Low-Latency Speaker Tracking and SOA-Compliant Services for Ambient Intelligence Environments

Maider Zamalloa [1,2], *Luis Javier Rodríguez-Fuentes* [1], *Germán Bordel* [1], *Mikel Penagarikano* [1], *Jorge Parra* [2], *Aitor Uribarren* [2], *Juan Pedro Uribe* [2]

[1] Department of Electricity and Electronics, University of the Basque Country, Spain
[2] Ikerlan - Technological Research Centre, Spain
{maider.zamalloa, luisjavier.rodriguez, german.bordel, mikel.penagarikano}@ehu.es,
{jparra, auribarren, JPUribe}@ikerlan.es

Abstract

As the most natural interface for human interaction, speech can be exploited to track users and then customize services as they get available. Low latency is required, since adaptation to user profiles must be done in a continuous fashion. However, most speaker tracking approaches found in the literature work offline, fully processing pre-recorded audio files by means of a two-stage procedure involving acoustic segmentation and speaker detection. In this work, a low-latency online speaker tracking approach is applied, which deals with continuous audio streams and outputs a decision at fixed intervals, by scoring fixed-length audio segments with regard to a set of target speaker models. Experimental results are reported on the AMI Corpus of meeting conversations, revealing the effectiveness of the proposed approach with regard to a traditional approach working offline. A speaker tracking service and a lower-level auxiliary speaker detection service have been also designed, based on the online low-latency speaker tracking approach mentioned above. These services are SOA-compliant and provide an interoperable, reusable and easily evolvable means to develop SOA-based speaker tracking applications for Ambient Intelligence (AmI) environments.

Index Terms: low-latency, speaker tracking, Service Oriented Architecture, Ambient Intelligence.

1. Introduction

In Ambient Intelligence (AmI) environments, human-computer interaction must be driven by intelligent and natural interfaces. Speech is a natural interface for human interaction and can be exploited to extract user related information such as location, identity, emotional state, etc. Speech is also a suitable means to support user adaptation. User adaptation must be done in a continuous fashion, which requires users to be continuously tracked (identified and located) in the AmI environment, so that customized services can be provided to all of them.

Speaker diarization and speaker tracking are well known speech processing tasks which aim to answer the question *Who spokes when?*, that is, to detect speaker turns in a continuous audio stream. Speaker tracking aims to detect segments corresponding to a known set of target speakers [1], whereas speaker diarization aims to detect speakers without any prior knowledge about them [2][3][4]. There are three primary application domains for speaker tracking and diarization: broadcast news audio, recorded meetings and telephone conversations. The methodologies applied in such domains assume that audio recordings are fully available before processing. So, common approaches to speaker tracking and diarization consist of two steps applied offline: (1) audio segmentation and (2) speaker detection. In speaker diarization, segments hypothetically uttered by the same speaker are clustered together and assigned the same label. In speaker tracking, once the audio stream is segmented, speaker detection is carried out through classical speaker recognition techniques [5][6][7]. In any case, these methodologies are not suitable for low-latency online speaker detection.

This paper presents results from our basic research on low-latency online speaker tracking, and describes tools which help shortening the deployment time of speaker tracking applications. Both elements were designed for an AmI scenario, for example an intelligent home environment, where the system continuously tracks known speakers (users), and the expected number of target speakers is low (i.e. the members of a family). As noted above, this scenario requires taking almost instantaneous (low latency) speaker tracking decisions.

The speaker tracking approach applied in this work jointly performs audio segmentation and speaker detection, by defining and processing fixed-length audio segments and scoring each of them to decide whether it belongs to a target speaker or to an impostor.

The performance of the proposed approach is compared to that of an offline system developed for reference, which follows a two-stage uncoupled approach. Speaker tracking experiments applying both systems were carried out on the AMI Corpus (Augmented Multi-party Interaction) [8], which contains human conversations in the context of smart meeting rooms, close to the AmI scenario described above.

From a practical point of view, the main contribution of this work regards the design of services helping the deployment of speech-based speaker tracking applications in a Service Oriented Architecture (SOA) framework [9][10]. SOA-based systems provide services to either end-user applications or to other services distributed in a network, via published and discoverable standard interfaces. SOA promotes the loose coupling between software components published as services, so they can be combined by service composition and reused in many applications. In addition, interoperability is also achieved, since services are neither dependent on the platform nor the programming language. In this work, two SOA-compliant UPnP services have been defined (a speaker tracking service and a lower-level auxiliary speaker detection service) based on the low latency real time speaker tracking approach described above.

