16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

# UNSUPERVISED SPEAKER TRACKING IN A SPEECH RECOGNITION MODULE FOR MULTI-PARTY HUMAN-COMPUTER DIALOGUE

*Vladimir Popescu*[1, 2]*, Corneliu Burileanu*[2]*, and Jean Caelen*[1]

[1]Laboratoire d'Informatique de Grenoble, Grenoble Institute of Technology, France
email: {vladimir.popescu, jean.caelen}@imag.fr
[2]Faculty of Electronics, Telecommunications and Information Technology,
University "Politehnica" of Bucharest, Romania
email: cburileanu@messnet.pub.ro

## ABSTRACT

*Multi-party spoken dialogue systems are yet to be deployed in real applications, since several issues need to be mitigated, e.g. spontaneous speech, reliable voice activity detection (because of user barge-in), and real-time operation. For dialogues between the computer and several users at the same time, speaker tracking is needed, in order to ensure an appropriate analysis of the input speech. This paper addresses precisely this issue: starting from a speaker independent speech recognizer, we clone and adapt this system to each new input utterance, via unsupervised MLLR (Maximum Likelihood Linear Regression). Then, by taking into account the recognition confidence scores obtained by the speaker independent and speaker adapted recognizers, for each utterance, we retain a number of adapted systems, that model the speakers. Unlike in speaker tracking of "offline" multimedia content, in multi-party dialogue the data are not priorly available and the number and features of the speakers are not priorly known; moreover, runtime constraints apply, for ergonomic reasons. The proposed speaker tracking procedure is evaluated in the context of a book reservation service-oriented application, in Romanian language.*

## 1. INTRODUCTION

Human-computer dialogue is already a rather mature research field [5] that already stemmed to several commercial applications, either service or task-oriented [6]. Nevertheless, several issues remain to be tackled, when unrestricted, spontaneous dialogue is concerned: barge-in (when users interrupt the system or interrupt each other) must be properly handled, hence Voice Activity Detection is a crucial point [8]. Moreover, when multi-party interactions are allowed (i.e. the machine engages simultaneously in dialogue with several users), supplementary robustness constraints occur: the speakers have to be properly tracked, so that each utterance is mapped to a certain speaker that had produced it. This is needed in order to perform a reliable analysis of input utterances [1].

As for the current state of the art regarding speaker tracking, most of the work is related to speaker segmentation and/or indexing of offline multimedia content (or recorded meetings); in that case, the task is eased by several points: meetings usually take place indoor, speakers have rather fixed positions, their number is rather constant throughout the meeting [9]. Thus, one of the few previous works on multi-party dialogue segmentation assigns turn taking likelihoods to a language model that reflects the nature of the conversations [8]; two algorithms run in parallel for speaker and speech content estimation on TV sports news. Hence, dialogue issues are not directly considered, since *enough* data is available offline and runtime constraints do not apply. Another strand of research stems at performing both environment (i.e. noise features) adaptation and speaker adaptation and tracking, in pre-recorded meetings as well [9], [10]. For example, in [10], an unsupervised speaker adaptation in noisy environments is performed, in order to segment recorded meetings, where usually several microphones (microphone arrays) exist and the relative positions of speakers with respect to the microphones can be exploited [7]. Usually, the approaches adopted in this context start from GMM (Gaussian Mixture Models)-based speaker identification systems, that are coupled with HMM (Hidden Markov Models)-based speech recognition systems [9], [10]. Concerning the microphone arrays approach, it usually relies on cross-correlations computed on signals coming from pairs of acoustic sensors [7]. However, none of these procedures apply to service-oriented dialogue applications, since they usually involve outdoor processing, where non-relevant speech signals exist as well and the geometry of the users positions with respect to the acoustic environment is not too much controllable [1], [6]. Moreover, there is another research strand that relies on multimodal input for speaker tracking, e.g. combining acoustics with vision [4]. The research closest to ours was pursued by Furui and colleagues [12], where one proposes an unsupervised, on line and incremental speaker adaptation methods that improves the performance of speech recognizers when there are *frequent* changes in speaker identities and each speaker produces a series of several utterances. Basically, an HMM based scheme is proposed, where the likelihood given by a speaker independent decoder is compared to the scores given by speaker adapted HMMs. The adaptation is achieved using the MLLR method; then, speaker tracking supposes simply recognizing an utterance with a speaker independent system and with a set of speaker adapted decoders and comparing the likelihoods yielded by these processes.

