by the last 2.5 seconds). Since agglomerative hierarchical clustering requires initialization with a higher number of models than assumed speakers – and also new speakers could be introduced at any time – the remaining $16 - n$ models are initialized at random using frames contained in the existing audio recording. Since the output cluster

Table 1. *Comparison of the Diarization Error Rate of the baseline system vs the online system*

Meeting ID	Offline	Online
IS1000a	42.40 %	41.82 %
IS1001a	39.40 %	39.43 %
IS1001b	35.50 %	35.34 %
IS1001c	30.40 %	28.92 %
IS1003b	31.40 %	30.84 %
IS1003d	56.50 %	57.30 %
IS1006b	24.10 %	24.34 %
IS1006d	60.40 %	60.00 %
IS1008a	8.20 %	7.92 %
IS1008b	10.10 %	11.11 %
IS1008c	14.40 %	15.29 %
IS1008d	32.30 %	31.34 %
Average	32.09 %	31.97 %

labels might not match for each run, e.g. speaker 1 of a previous offline diarization run might now be labelled speaker 2, the labels have to be matched after each run. This is done by using the label matching algorithm of the NIST *mdeval* tool which is used for measuring Diarization Error Rate, as explained in Section 6. The tool uses a dynamic programming procedure to find the optimal one-to-one mapping between the hypothesis and the ground truth segments so that the total overlap between the reference speaker and the corresponding mapped hypothesized speaker cluster is maximized [1]. The processing time needed for performing this mapping turned out to be computationally negligible. Each diarization is used to output speaker labels for the most recent 2.5 seconds of recording time.

Note that new speakers are implicitly detected by the agglomerative hierarchical clustering algorithm and there is no need to compare speakers against a background model, which often causes the problem of having to determine a threshold.

6. RESULTS

In order to evaluate the accuracy of the described approach, a subset of 12 meetings (5.4 hours) from the Augmented Multi-Party Interaction (AMI) corpus[2] was used. The AMI corpus consists of audio-visual data captured of four to six participants in a natural meeting scenario. The 12-meeting subset used is popular as it contains the most comprehensively annotated meetings in the corpus, and is preferable since it allows for the quantitative evaluation of meeting analysis algorithms and the comparison of different approaches to the same task on a common dataset. Thus, it is commonly used by many researchers. Since our work investigates an unsupervised approach and we are re-using the already separately tuned parameters of the offline engine, there is no need to split the data into test and training sets.

For the experiments described here, the beamformed far-field array microphone signals were used. To make the scores compatible with the baseline system, the Shout speech activity detection was used as described in Section 3 and in [3]. Although Shout requires the entire audio recording and does not work incrementally, many speech activity detectors work online and have accuracies in the high-ninety percents. Also, MFCC features were calculated as a preprocessing step.

Once a decision is made for a 2.5-second chunk we do not retroactively change previous outputs of the online diarization. Therefore, the online system can be scored just like the offline system once processing of the entire meeting has finished. This allows us to compare both systems. The output of both online and offline diarization is scored using Diarization Error Rate, which is defined by NIST [11]. DER is composed of two additive components: speech/non-speech error and speaker error. Table 1 shows the results comparing the offline system and the online system. The online system being 0.12 % better on average can be disregarded as insignificant.

7. CONCLUSION AND FUTURE WORK

This article presents an online diarization system that takes advantage of the fact that offline diarization can be performed very quickly using current GPU hardware. Informally speaking, the presented realization of agglomerative hierarchical clustering on a GPU allows treating offline speaker diarization computationally similar to a low-level filter operation. The experiments show that only very few further steps are necessary to convert a series of repeated offline diarization runs to an online system that outputs the current speaker every couple of seconds. The system implicitly detects new speakers, eliminating an accuracy bottleneck of previous systems and has a similar Diarization Error Rate to the offline system, which is run on the entire audio recording. Note that the proposed approach is not specific to any particular algorithm and replacing the offline diarization component with a more accurate one will most likely result in a better online diarization as well. Future work includes the development of a highly efficient online speech activity detector and feature extractor that allows the system to work in an actual demo setting. A further analysis of the behavior of the online system versus the offline system will be interesting, especially an analysis on other domains and with complicated cases where new speakers are introduced often.

