IDIAP RESEARCH REPORT

Role Recognition in Radio Programs using Social Affiliation Networks and Mixtures of Discrete Distributions: an Approach Inspired by Social Cognition

A.Vinciarelli [a] [b] and S.Favre [a] [b]

IDIAP–RR 07-40

September 2007

submitted for publication

[a] IDIAP - {vincia,sfavre}@idiap.ch
[b] Ecole Polytechnique Fédérale de Lausanne (EPFL) - 1015 Lausanne (Switzerland)

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny − Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Role Recognition in Radio Programs using Social Affiliation Networks and Mixtures of Discrete Distributions: an Approach Inspired by Social Cognition

A.Vinciarelli and S.Favre

September 2007

submitted for publication

Abstract. This paper presents an approach for the recognition of the roles played by speakers participating in radio programs. The approach is inspired by social cognition, i.e. by the way humans make sense of people they do not know, and it includes unsupervised speaker clustering performed with Hidden Markov Models, Social Network Analysis and Mixtures of Bernoulli and Multinomial Distributions. The experiments are performed over two corpora of radio programs for a total of around 45 hours of material. The results show that more than 80 percent of the data time can be labeled correctly in terms of role.

1 Introduction

In this work, we address the problem of recognizing the *role* of speakers in radio programs, i.e. of mapping the speakers into categories such as *anchorman* or *guest* which correspond to specific functions in broadcast data. The approach we propose is inspired by *social cognition*, the mental process that takes place unconsciously each time we *make sense of people*, i.e. each time we predict the behavior of people we know little or nothing about in order to select the most appropriate form of interaction, or non-interaction, with them [14].

The mechanisms underlying social cognition are still debated, but two points seem to be widely accepted: the first is that social cognition is a form of *categorical thinking* about others [16], i.e. it consists in mapping people into categories (or *stereotypes*) which reasonably complete the information we miss about them. The second is that social cognition is based on *relationship patterns* [9], i.e. it infers information from the relationships that individuals have with other, if possible better known, people.

The approach we propose is composed of two stages which correspond to the above two elements (see Figure 1): the first is the extraction of feature vectors accounting for people relationships, the second is the recognition algorithm mapping the feature vectors into categories corresponding to the roles. The feature extraction stage (left dotted box in Figure 1) starts by splitting the news bulletins into single speaker segments using an unsupervised clustering approach [23]. The output of the clustering is used to extract a Social Affiliation Network [27] and to model the distribution of the time associated to each role. Since the Social Affiliation Network captures the pattern of the interactions between the different speakers, this step corresponds to the use of relationships in social cognition.

The second stage (right dotted box in Figure 1) maps the feature vectors into six classes corresponding to six different roles. This task is performed using mixtures of Bernoulli and Multinomial distributions [6] trained using the Expectation-Maximization technique [15]. This step can be interpreted as a simulation of the categorical thinking in social cognition.

To our knowledge, this is one of the earliest approaches using Social Network Analysis to extract information about people in audio or video data. Other works using SNA for similar purposes include only an approach to identify the main characters in movies [29], and another role recognition approach based on Social Networks and Machine Learning techniques different from those presented in this work [25]. The experiments are performed over two corpora of radio programs for a total of around 45 hours of material. The results show that more than 80 percent of the data time can be labeled correctly in terms of role.

Role recognition can be useful in several applications: browsers can be enhanced by enabling users to select interventions corresponding to a given role, retrieval systems can use the role as a clue for filtering the results, summarization systems can use the role as a criterion for the selection of information rich data segments, etc.

The rest of the paper is organized as follows: Section 2 presents a survey on related work, Section 3 presents the interaction pattern extraction, Section 4 describes the role assignment technique, Section 5 presents experiments and results, and Section 6 draws some conclusions.