In this paper we propose an "on the fly" unsupervised MLLR adaptation to speakers, where we derive a decision tree based on speech recognition scores at utterance level. This decision tree is then used to cluster utterances into speaker identities. An important point to emphasize is that the clustering process is performed *on-line*, for each new user utterance, but relying on previous utterances as well. The novelty of the method proposed in this paper resides in that we rely only on "classical" unsupervised MLLR adapta-

tion, but through a careful handling of confidence scores and decision-making based on these. Moreover, the computational simplicity (essentially assuming a top-down left-right traversal of a decision tree) is suited for dialogue applications where real-time operation is an important constraint.

Essentially, the proposed algorithm consists in using a speaker independent speech recognizer that is adapted (via MLLR, in an unsupervised manner) to new utterances in dialogue. Hence, for each new utterance, a speaker dependent system, adapted to a particular voice, is obtained. Then, for each subsequent utterance, log-likelihood scores obtained with the speaker independent decoded and the speaker dependent systems are compared: if the best score is given by one of the speaker dependent systems, then we decide that the producer of the utterance is the speaker to whom that speaker dependent system is adapted. Otherwise, if the best score is given by the speaker independent system, then we decide that a new speaker produced the utterance and hence we adapt the speaker independent decoder to this new utterance (actually, to the voice of its "producer"). Moreover, if there are two or more speaker dependent (i.e. adapted) systems that give approximately the same *best* scores, then we decide that these systems need to be further adapted (to this new utterance), in order to study the evolution of the recognition score: if only one of the *re*-adapted systems now gives the best score, we decide that the speaker whose voice is modeled by this system produced the utterance. Otherwise, if there still are several re-adapted systems that give the best score, then we decide that the utterance was produced by a new speaker and hence we adapt the speaker independent system to this new voice.

The paper is structured as follows: the next section presents the speaker tracking algorithm, emphasizing at the same time the information structures used, and the decision-making process; the third section presents a series of experiments performed on a dialogues database in Romanian language, along with results obtained and remarks on these; the fourth section concludes the paper and provides pointers to further refinements to the procedure.

## 2. UNSUPERVISED SPEAKER TRACKING ALGORITHM

### 2.1 Information Structures

The speaker tracking algorithm proposed in this paper consists essentially in adapting a speaker-independent speech recognition system to each new utterance, then clustering these adapted systems into a more restrained set, denoting the speakers in multi-party conversation. The adaptation process is represented by an unsupervised MLLR adaptation of a set of speaker-independent HMMs, whereas the clustering is based on the top-down traversal of a decision tree involving the utterance-level log-likelihood scores obtained in speech recognition.

The inputs to the algorithm consist in:

- a set of speaker-independent trained HMMs (at a word, triphone or phoneme level); these are denoted by the system $S_0$;
- a set of acoustic features extracted from a test speech signal; such an utterance is denoted by $\varepsilon_i$, for the $i$-th user utterance in dialogue.

The output of the algorithm consists in an assignment of a speaker identifier to the input utterance. As for the intermediary information structures used, these consist in confidence scores obtained for each acoustic unit that occurs in an utterance; these scores are then averaged and the value obtained is denoted by $\sigma_{0i}$, for the $i$-th utterance and the system $S_0$. Another valuable set of intermediary data structures is represented by the MLLR transformation matrices [3], one matrix for each new adapted system. A system adapted to an utterance $\varepsilon_i$ is obtained from $S_0$ via unsupervised MLLR using this utterance; hence, such a system consists in the original HMM set ($S_0$) together with the transformation matrix for the MLLR adaptation to $\varepsilon_i$, and is denoted by $S_{ai}$.

In order to avoid confusion and to better settle the purview of our work, we state that by "utterance" we understand, throughout the paper, a sequence of acoustic features that represent *one* speech turn produced by *one* speaker in dialogue. Obviously, this definition restricts the situations where our algorithm works, to the case where speakers' turns do not overlap in dialogue: thus, *cocktail party effects* [3] cannot be handled by the present version of the algorithm: our speaker tracking procedure only works if the speakers' utterances are disjoint (in time).