The rest of the paper is organized as follows. In section 2, the main features of the online and offline speaker tracking systems are described, including speaker detection, score calibration and score smoothing. Section 3 gives details about the experimental setup. Section 4 presents and briefly discusses results attained in speaker tracking experiments. Section 5 describes the SOA-compliant speaker tracking services. Finally, conclusions are summarized in section 6.

2. Speaker tracking systems

2.1. Speaker detection

The online speaker tracking system applied in this work computes a detection score per target speaker and outputs a speaker identification decision at fixed-length intervals. That length has been empirically set to one second, which provides relatively good time resolution and spectral richness, and a reasonably small latency for most online speaker tracking scenarios. The offline system developed for reference does the same computation, but using the segments produced by an audio segmentation algorithm [11]. Regardless the way audio segments are obtained, scores are computed by means of acoustic models (corresponding to target speakers) estimated via Maximum A Posteriori (MAP) adaptation of a Universal Backround Model (UBM) [12]. Besides yielding good speaker recognition performance, the MAP-UBM methodology allows for a fast scoring technique which speeds up the score computation.

Acoustic vectors consist of 12 Mel-Filter Cepstral Coefficients (MFCC) + 12 ΔMFCC. Given an acoustic observation X (consisting of a sequence of acoustic vectors), the acoustic model λ_s for the target speaker s and the UBM, λ_{UBM}, the detection score $\Delta_s(X)$ is computed as follows:

$$\Delta_s(X) = L(X|\lambda_s) - L(X|\lambda_{UBM}) \quad (1)$$

where $L(X|\lambda)$ is the log-likelihood of X given λ. Once the detection scores are computed for all the target speakers, speaker detection can be accomplished according to two possible approaches:

1. In the *speaker identification (SI) approach*, X is marked as coming from the most likely target speaker $s^* = \arg\max_{s\in[1,S]}\{\Delta_s(X)\}$, if $\Delta_{s^*}(X) > \theta_I$. Otherwise X is marked as coming from an impostor. The decision threshold θ_I can be heuristically established to optimize the discrimination among target speakers. Note that, for any given segment X, there could actually be two or more speakers speaking at the same time. However, the detection approach described above cannot inform of speaker overlaps, because only the most likely speaker can be detected.

2. In the *speaker verification (SV) approach*, each target speaker s is accepted or rejected by comparing the detection score $\Delta_s(X)$ to a decision threshold θ_V. If $\Delta_s(X) > \theta_V$ the target speaker s is accepted; otherwise it is rejected. This approach allows to handle segments with overlapped speech, since all the target speakers for which $\Delta_s(X) > \theta_V$ are accepted.

2.2. Score calibration

Calibration maps detection scores $\{\Delta_s \mid s\in[1,S]\}$ to likelihood ratios $\{C(\Delta_s) \mid s\in[1,S]\}$ without any specific application in mind. The scaling parameters are computed over a development corpus by maximizing *Mutual Information*, which is equivalent to minimizing the so called C_{LLR} (a metric defined in [13]), which integrates the expected cost over a wide range of operation points. The final decision is taken by applying the minimum expected cost Bayes decision threshold to calibrated scores $C(\Delta)$. The target speaker is accepted only if the following inequality holds:

$$C(\Delta) \geq \ln\left(\frac{C_{fa}(1-P_{target})}{C_{miss}P_{target}}\right) \quad (2)$$

where C_{miss} and C_{fa} are miss and false-acceptance error costs, and P_{target} the prior probability of target speakers. Scores are calibrated by means of the *FoCal toolkit*, applying a linear mapping strategy (see http://www.dsp.sun.ac.za/~nbrummer/focal/).

2.3. Score smoothing

Since speaker detection is done for very short (one-second length) segments, the performance of the low-latency online speaker tracking system may degrade due to local variability. To increase the robustness to such variability, information from previous segments can be taken into account, that is, the acoustic scores of target speakers may be computed on speech segments lasting more than one second. Assuming that no speaker change takes place in the previous segments, scores will be more accurate as more samples are used to compute them. On the other hand, this does not affect the online processing and low-latency decision-making constraints. In practice, a smoothed score is computed by linearly combining the scores of the last *w* (one-second length) segments, weighting them according to rectangular (uniform) or triangular (linearly decreasing as going back in time) functions.