2 Related Work

To our knowledge, only few works have addressed the problem of role recognition so far and they can be split into two main groups: the first addresses data like TV and radio programs where the roles are defined a-priori and correspond to specific functions (e.g. *anchorman* or *interview participant*), the second addresses data like meeting recordings where the interactions are more spontaneous and the roles are defined according to sociological criteria (e.g. *attacker* or *neutral*). The first group includes the work in [4], where the roles are recognized through lexical specificities, and the approaches proposed in [25][24], where the roles are recognized using Social Networks and statistical modeling of network related features like the *centrality* [27]. The second group includes the works in [30][21], where features

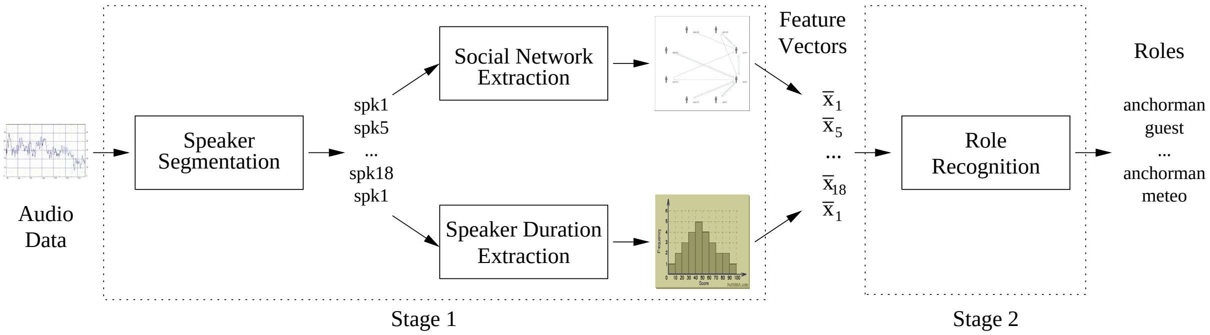

Figure 1: Role recognition approach. The picture show the two main stages of the approach: the feature extraction and the actual role recognition.

extracted from audio and video are fed to Support Vector Machines for role classification, and the approach proposed in [3], where the roles are detected through audio features and tree-based classifiers.

Many other works have aimed at recognizing *human behavior*, i.e. at understanding what people do in data captured through cameras, microphones, wearable devices and smart rooms [19]. Such works can be divided into two broad groups: the first includes the approaches trying to recognize the activity of a single individual, the second involves the techniques capturing the interaction between several persons.

There are two main ways of addressing activity recognition for single persons: the first is trying to recognize generic behaviors corresponding to sequences of movements, the second is to take into account the context of the actions and to consider not only person movements, but also data from the environment where the person operates. Examples of the first approach are mostly based on video analysis: the work in [5] represents the movements of legs, arms and other body parts using vectors and then models actions as vector sequences extracted from few video frames, the work in [7] uses Hidden Markov Models to segment videos into single action shots, the work in [8] uses 3-dimensional models of the human body to follow the sequence of steps composing a given action or behavior, the approach in [31] tries to avoid the definition of a predefined set of actions by clustering videos similar from the point of view of the displayed movements, and the work in [28] investigates the use of multiple cameras. Examples of the second approach for the recognition of single person activities take advantage from a wide spectrum of sensors: the work in [20] tries to predict the actions of a car driver by taking as input steering and acceleration data, the approach in [26] recognizes the actions of the workers on an assembly line by using worn accelerometers and microphones.

While single individual actions are recognized mostly through video analysis, collective actions are recognized in general through multiple sensors. The main reason is that the simultaneous location and tracking of different people is difficult in videos, while it is more simple using other kinds of sensors. Examples of video based collective action recognition use stochastic grammars modeling sequences of elementary movements and interactions [13][22]. Works involving multiple sensors include the following examples: the work in [11] describes the use of infrared cameras, the approach in [18] uses videos as well as activities such as calling or typing on the computer keyboard to recognize the interactions between work colleagues, and the work in [17] models jointly audio and video to recognize meeting collective events such as discussions, agreement or presentations.

3 Interaction Pattern Extraction

This section presents the technique used in this work to extract and represent the interaction pattern of each speaker. The technique includes two steps: the first is the segmentation of the recordings into single speaker segments, the second is the extraction of the corresponding Affiliation Network. The next two sections show the two steps in detail.

3.1 Speaker Clustering

This section provides a general description of the speaker clustering approach used in this work (for a full description see [2][1]).