### 2.2 Decision Tree Traversal

As stated before, the speaker tracking algorithm uses the information structures described in Section 2.1 by constructing a fixed decision tree and then traversing it accordingly. The tree is specified offline, whereas its traversal depends on the confidence score obtained in recognizing the input utterances; thus, the procedure goes as follows (numbers indicate successive steps, while letters mark alternative paths; the $\approx$ sign between two recognition scores denotes an equality up to a difference of $\pm$ 5 % of the mean value of the two scores):

1. start with the speaker-independent speech recognition system $S_0$, a total number of utterances $N \leftarrow 0$ and of speakers $L \leftarrow 0$;
2. for an input utterance $\varepsilon_1$:
   - 2.1 perform unsupervised MLLR of $S_0$ on $\varepsilon_1$, obtaining the adapted system, $S_{a1}$;
   - 2.2 perform speech recognition of $\varepsilon_1$, with both $S_0$ and $S_{a1}$; two recognition scores $\sigma_{01}$ and $\sigma_{a11}$ result, respectively; from the definition of MLLR, we should have that $\sigma_{a11} > \sigma_{01}$; we mark that $\varepsilon_1$ has been produced by the speaker $l_1$;
   - 2.3 $N \leftarrow N+1, L \leftarrow L+1$;
3. for a new utterance $\varepsilon_m$ (with $m \geq 2$), assuming that we have $1+W$ speech recognition systems, $S_0$, $S_{a1}$, ..., $S_{aW}$ built as above, perform speech recognition on $\varepsilon_m$, with all the $1+W$ systems, obtaining the scores: $\sigma_{0m}$, $\sigma_{a1m}$, ..., $\sigma_{aWm}$; we can have one of the following possibilities:
   - (a) if $\sigma_{0m} > \max_{i=1,...,W}(\sigma_{aim})$, then $\varepsilon_m$ has been produced by a new speaker, $l_{L+1}$, different from the already detected $L$ speakers; in that case, we perform unsupervised MLLR of $S_0$ on $\varepsilon_m$, obtaining a new system $S_{a(W+1)m}$; we perform $L \leftarrow L+1$ and $N \leftarrow N+1$ as well (actually, $m = N+1$);
   - (b) else, if there exists an $i \in \{1,...,W\}$ so that $\sigma_{aim} > \max(\sigma_{0m}, \sigma_{a1m}, ..., \sigma_{aWm})$, then $\varepsilon_m$ has been produced by the emitter of a preceding utterance $\varepsilon_i$, with $i < m$; in this case, $L$ remains unchanged and $N$ gets incremented by one, to obtain $m$;
   - (c) else, if there exists a $k$ in $\{1,...,W\}$ such that $\sigma_{akm} \approx$

$\sigma_{0m}$, then we have an error, from the definition of MLLR;

(d) else, if there exist $j$ and $k$ in $\{1,...,W\}$ so that $j \neq k$ and $\sigma_{ajm} \approx \sigma_{akm} > \max(\sigma_{0m}, \max_{t \neq j,k}(\sigma_{atm}))$; in this case:

3.1 perform unsupervised MLLR of $S_{aj}$ and $S_{ak}$ on $\varepsilon_m$, obtaining the systems $\tilde{S_{aj}}$ and $\tilde{S_{ak}}$, respectively;

3.2 perform speech recognition with $\tilde{S}_{aj}$ and $\tilde{S}_{ak}$ on the utterance $\varepsilon_m$; the scores $\tilde{\sigma}_{ajm}$ and, respectively, $\tilde{\sigma}_{akm}$ are obtained; in this point, two situations are possible:

(a) if $\tilde{\sigma}_{ajm} \approx \tilde{\sigma}_{akm}$ and $\tilde{\sigma}_{ajm} \geq \sigma_{ajm}$ and $\tilde{\sigma}_{akm} \geq \sigma_{akm}$, then $S_{aj} \equiv S_{ak}$ and $\varepsilon_m$ has been produced by the emitter of $\varepsilon_j$ *and* $\varepsilon_k$; in this case, discard $S_{ak}$ and $L \leftarrow L-1$, $N \leftarrow N+1$;