3. Experimental setup

3.1. The AMI Corpus

Experiments were carried out on the AMI Corpus of meeting conversations (http://corpus.amiproject.org/). The AMI Corpus is a multimodal dataset concerned with real-time human interaction in the context of smart meeting rooms. Data, collected in three instrumented meeting rooms, include a range of synchronized audio and video recordings. Meetings contain speech in English, mostly from non native speakers.

In this work, data for train, development and evaluation of speaker tracking systems were taken from a subset of the AMI Corpus, the Edinburgh scenario meetings, including 15 sessions: ES2002-ES2016, with four meetings per session, each meeting being half an hour long on average. The audio stream is obtained by mixing the signals from the headset microphones of the speakers. Three of the four speakers participating in each session are taken as target speakers, the remaining one being assigned the role of impostor. Careful impostor selection –not random– is made to account for gender unbalanced sessions. In sessions containing just one female speaker, the impostor is forced to be male (and vice versa), in order to avoid that gender favors impostor discrimination.

In order to assess the speaker tracking performance in realistic conditions, two independent subsets are defined, consisting of different sessions (and therefore different speakers), for development and evaluation purposes, respectively. The development set, consisting of 8 sessions (32 meetings), is used to tune the configuration parameters of the speaker tracking systems. The evaluation set, including the remaining 7 sessions (28 meetings), is used only to evaluate the performance of the previously tuned speaker tracking systems. Both the development and evaluation subsets are further divided into train and test datasets. Two meetings per session are randomly selected to estimate the UBM and the speaker models, and the remaining two are left for testing purposes. Time references are based on manual annotations provided in the AMI Corpus.

3.2. Performance measures

The performance of speaker tracking systems is commonly analyzed by means of *Detection Error Tradeoff* (DET) plots [14]. Performance is measured in terms of time that is correctly or incorrectly classified as belonging to a target. Therefore, miss and false alarm rates are computed as a function of time [1] and not as a function of trial number, like in speaker detection experiments. DET performance can be summarized in a single figure by means of the *Equal Error*

Rate (EER), the point of the DET curve at which miss and false alarm rates are equal. Obviously, the lower the EER, the higher the accuracy of a speaker tracking system.
Another way to summarize in a single figure the performance of a speaker tracking system is the so called *F-measure*, defined as follows:

$$F = \frac{2 \cdot P \cdot R}{P + R} \quad (3)$$

where precision (P) and recall (R) are related to false alarm and miss rates respectively. Precision measures the correctly detected target time from the total target time detected. Recall computes the correctly detected target time from the actual target time. The F-measure ranges from 0 to 1, with higher values indicating better performance. Collar periods of 250 milliseconds at the end of speaker turns are ignored for scoring purposes. Thus, speaker turns of less than 0.5 seconds are not scored.

4. Speaker tracking experiments

4.1. Online vs. offline systems under the speaker identification approach

Under the speaker identification approach, speaker overlaps cannot be detected, so all the segments containing speech from two or more speakers are removed when scoring test meetings. As expected, the classical offline system outperformed the proposed low-latency online system, but the performance of the latter was quite good, yielding only a 10.85% relative degradation (from 19.07 to 21.14% EER).

4.2. The effect of the speaker detection approach and score calibration

Attending to DET curves (not shown here for a lack of space) and EER, the system applying the speaker verification (SV) approach outperformed that applying the speaker identification (SI) approach. Using uncalibrated scores, the EER was 21.14% for the SI system and 12.03% for the SV system. But attending to the F-measure, the SI system outperformed the SV system (see Table 1). How may this be possible?
The DET curve is a very valuable means to compare the global discrimination capability of several speaker detection systems by presenting them the same set of trials. Note, however, that the sets of trials considered in DET curves for the SI and SV systems were different. The SV system considered as many trials as target speakers per test utterance (meaning that the same test utterance was evaluated many times), whereas the SI system considered a single trial per test utterance. Therefore, DET curves of SV systems were computed on much more trials than those of SI systems, and comparing them makes no sense. SV systems featured a high number of impostor trials. Since most of them were rejected, false alarm rates resulted remarkably lower than those of SI systems, thus yielding a better performance.
On the other hand, the F-measure is not defined in terms of trials but in terms of the time that was correctly detected. Therefore the F scores of SI and SV systems can be directly compared. Since the F scores of SI systems were better than those of SV systems, we conclude that SI systems outperform SV systems on the speaker tracking task defined on the AMI Corpus.
The F scores presented in Table 1 correspond to the operation points (thresholds) considered optimal in the DET curve. The threshold used for calibrated scores is based on application-dependent costs and target priors, which are adjusted using the development corpus. For uncalibrated scores, the threshold is fixed to zero, i.e. a target speaker is detected if the likelihood of the null hypothesis is higher than that of the alternative hypothesis. Results in Table 1 demonstrate the usefulness of the calibration stage, which leads to better performance in all the cases. The relative improvement is higher for SV systems, because calibration can compensate for the high number of false alarms at the cost of some misses.