The algorithm is based on an ergodic continuous density Hidden Markov Model (HMM) where each state corresponds to a cluster of observation vectors (see below) and, in principle, to a single speaker voice. The emission probability is modeled with Gaussian Mixture Models (GMM) [6], the observation vectors are 12 dimensional *Mel Frequency Cepstral Coefficients* (MFCC) vectors extracted every 10 ms from a 30 ms long window [12]. The reason is that these features are effective in speaker recognition tasks and seem to capture the characteristics of the voice [1].

The first step of the process is the initialization of the above HMM. The audio data is segmented into M uniform non-overlapping segments, where M is the initial number of states in the HMM and it is a number significantly higher than the expected number of speakers. The HMM is trained using the uniform segmentation as groundtruth and the result is a parameter set $\Theta^{(0)}$. The resulting HMM can be aligned with the data using the Viterbi algorithm to find the best sequence of states (i.e. speakers):

$$q^{(0)} = \arg\max_{q} p(q|O, \Theta^{(0)}) \tag{1}$$

where q is a sequence of states and $O = \{\vec{o}_1, \ldots, \vec{o}_K\}$ is the sequence of the observation vectors. The alignment results into a segmentation different from the uniform one used for the initialization. The HMM can thus be retrained and a new parameter set $\Theta^{(1)}$ is obtained:

$$\Theta^{(1)} = \arg\max_{\Theta} p(q^{(0)}|O, \Theta) \tag{2}$$

where $\Theta = \{\theta_1, \ldots, \theta_M\}$, i.e. the parameter set of the HMM, can be thought of as a set of GMM parameters.

Since the number M is higher than the expected number of speakers, the data is oversegmented and there are clusters that should be merged since they contain data belonging to the same speaker. For this reason, two states are merged when the following condition is met:

$$\log p(O_{m+n}|\theta_{m+n}) \geq \log p(O_m|\theta_m) + \log p(O_n|\theta_n) \tag{3}$$

where O_m, O_n and O_{m+n} are the observation vectors attributed to cluster m, n and their union respectively, θ_m and θ_n are the parameters of GMMs in states m and n and θ_{m+n} are the parameters of a GMM trained with Expectation-Maximization on O_{m+n}.

After the merging, the HMM has less states and it can be realigned with the data in order to obtain a new segmentation which can be used to train again the HMM. The new states satisfying the above condition will be thus merged again and the whole procedure will be iterated. The merging between states is performed by keeping constant the number of parameters:

$$|\theta_{m+n}| = |\theta_m| + |\theta_n|, \tag{4}$$

so the likelihood will not decrease just because the number of parameters gets lower, but rather it increases until the states that are merged actually correspond to the same or similar voices and it decreases when the states that are merged correspond to voices too different. This provides the stopping criterion for the iteration, in fact the alignment and training steps are repeated until the likelihood reaches its maximum. The segmentation corresponding to the likelihood maximum is retained as the result of the speaker clustering process.