(b) else, if $\tilde{\sigma}_{ajm} > \tilde{\sigma}_{akm}$ or $\tilde{\sigma}_{akm} > \tilde{\sigma}_{ajm}$, then denote by $j_0$ the index of the maximal score: $j_0 = \arg\max(\tilde{\sigma}_{ajm}, \tilde{\sigma}_{akm})$:

(i) if $\tilde{\sigma}_{aj_0m} \geq \sigma_{aj_0m}$, then $\varepsilon_m$ has been produced by the same speaker as $\varepsilon_{j_0}$ (the utterance used to obtained the system $S_{aj_0}$); in this case, keep $L$ unchanged and $N \leftarrow N+1$;

(ii) else, $\varepsilon_m$ has been produced by a new speaker, which is neither the producer of $\varepsilon_j$, nor the producer of $\varepsilon_k$; in this case, $L \leftarrow L+1$ and perform an unsupervised MLLR of $S_0$ on $\varepsilon_m$;

4. while there is an input utterance, go to step 3;
5. for $i$ from 1 to $N$, return the identifier of the speaker that produced utterance $\varepsilon_i$.

In this algorithm, the decision tree is constituted by the "if" alternatives at steps 3.(d).3.2, and 3.2.(b); the depths of the leaves are given by the nesting levels in the algorithm. The bottom-up traversal of the tree is inherently given by the nestings in the algorithm, whereas the left-right traversal is given by the order of the clauses: first, steps 1. and 2. are executed, then, the loop in steps 3.-4. In Figure 1 this tree is represented; dotted arrows indicate the flow of the algorithm and the continuous lines mark alternative possibilities (the intersection of a set of such lines is a decision point). The rest of the symbols mimic those used in the specification of the algorithm; the tree should be read top-down, left-right.

As for the reliability of the algorithm, one objection might be that the variations in recognition scores can be induced by the variations in the content of the utterances used in adaptation or in recognition. However, this apparent problem is reduced by the fact that each comparison is performed between scores obtained on the *same* utterance, although the systems used for this might or might not have used the same utterance in training (or adaptation). If we had compared two scores obtained with two systems that had not used the same utterance set in training or adaptation, we could have had results "corrupted", i.e. the scores reflect rather the differences in utterances than the differences in speakers. The answer to this is that the scores are computed by averaging at the utterance level the scores obtained for each acoustic unit (e.g. word, triphone, phoneme); hence, the scores that are compared might depend only on the particular distribution of the acoustic units within the utterance, being independent of the length of that utterance. Moreover, if the speaker-independent system is well trained, then the scores for the individual acoustic units exhibit low variance from one unit to another; if this variance is smaller than the difference between scores obtained on utterances from different speakers, then the problem is alleviated.

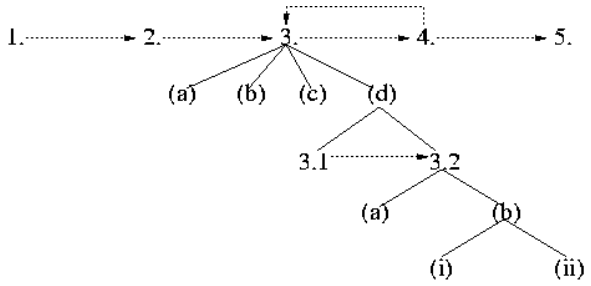


Figure 1: Recognition score-based decision algorithm.

Concerning the complexity of the algorithm, expressed in terms of the number of speech recognition processes (performed as Viterbi decoding [3]), the number of MLLR adaptation processes, the number $N$ of utterances in dialogue, and the number of speakers (denoted by $\overline{L}$), an estimate is provided here. Thus, considering the worst-case scenario where all the branches in the tree are visited, and denoting by $\tau_{MLLR}$ the average time needed for a MLLR adaptation process, by $\tau_{ASR}$, the average time needed for a speech recognition process, and by $\tau_{CMP}$, the time required for a comparison, we obtain that the execution time of the algorithm has an expression of the form ($\alpha$, $\beta$, $\alpha'$ and $\beta'$ indicate constant non-zero real numbers):

$$T = \tau_{MLLR} \times (\alpha\overline{L} + \beta N) + (\tau_{ASR} + \tau_{CMP}) \times (\alpha'\overline{L}^2 + \beta'\overline{L}N).$$

Therefore, the algorithm is quadratic in $\overline{L}$ and linear in $N$, for a specified $\overline{L}$ which becomes a constant for a running instance of the algorithm[1].