Table 1. Precision, Recall and F-measure of SI and SV online speaker tracking systems in experiments on the evaluation set of the AMI Corpus.

	Uncalibrated			***Calibrated***		
	precision	***recall***	***F***	***precision***	***recall***	***F***
SI	0.71	0.91	0.80	0.81	0.85	0.83
SV-ExcOvlp	0.49	0.96	0.65	0.76	0.83	0.80
SV-IncOvlp	0.44	0.96	0.60	0.72	0.81	0.76

In the case of SV systems, which could theoretically detect various speakers at the same time, scores were computed either excluding or including overlapped segments. Both results (SV-ExclOvlp and SV-InclOvlp) are presented in Table 1. As expected, the performance of the SV-InclOvlp system was worse than that of the SV-ExclOvlp system: 7.69% worse when using uncalibrated scores, and 5% worse when using calibrated scores.

4.3. The effect of smoothing scores

The optimal *w* for the smoothing functions (which somehow depends on the average length of speaker turns) was heuristically determined on the development set. For the rectangular function, the optimal value was *w=2*. For the triangular function, it was *w=3*. Smoothing the scores consistently improved the speaker tracking performance on the test set of the AMI Corpus, the EER decreasing from 21.14% (no smoothing, *w=1*) to 19.37% (rectangular, *w=2*) and 18.64% (triangular, *w=3*), respectively. In terms of F-measure, a relative improvement of 3.61% was observed, from F=0.83 (no smothing, *w=1*) to F=0.86 (triangular, *w=3*).

5. SOA-compliant services

Two SOA-compliant services have been designed and implemented as the core elements for the deployment of SOA-based speaker tracking applications: a speaker tracking service (STservice) and a lower-level auxiliary speaker detection service (SDservice).
The SDservice captures the audio stream from an audio source (a field microphone integrated in the AmI environment) and outputs the likelihood scores of the target speakers at fixed-length (typically, one second) intervals, based on the analysis of the most recent window of speech. The SDservice performs feature extraction, speaker detection (based on target speaker models) and score calibration (based on a linear transform, optimized on a development corpus).
Taking advantage of service composition, a speaker tracking service (STservice) has been also designed which outputs actual speaker tracking decisions based on the outputs (detection scores) received from the SDservice. Decisions may be taken following either the speaker identification or the speaker verification approaches described in section 2. When performing speaker identification, the STservice outputs the identity of the most likely speaker. When performing speaker verification, several speakers can be detected simultaneously; on the other hand, if none of the scores is higher than the verification threshold, the STservice will output an impostor identifier.

The detection criterion (identification or verification) applied in making decisions, as well as the use of score calibration and the use of score smoothing, are optional features and depend on configuration parameters of the SDservice and the STservice, which can be updated from the speaker tracking application.

In practice, a speaker tracking application will invoke one STservice instance for each audio source (microphone) detected in the environment. As noted above, configuration parameters are determined by the application and depend on the scenario: decisions may be based on speaker verification if overlapped speech is allowed; smoothing based on past scores is activated if tracking robustness has to be increased, etc. Each STservice instance invokes a SDservice which continuously captures an audio stream and outputs a detection score per target speaker. The interaction between services and applications is based on subscriptions. Event subscription allows to get SDservice detection scores in a STservice instance, and to get STservice decisions in a speaker tracking application. Currently, the SOA based speaker tracking application is implemented following the UPnP standard.

The SDservice could be further composed by an audio capturing service which could be reused from many speech-based services and applications, such as speech recognition, language identification, etc. The SDservice could be also invoked for speaker adaptation in speech recognition tasks.