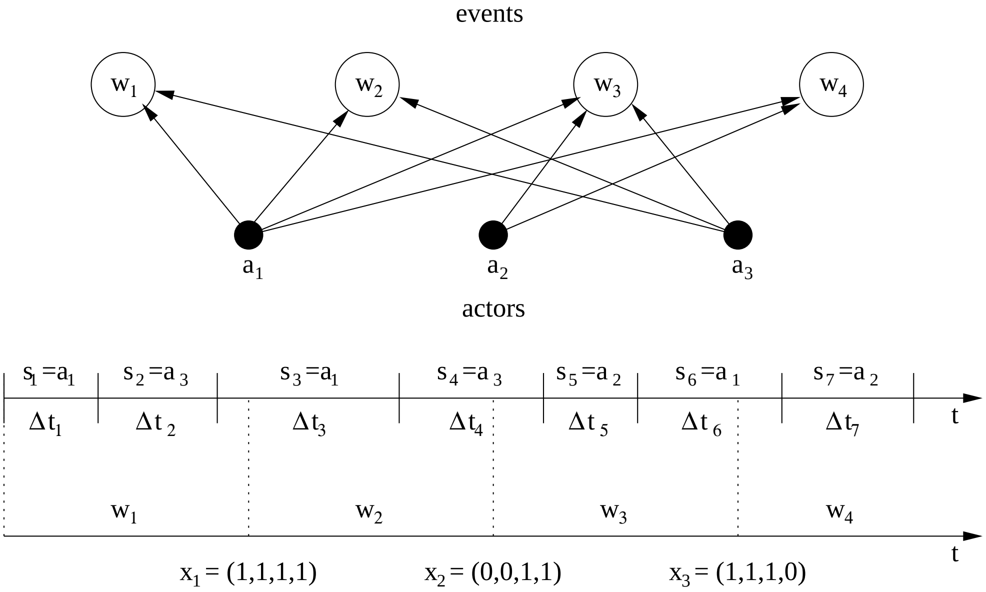

Figure 2: Interaction pattern extraction. The picture shows the Affiliation Network extracted from a speaker segmentation. The events of the network correspond to the windows w_j and the actors are linked to the events when they talk during the corresponding windows. The actors are represented using vectors $\vec{x}_i$ where the components account for the links between actors and events.

3.2 Affiliation Network Extraction

The result of the speaker clustering process is that each recording is split into a sequence $S = \{(\Delta t_i, s_i)\}$, where $i = 1, \ldots, |S|$, Δt_i is the duration of the i^{th} segment, and s_i is the speaker label of the i^{th} segment. The label s_i belongs to the set $A = \{a_1, \ldots, a_G\}$ of unique speaker labels output by the speaker clustering process (see lower part of Figure 2).

This information can be used to create an *Affiliation Network*, i.e. a Social Network where there are two classes of nodes: the *actors* and the *events* [27]. Actors can be linked to events, but no links are allowed between nodes of the same kind (see upper part of Figure 2). In our experiments, the actors correspond to the speakers in the broadcast news and the events correspond to uniform non-overlapping windows spanning the whole length of the recordings. The reason is that the network is expected to capture the relationships between the speakers and one of the most reliable evidences of interaction is the proximity in time [10]. In fact, two persons talking during the same window are more likely to interact with each other than two people talking in different windows.

One of the main advantages of this representation is that each actor a_i can be represented with a vector $\vec{x}_i$ where the component j accounts for the participation of a_i in the j^{th} event. In our experiments, we used two kinds of representation: in the first one, the j^{th} component is 1 if the speaker talks during the j^{th} window and 0 otherwise (the corresponding vectors are shown at the bottom of Figure 2). In the second the j^{th} component is the number of times that speaker a_i talks during the j^{th} window. In the first case the vectors are binary, in the second case they have integer components higher or equal to 0. In both cases, people that interact more with each other tend to talk during the same windows and are represented by similar vectors.

4 Role Recognition

This section describes the statistical foundations of the role assignment process used in our experiments.

Section 3 has shown that the relationship pattern of each speaker a_i is represented by a vector $\vec{x}_i = (x_{i1}, \dots, x_{iD})$, where D is the number of windows, that can have either binary or semidefinite positive integer components. Speaker a_i talks during a fraction τ_i of the total time of a bulletin and $\sum_{k=1}^{G} \tau_k = 1$, where G is the total number of speakers in the bulletin. In this way, each speaker is represented by a vector $\vec{y}_i = (\tau_i, \vec{x}_i)$.

Consider the vector $\vec{r} = (r_1, \dots, r_G)$, where r_i is the role of speaker a_i, and the set $Y = \{\vec{y}_1, \dots, \vec{y}_G\}$, where $\vec{y}_i$ is the vector representing speaker a_i. The problem of assigning the role to all speakers can be thought of as the maximization of the *a-posteriori* probability $p(\vec{r}|Y)$. By applying the Bayes Theorem and by keeping into account that $p(Y)$ is constant during the recognition the problem can be thought of as finding $\hat{\vec{r}}$ such that:

$$\hat{\vec{r}} = \arg \max_{\vec{r} \in \mathcal{R}^G} p(Y|\vec{r}) p(\vec{r}), \tag{5}$$

where $\mathcal{R}$ is the set of the predefined roles. In order to simplify the problem, we make the assumption that the roles of the different speakers are statistically independent and the above expression becomes:

$$\hat{\vec{r}} = \arg \max_{\vec{r} \in \mathcal{R}^G} \prod_{k=1}^{G} p(\vec{y}_k|r_k) p(r_k). \tag{6}$$

The maximization of the product can be thus achieved by maximizing separately each factor $p(\vec{y}_k|r_k)p(r_k)$.