As compared to the work of Furui and colleagues [12], in our procedure we have introduced an essential supplementary step: when recognition scores obtained with two or more speaker adapted decoders are identical (up to a slight difference, less than 5 % of the average value of the scores), then these systems are *further* adapted to the current input utterance and the variations of the recognition likelihoods are studied (step 3.(d)3.1).

## 3. EXPERIMENTS

### 3.1 Speech Recognition System

The algorithm described in this paper was applied in a continuous speech recognition system, designed for "virtual librarian" multi-party dialogue applications, in Romanian language. The system was trained at word level, using no language modeling information. Thus, 92 words, related to library services were used, along with a supplementary set of 16 cue words; for each word a left-right (Bakis) [3] HMM was trained, with a variable number of states for each word (equal to 2 (initial and final, non-emissive states) + 3 $\times$ the number of phonemes in the word); the output observations are modeled with one Gaussian for each emissive state.

[1]This is true, since in multi-party human-computer dialogues the number of participants tends to stabilize in the early stages of the interaction.

Each word-level HMM was trained in a speaker-independent manner, using around 4 hours of recorded speech (in laboratory conditions: SNR $\geq$ 25 dB), containing these words, uttered in context. The acoustic characteristics of the training data are (i) acquisition: unidirectional head-set microphone; (ii) sampling frequency: 16 kHz; (iii) signal frame size: 300 samples; (iv) weighting window type: Hamming; (v) parameterization: 12 MFCC (mel frequency cepstrum coefficients) per frame, along with the energy and with the first and second-order derivatives of these features; this results in a total of 39 acoustic features per frame.

The training was performed in two steps, following a classical "isolated unit training" strategy, based on hand-labeled data (at a word level)[2]:

1. the parameters of the set of *prototype* word-level HMMs[3] are initialized through Viterbi alignment to the data [11];
2. the parameters of the initialized HMMs are re-estimated via the Baum-Welch procedure [3].

The system achieves a word-level accuracy of around 79 % when tested against spontaneously-uttered speech produced (in laboratory conditions, akin to those used in training) by locutors not used in the training process; this relative low percentage can be explained through the spontaneous nature of the test utterances, where the word boundaries are not easy to localise and, moreover, words exhibit incomplete or altered acoustic realisation.

We stress on the fact that language modeling was not used in the system; we did not even use phonemic acoustic modeling, since the speech recognition system can be and is used as a word spotter for dialogue purposes: bearing in mind that the ultimate goal of speech recognition in dialogue contexts is to *analyze* the utterance from a semantic and pragmatic point of view [2], spotting words that are relevant to the task (that convey semantic information) and cue words (that convey discourse and pragmatic information) is more effective and efficient that full-blown continuous speech recognition [6], [5]. However, in the experiments described in this paper, the utterances (for training and testing) were chosen so that they contain only the words considered, and the system was used as a speech recognizer.

### 3.2 Speaker Tracking Results

The speaker tracking procedure was tested using a maximum of four speakers in multi-party dialogues. The dialogues are driven by a set of scenarios involving several typical tasks: (i) enrollment of a new customer of the library; (ii) request for books on a certain topic or subject; (iii) request for a specific book; (iv) restitution of a book; (v) payment of fines due to delays in book restitution.

The multi-party dialogues are constructed so that every possible speaker order and "weight" (in terms of number of turns per speaker / number of turns / conversation) is achieved; for the moment, given the fact that the entire dialogue system is, for the time being, a work in progress [2], the conversations are only between humans, out of which one plays the part of the librarian. Thus, a number of around 400 conversations were used for testing the ability of the algorithm to map utterances to the appropriate speaker identifiers. It is worth mentioning that the number of speakers (between two and four, the maximum, as stated above) and their utterances were not previously known to the system; moreover, the speakers used in testing were different from those used in training the speaker-independent speech recognizer described in Section 3.1.