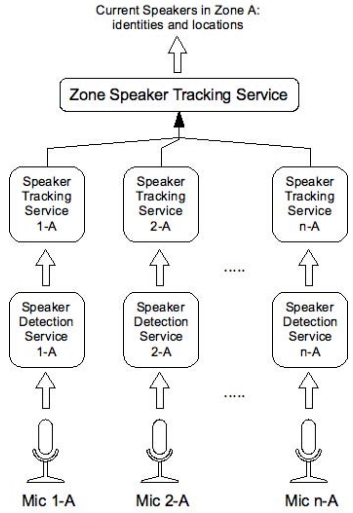

Figure 1. Definition of a Zone Speaker Tracking service.

Finally, a Zone Speaker Tracking service (ZSTservice) has been also defined, which assumes that AmI environments may be divided in separate spaces (e.g. rooms), and various microphones installed in each space. As shown in Figure 3, STservices can be organized hierarchically and their information gathered and interpreted by one ZSTservice. This way, by defining various ZSTservices (one per room), a high level management service could easily locate users in the AmI environment and follow their movements to customize services as users change their location.

6. Conclusions

A low-latency online speaker tracking approach and two SOA-compliant services, based on such approach, have been presented in this paper. Both elements have been designed for an Ambient Intelligence scenario with few users. The online speaker tracking system processes continuous audio streams and outputs a speaker identification decision for fixed-length (one second) segments. Speaker detection is done by means of a MAP-UBM speaker verification backend.

The proposed system was compared to a traditional system working offline, in experiments on a subset of the AMI Corpus of meeting conversations. Though offline segmentation of audio streams led to better results than using fixed-length segments, depending on the scenario and the required latency, offline audio segmentation may be unfeasible. The proposed approach provides low-latency online speaker tracking with little performance degradation.

To increase the robustness to local variability, a simple smoothing scheme was applied, consisting on a linear combination of the current score and a number of past scores. Promising results have been obtained in preliminary experiments.

Finally, two SOA-compliant services have been defined and implemented using UPnP, for speaker detection and tracking on a single audio source. A Zone Speaker Tracking Service has been also defined, which illustrates how those services could be connected and integrated in a home environment.

7. Acknowledgements

This work has been partially funded by the Government of the Basque Country, under program SAIOTEK, project S-PE09UN47; and the Spanish MICINN, under PN-I+D+i, project TIN2009-07446.

8. References

[1] A.F. Martin and M.A. Przybocki, "Speaker Recognition in a Multi-Speaker Environment", in Proceedings of European Conference on Speech Communications, Denmark, 2001.

[2] S.E. Tranter and D.A. Reynolds, "An Overview of Automatic Speaker Diarization Systems", IEEE Transactions on Audio, Speech and Language Processing, 14(5): 1557-1565, 2006.

[3] S. Meignier, D. Moraru, C. Fredouille, J.F. Bonastre and L. Besacier, "Step-by-step and Integrated Approaches in Broadcast News Speaker Diarization", Computer Speech and Language, 20:303-330, 2006.

[4] M. Kotti, V. Moschou and C. Kotropoulos, "Speaker Segmentation and Clustering", Signal Processing, 88: 1091-1124, 2008.

[5] D. Moraru, M. Ben and G. Gravier, "Experiments on Speaker Tracking and Segmentation in Radio Broadcast News", in Proceedings of the ICSLP, Lisboa, 2005.

[6] D. Istrate, N. Scheffer, C. Fredouille and J.F. Bonastre, "Broadcast News Speaker Tracking for ESTER 2005 Campaign", in Proceedings of the ICSLP, Lisboa, 2005.

[7] J.F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin and C. Wellekens, "A Speaker Tracking System based on Speaker Turn Detection for NIST Evaluation", in Proceeding of the IEEE ICASSP, Istanbul, Turkey, 2000.

[8] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus". Language Resources and Evaluation Journal , 41(2): 181-190, 2007.

[9] Reference Model for Service Oriented Architecture. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[10] Reference Architecture for Service Oriented Architecture. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf

[11] L.J. Rodríguez, M. Peñagarikano and G. Bordel, "A Simple But Effective Approach to Speaker Tracking in Broadcast News", Pattern Recognition and Image Analysis , LNCS 4478: 48-55, Springer-Verlag, 2007.

[12] D.A. Reynolds, T.F. Quatieri and R.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", Digital Signal Processing, 10:19-41, 2000.

[13] N. Brummer and J. du Preez, "Application Independent Evaluation of Speaker Detection", Computer Speech and Language, 20:230-275, 2006.

[14] A.F. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET curve in assessment of detection task performance", in Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech), Vol. 4, pp. 1895-1898, 1997.