In order to further simplify the problem, we assume that $\vec{x}_i$ and τ_i are statistically independent given the role, thus:

$$\hat{r}_i = \arg \max_{r \in \mathcal{R}} p(\vec{x}_i|r) p(\tau_i|r) p(r). \tag{7}$$

The problem left open is the estimation of the probabilities $p(\vec{x}|r)$, $p(\tau|r)$ and $p(r)$. This is the subject of the next three sections.

4.1 Modeling Binary Interaction Patterns

This section shows how we model the interaction patterns extracted from the Affiliation Networks when the components are binary. Given a labeled training set, there are N_r speakers playing the role r. Each one of them is represented by a binary vector $\vec{x}$. We estimate $p(\vec{x}|r)$ using mixtures of Bernoulli distributions (the dependence on the role r is omitted for simplicity) [6]:

$$p(\vec{x}|\vec{\mu}, \vec{\pi}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{D} \mu_{kj}^{x_j} (1 - \mu_{kj})^{1 - x_j} \tag{8}$$

where $\vec{\mu} = (\vec{\mu}_1, \dots, \vec{\mu}_K)$ is the concatenation of K vectors $\vec{\mu}_k$, $\vec{\pi} = (\pi_1, \dots, \pi_K)$ with the constraint $\sum_k \pi_k = 1$ is the vector of the mixing weights, D is the number of windows used to split the recordings, μ_{kj} is the component j of $\vec{\mu}_k$, and x_j is the component j of $\vec{x}$.

When $K = 1$, the Maximum-Likelihood (ML) estimate of the parameters has an analytical expression:

$$\mu_i = \frac{1}{N_r} \sum_{n=1}^{N_r} x_i^{(n)}. \tag{9}$$

The parameter μ_i is thus the average of the x_i values in the training set.

When $K > 1$, the maximization of the likelihood does not lead to closed form solutions and it is necessary to apply the Expectation-Maximization (EM) technique [6][15], an iterative procedure that leads to parameter estimates corresponding to a local maximum of the likelihood.

4.2 Modeling Multinomial Interaction Patterns

This sections shows how to model the vectors extracted from the Affiliation Networks when the components are integer higher or equal to 0. Given a vector $\vec{x} = (x_1, \dots, x_D)$, where D is the number of windows, each component x_i can be represented with a vector $\vec{z}_i$ defined as follows:

$$\vec{z}_i = (z_{i1}, \dots, z_{iN}), \tag{10}$$

where $z_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{N} z_{ij} = 1$. In other words, x_i is represented with a N-dimensional vector where all the components are 0 except one, i.e. the component $z_{in} = 1$, where n is the number of times that the actor represented by $\vec{x}$ talks during event i. As a result, $\vec{x}$ is represented with a concatenation of vectors $\vec{z} = (\vec{z}_1, \dots, \vec{z}_D)$. The vector $\vec{z}$ can thus be modeled with a multinomial distribution:

$$p(\vec{z}|\vec{\mu}) = \prod_{i=1}^{D} \prod_{j=1}^{N} \mu_{ij}^{z_{ij}}, \tag{11}$$

or with a mixture of multinomial distributions:

$$p(\vec{z}|\vec{\mu}, \vec{\pi}) = \sum_{k=1}^{K} \pi_k \prod_{i=1}^{D} \prod_{j=1}^{N} \mu_{kij}^{z_{ij}}. \tag{12}$$

The parameters $\vec{\mu}$ and $\vec{\pi}$ can be estimated by maximizing the likelihood over a training set $\mathcal{X}$. In the case of the single Multinomial, this leads to a closed form expression for the parameters:

$$\mu_{ij} = \frac{1}{N_r} \sum_{n=1}^{N_r} z_{ij}^{(n)}, \tag{13}$$

where N_r is the number of vectors corresponding to role r. In the case of the mixture, the maximization of the likelihood is performed using the Expectation-Maximization technique [6][15].