Table 1: Confusion matrix for speaker identity assignments.

| | $M_1$ | $M_2$ | $M_3$ | $F_1$ |
|---|---|---|---|---|
| $M_1$ | **1216** | 192 | 180 | 12 |
| $M_2$ | 172 | **1280** | 144 | 4 |
| $M_3$ | 112 | 184 | **1296** | 8 |
| $F_1$ | 8 | 96 | 88 | **1408** |

Sociolinguistic evidence prove that spontaneous multi-party dialogues usually tend to involve at most 5 speakers [1]; for a greater number of speakers, we tend to have several independent dialogues, although the interlocutors might still share the same environment (table, desk, etc.). In order to test these evidence ourselves, we have considered a corpus of multi-party dialogues; the data consists in three vaudevilles written in the 19th century by Eugène Labiche (in French language): "The Jackpot", "The Railroads", and "The Martin Prize"[4]. The rationale behind this choice stems from the fact that these dialogues are performed using spontaneous speech and thus represent a good approximation of naturally occurring multi-party human dialogues and, moreover, multi-party human-computer dialogue corpora are not available, for the time being, in Romanian language. Thus, from the total number of 133 multi-party dialogues (i.e. scenes in the plays), 105 (i.e. around 80 %) have at most 5 dialogue partners, whereas the average number of dialogue partners is around 3.14 in these dialogues. Hence, the choice of at most four dialogue partners seems appropriate for spontaneous spoken multi-party dialogue situations.

A first performance figure in our experiment is given by evaluating the word recognition of a system that is adapted to a certain speaker via unsupervised MLLR, on an utterance of that speaker, versus the same measure obtained, in the same test conditions, with the speaker-independent system. Thus, the average word recognition rate on a set of utterances produced by a certain speaker (out of those used in testing) reaches more than 80 % for the adapted system, versus around 79 % for the speaker-independent system.

The most relevant performance measure of our algorithm is represented by the number of correct speaker identifier assignments for each utterance, divided by the total number of utterances (since each utterance has a speaker identifier associated with it). An average score of 81.2 % was obtained for this measure.

A further refined analysis of the performances of the algorithm can be emphasized, namely by showing the confusion matrix concerning the assignment of speaker identities to utterances. Thus, denoting by $M_1$, $M_2$ and $M_3$ the three male speakers, and by $F_1$ the female speaker used in testing the algorithm, the confusion matrix is shown in Table 1, for a series of around 400 dialogues, each one counting between 4 and 20 utterances (speech turns). In Table 1, the figures on the rows indicate the assignments performed by the algorithm.

[2]The labels include temporal information as well.

[3]The prototype HMMs are specified by the user and contain constraints on the size and topology of the models.

[4]The electronic versions of these plays were downloaded from http://fr.wikisource.org.

Table 2: Performance measures for every test speaker.

| | Precision | Recall | $F$-measure |
|---|---|---|---|
| $M_1$ | 0.81 | 0.76 | 0.78 |
| $M_2$ | 0.73 | 0.80 | 0.76 |
| $M_3$ | 0.76 | 0.81 | 0.78 |
| $F_1$ | 0.98 | 0.88 | 0.93 |

From this confusion matrix, *precision* (defined as the number of correct speaker assignments, divided by the total number of speaker assignments) and *recall* (defined as the number of correct speaker assignments, divided by the total number of real speaker-to-utterance mappings) can be derived, for each speaker and, consequently, the corresponding $F$-measures (defined as the harmonic mean of precision and recall). These quantities are shown in Table 2 for each of the four test speakers considered. We can see that these quantities are evenly balanced, although, as we could expect, the best results are obtained for the (only) female speaker. This shows, on the one hand, that the performances of the algorithm are robust to speaker variations, and, on the other hand, that the usage of only one female speaker introduces an artificial bias on the results. Therefore, the most relevant performance figures are those obtained for the male speakers. Moreover, we can see that there is a balance between precision and recall as well, which might be a hint that further tests are needed in order to see whether this is a feature of the algorithm, or of the test data.