8. REFERENCES

[1] Fiscus, J. G., Ajot, J., Radde, N., and Laprun, C.: "Multiple Dimension Levenshtein Edit Distance Calculations for Evaluating Automatic Speech Recognition Systems During Simultaneous Speech", Proceeding of LREC, 2006.

[2] Friedland, G., Chong, J., and Janin, A.: "Parallelizing Speaker-Attributed Speech Recognition for Meet-

[2] http://corpus.amiproject.org/

ing Browsing", Proceedings of IEEE International Symposium on Multimedia, pp.121-128, Taichung, Taiwan, December 2010.

[3] Friedland, G., Janin, A., Imseng, D., Anguera, X., Gottlieb, L., Huijbregts, M., Knox, M, and Vinyals, O.: "The ICSI RT-09 Speaker Diarization System", IEEE Transactions on Audio, Speech and Language Processing, to appear 2011.

[4] Friedland, G., and Vinyals, O.: "Live Speaker Identification in Conversations", Proceedings of ACM Multimedia, pp. 1017–1018, Vancouver, Canada, October 2008.

[5] Huang, Y., Vinyals, O., Friedland, G., Müller, C., Mirghafori, N., and Wooters, C.: "A Fast-Match Approach for Robust, faster than Real-Time Speaker Diarization", Proceedings of IEEE ASRU, pp. 693–698, Kyoto, Japan, December 9–13, 2007.

[6] Huijbregts, M: "Segmentation, Diarization, and Speech Transcription: Surprise Data Unraveled", Doctoral Disseration, PrintPartners Ipskamp, Enschede, The Netherlands, 2008.

[7] Hung, H. and Friedland, G., "Towards Audio-Visual On-line Diarization of Participants In Group Meetings", Workshop on Multi-camera and multi-modal Sensor Fusion, M2SFA2, 2008.

[8] Imseng, D., and Friedland, G.: "Robust Speaker Diarization for Short Speech Recordings", Proceedings of IEEE ASRU, Merano (Italy), pp. 432-437, December 2009.

[9] Markov, K. and Nakamura, S., "Never-Ending Learning System for On-line Speaker diarization", in Proc. IEEE ASRU'09, 699–704, Merano, Italy, 2007.

[10] Markov, K. and Nakamura, S., "Improved Novelty detection for Online GMM based Speaker Diarization", in Proc. Interspeech'08, 363–366, Brisbane, Australia, 2008.

[11] National Institute of Standards and Technologies: "Rich Transcription Spring 2004 Evaluation": http://www.itl.nist.gov/iad/mig/tests/rt/2004-spring/index.html

[12] Noulas, A.K. and Krose, B.J.A, "On-line Multi-Modal speaker Diarization", International Conference on Multi-modal Interfaces, ICMI'07, 350–357, 2007.

[13] Pangborn, A. D.: "Scalable Data Clustering using GPUs, Master Thesis, Rochester Institute of Technology, May 2010.

[14] Reynolds, D. A. and Torres-Carrasquillo, P., "Approaches and applications of audio diarization", In Proc. IEEE ICASSP, V:953–956, Philadelphia, PA, 2005.

[15] Schmalenstroeer, J. et al, "Fusing audio and Video Information for Online Speaker Diarization", in Proc. Interspeech'09, 1163–1166, Brighton, UK, 2009.

[16] C. Vaquero, O. Vinyals, G. Friedland: "A Hybrid Approach to Online Speaker Diarization", Proceedings of ISCA Interspeech, pp. 2638-2641, Makuhari, Japan, September 2010.

[17] Vinyals, O. and Friedland, G., "Towards semantic analysis of conversations: A system for the live identification of speakers in meetings". In Proc. IEEE ICSC'08, 426–431, Santa Clara, CA, 2008.