4.3 Modeling Durations

This section shows how we estimate the probabilities $p(\tau|r)$. Given a labeled training set, there is a number N_r of speakers playing role r. Each one of them accounts for a fraction $\tau^{(n)}$ of the bulletin he or she is involved in, where $n = 1, \dots, N_r$. We estimate $p(\tau|r)$ using a Gaussian Distribution $\mathcal{N}(\tau|\mu_r, \sigma_r)$, where μ_r and σ_r are mean and variance respectively. The Maximum Likelihood estimates of the parameters are sample mean:

$$\mu_r = \frac{1}{N_r} \sum_{n=1}^{N_r} \tau^{(n)} \tag{14}$$

and sample variance:

$$\sigma_r = \frac{1}{N_r} \sum_{n=1}^{N_r} (\tau^{(n)} - \mu_r)^2. \tag{15}$$

A different Gaussian distribution is obtained for each role.

4.4 Estimating Role Probabilities

This section shows how we estimate the probability $p(r)$ of a given role being observed. Given a labeled training set, the total duration of the recordings belonging to it is T, and the sum of the intervention lenghts of the speakers playing the role r is T_r. The probability $p(r)$ is estimated as follows:

$$p(r) = \frac{T_r}{T}, \tag{16}$$

i.e. as the fraction of training set that the role r accounts for.

Corpus	AM	SA	GT	IP	AB	MT
C1	41.2%	5.5%	34.8%	4.0%	7.1%	6.3%
C2	17.3%	10.3%	64.9%	0.0%	4.0%	1.7%

Table 1: Role fractions. The table reports the percentage of data time each role accounts for.

5 Experiments and Results

This section presents experiments and results obtained in this work. The next three sections describe data and roles, the performance measures and the role recognition results.

5.1 Data and Roles

The experiments of this work have been performed over two different corpora of broadcast news provided by *Radio Suisse Romande*, the French speaking Swiss National broadcasting service. The first corpus (referred to as C1 in the following) contains 96 news bulletins with an average length of 11 minutes and 50 seconds. The corpus contains all news bulletins broadcasted during February 2005 and can thus be considered a representative sample of this kind of programs. The second corpus (referred to as C2 in the following) contains 27 one hour long talk-shows called *Forum* and broadcasted during February 2005 (one recording has been lost for technical reasons). Also in this case, the corpus can be considered a representative sample of this specific kind of program.

The roles are the same for both C1 and C2: the *Anchorman* (AM), i.e. the person managing the program, the *Second Anchorman* (SA), i.e. the person supporting the AM, the *Guest* (GT), i.e. the person invited to report about a single and specific issue, the *Interview Participant* (IP), i.e. interviewees and interviewers, the *Abstract* (AB), i.e. the person reading a short abstract at the beginning of the program, and the *Meteo* (MT), i.e. the person reading the wheather forecasts. Table 1 shows the distribution of the data time across different roles. The distributions are significantly different in C1 and C2 and this enables us to show how robust is the role reognition approach with respect to such a characteristic.

5.2 Speaker Clustering Results

The relationship patterns used at the role assignment step are extracted from the speaker segmentation obtained with the clustering process. Errors in the clustering (inclusion of different speakers into a single cluster, or split of a single speaker into several clusters) lead to spurious interactions that can mislead the role assignment process.

The effectiveness of the clustering is measured with the *Purity* π, a metric showing on one hand to what extent all vectors corresponding to a given speaker are grouped into the same cluster, and on the other hand to what extent all vectors in a given cluster correspond to a single speaker. The Purity ranges between 0 and 1 (the higher the better) and it is the geometric mean of two terms: the *average cluster purity* π_c and the *average speaker purity* π_s. The definition of π_c is as follows:

$$\pi_c = \sum_{k=1}^{N_c} \sum_{l=1}^{N_s} \frac{n_k}{N} \frac{n_{lk}^2}{n_k^2}, \tag{17}$$

where N_s is the number of speakers, N_c is the number of clusters, n_{lk} is the number of vectors belonging to speaker l that have been attributed to cluster k, n_k is the number of feature vectors in cluster k and N is the total number of feature vectors. The definition of π_s is as follows:

$$\pi_s = \sum_{l=1}^{N_s} \sum_{k=1}^{N_c} \frac{n_l}{N} \frac{n_{lk}^2}{n_l^2} \tag{18}$$

(see above for the meaning of the symbols).