Concerning the values of the recognition scores obtained, the word-level recognition log-likelihood scores evolve around $-2000$ for adapted systems recognizing utterances produced by the speakers that the systems are adapted to, and around $-3700$ for the non-adapted speaker-independent system. On the other hand, the word-level log-likelihood score variances are in the range of around $\pm 800$, thus, less than the difference between the first two average scores. Therefore, tests indicate that the usage of log-likelihood scores is a reliable strategy for speaker tracking.

Finally, it is worth noticing that step 3.(d)3.1 in the algorithm, which further adapts previously adapted decoders if recognition scores are (approximately) identical for at least two such adapted recognizers, proves very important when there are speakers whose voices are very similar, as it is the case in our experiment, with speakers $M_1$ and $M_2$. Indeed, if the further adaptation step were not used, we would have obtained precision figures of about 0.25 for these two speakers, as compared to the values of around 0.75, presented in Table 2 and obtained when further adaptation is performed.

## 4. CONCLUSIONS AND FURTHER WORK

In this paper we have presented a computationally-simple strategy for speaker tracking in multi-party human-computer dialogue. The approach is based on the traversal of a decision tree that relies on speech recognition scores and unsupervised MLLR adaptation of a speech recognition system to input utterance so that these scores are maximized. Thus, an algorithm linear in the number of previous utterances in dialogue is obtained, that achieves a speaker tracking performance of around 80 % on spontaneous speech. The algorithm has been tested in the context of a virtual librarian dialogue application, in Romanian language, and exhibits good performance.

In the near future, the speaker tracking strategy ought to be implemented as a stand-alone module (instead of the current scripts driving HTK tools) and coupled with a multi-party dialogue system, currently under development [2]. Moreover, the algorithm could then be coupled either with a word spotter and used as input to a speech understanding system, or with a triphone-based continuous speech recognition system for Romanian language. Concerning real-time constraints, strategies for optimizing the algorithm runtime (e.g. through parallel processing) are currently under development.

**Acknowledgement**

The research reported here was funded by the Romanian Government, under the National Research Authority grant IDEI no. 930/2007.

## REFERENCES

[1] H. Braningan, "Perspectives on Multi-party Dialogue", *Research on Language and Computation*, vol. 4, pp. 153–177, Springer, 2006.

[2] J. Caelen, Anne Xuereb, *Interaction et pragmatique*, Paris: Hermès Science, 2007.

[3] X. Huang, A. Acero, H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, New Jersey: Prentice Hall, 2001.

[4] F. Landragin, *Dialogue homme-machine multimodal*, Paris: Hermès Science, 2005.

[5] M. F. McTear, "Spoken Dialogue Technology: Enabling the Conversational Interface", *ACM Computing Surveys*, vol. 34, no. 1, pp. 90–169, March 2002.

[6] W. Minker and S. Bennacef, *Parole et dialogue homme-machine*, Paris: CNRS Editions, 2001.

[7] P. Motlicek, L. Burget, and J. Cernoký, "Non-parametric Speaker Turn Segmentation of Meeting Data ", in *Proc. EUROSPEECH 2005*, Lisbon, Portugal, 2005, pp. 657–660.

[8] N. Murani and T. Kobayashi, "Dictation of multiparty conversation considering speaker individuality and turn taking", *Systems and Computers in Japan*, vol. 34, no. 13, pp. 103–111, Wiley InterScience, 2003.

[9] S. Sato, H. Segi, K. Onoe, E. Miyasaka, H. Isono, T. Imai, and A. Ando, "Acoustic model adaptation by selective training using two-stage clustering", *Electronics and Communications in Japan*, vol. 88, no. 2, pp. 41-51, Wiley InterScience, 2004.

[10] M. Yamada, A. Baba, S. Yoshizawa, Y. Mera, A. Lee, H. Saruwatari, and K. Shikano, "Unsupervised acoustic model adaptation algorithm using MLLR in a noisy environment", *Electronics and Communications in Japan*, vol. 89, no. 3, pp. 48–58, Wiley InterScience, 2005.

[11] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and Ph. Woodland, *The HTK Book*, Cambridge University, United Kingdom, 2005.

[12] Z. Zhang, S. Furui, K. Ohtsuki, "On-line incremental speaker adaptation for broadcast news transcription", *Speech Communication*, vol. 37, pp. 271–281, 2002.