The application of the speaker clustering process requires the setting of the initial number of states M in the fully connected Hidden Markov Model (see Section 3). The value of M must be significantly higher than the number of expected speakers for the clustering process work correctly. In our experiments, we set *a-priori* $M = 30$ for C1 and $M = 90$ for C2 and not other values have been tested. The average purity is 0.81 for C1 and 0.79 for C2.

5.3 Role Recognition Results

The application of the role assignment algorithm requires to set two hyperparameters: the first is the number D of windows, the second is the number K of distributions in the mixtures. The value of D has been set to 15 for C1 and to 30 for C2. In both cases, the value of D has been set *a-priori* and no other values have been tested. For what concerns K, values between 1 and 5 have been tested showing that there are no major changes in the role recognition performance. For this reason, no crossvalidation has been performed in order to find the best value of K and the results show that the use of mixtures rather than single distributions does not really help in this case. In the following, the performance is measured with the *Accuracy* α, i.e. the percentage of data time correctly labeled in terms of role.

The experiments have been performed using a *leave-one-out* approach [6]: the models are trained using all the recordings in a given corpus except one which is used as a test set. Each recording in a corpus is used once as test set so it is possible to test the approach over all data at disposition.

Tables 2 and 3 report the role recognition results for corpora C1 and C2 respectively as a function of K. The distribution used to model the interaction patterns is indicated with B (Bernoulli) and M (multinomial). The best overall α is around 80 percent for both C1 and C2 independently of the corpus and this means that the role recognition approach is robust with respect to changes in the time distribution across the roles. This is important because the same role is played in different ways depending on the specific program and the approach seems to ba capable of adapting automatically to the different situations.

The multinomial is significanlty less effective over the C1 corpus. The loss is particularly evident in the case of the MT which accounts for most of the performance difference. The reason is probably that the MT is often accompanied by background music that induces errors in the clustering process. The MT interventions are thus split into several segments separated by spurious speakers: the number of times the MT talks during a window is thus multiplied and this creates a difference between test and training conditions (where the MT talks only once per window). The Bernoulli distribution is not sensitive to this effect because it does not take into account the number of times speakers talk, but only their presence or absence in a window.

However, for the rest of the roles and the data, the use of the multinomial rather than the Bernoulli distribution seems not to affect the performance. This means that the information about the number of times a speaker talks (conveyed by the multinomial) does not add information with respect to the simple absence or presence of the speakers in a given window (conveyed by the Bernoulli distribution). This is not surprising because the important aspect for the role recognition, at least in this case, is the fact that two people interact and not how they interact. In other words, the results do not change if two people interact through just one question and one answer or through a discussion including a large number of interventions.

The use of mixtures does not improve the Accuracy. This might mean that the interaction patterns associated to each role are stable enough to be modeled with a single distribution, but also that the data at disposition are not sufficient to train appropriately the mixtures. In fact, the number of parameters increases linearly with K (the number of distributions in the mixtures), but the amount of data available for the training remains constant. Experiments over larger corpora can probably provide more reliable answers about this point.

The 20 percent of mislabeled data time is due to two main sources of error: the first is the delay of the clustering process in correspondence of speaker changes. On average, the speaker changes in the

K	all	AM	SA	GT	IP	AB	MT
1 (B)	81.3	94.9	0.0	95.8	0.0	58.9	76.9
3 (B)	81.2	97.8	4.0	89.9	0.0	58.9	79.1
5 (B)	80.4	97.8	4.5	90.1	0.0	58.9	79.0
1 (M)	73.9	94.9	1.6	85.0	3.3	53.6	13.0
3 (M)	73.7	94.9	3.1	86.8	3.0	53.6	13.0
5 (M)	73.7	94.9	3.1	86.8	3.0	53.6	13.0

Table 2: Role recognition performance for C1. The table reports the role recognition results for the corpus C1. The results are reported as a function of K and show both the overall accuracy and the accuracy for each role. The "B" stands for *Bernoulli*, and the "M" stands for *Multinomial*.

K	all	AM	SA	GT	IP	AB	MT
1 (B)	83.9	62.2	88.3	96.4	0.0	22.0	0.0
3 (B)	83.6	62.2	88.3	96.0	0.0	22.0	0.0
5 (B)	83.6	62.2	88.3	95.9	0.0	22.0	0.0
1 (M)	83.6	70.2	88.3	93.3	0.0	18.2	35.1
3 (M)	81.9	62.5	84.8	93.5	0.0	18.3	35.1
5 (M)	81.9	92.5	84.8	93.5	0.0	18.3	35.1

Table 3: Role recognition performance for C2. The table reports the role recognition results for the corpus C2. The results are reported as a function of K and show both the overall accuracy and the accuracy for each role. The "B" stands for *Bernoulli*, and the "M" stands for *Multinomial*.

output of the clustering process are delayed by around 2 seconds with respect to the actual speaker changes. The average number of changes in C1 is 30 and this results into roughly 60 seconds of mislabeling (around 10 percent of the average C1 recording length). Similar figures can be found for C2 where roughly 10 percent of the time again is mislabeled beacuse of the delays between actual and detected speaker changes. The performance of the system when using the groundtruth speaker segmentations rather than the output of the speaker clustering is 92.8 percent for C1 and 93.9 for C2. This seems to confirm that around 10 percent of the error is actually due to the above phenomenon (the results have been obtained using a single Bernoulli distribution).

The second major source of error is the classification of IP, MT and AB into GT. Such roles have similar interaction patterns, but the a-priori probability of the GT is much higher. In this way the system tends to favor the GT role. Fortunately, the IP, MT and AB do not account for a large fraction of the data time and the impact on the overall performance is small.

6 Conclusions

This paper has presented experiments on the recognition of roles in radio broadcast news. The experiments have been performed over two corpora for a total of roughly 45 hours of material. The results show that around 80 percent of the data time can be labeled correctly in terms of role with a fully automatic process. The errors are due in part to speaker clustering problems, in part to misclassifications performed by the role recognition step.

The main limit of the experiments is that they are performed over broadcast data where the interactions follow some more or less rigorous constraints. This means that the roles are actually defined a-priori and result into relatively stable patterns across different recordings. The situation is likely to be different in data where the interactions are more spontaneous, e.g. meeting recordings, and this can represents a problem for the approach presented here. However, brodcast data are an important case and the results obtained in this work can have a positive impact on any application dealing with digital libraries of TV or radio programs.

The main novelty of the proposed approach is the use of Social Network Analysis [27] for extracting features that account for the social relationships of the speakers. SNA seems to be a suitable tool for role recognition because roles are not characteristics of a single individual, but rather of individuals interacting with the others. To our knowledge, only few works have used SNA before to extract information about people in audio and video data [29][25].

Two main directions have been identified as a future work: the first is the use of clustering techniques as a mean to identify roles in an unsupervised way, the second is the use of role models as a-priori information to improve the performance of technologies like speaker diarization and speaker segmentation.

References

[1] J. Ajmera. *Robust Audio Segmentation*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2004.

[2] J. Ajmera and C. Wooters. A robust speaker clustering algorithm. In *Proceedings of IEEE Workshop on Automatic Speech Recognition Understanding*, 2003.

[3] S. Banerjee and A.I. Rudnicky. Using simple speech based features to detect the state of a meeting and the roles of the meeting participants. In *proceedings of International Conference on Spoken Language Processing*, 2004.

[4] R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker. The rules behind the roles: identifying speaker roles in radio broadcasts. In *Proceedings of American Association of Artificial Intelligence Symposium*, 2000.

[5] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram. Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1091–1104, 2006.

[6] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[7] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, 2000.

[8] Q. Cai and J. Aggarwal. Tracking human motion in structured environments using a distributed-camera system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1241–1247, 1999.

[9] A.P. Fiske and N. Haslam. Social cognition is thinking about relationships. *Current Directions in Psychological Science*, 5(5):143–148, 1996.

[10] E.L. Glaeser and J.A. Scheinkman. Measuring social interactions. In S.N. Durlauf and H.P. Young, editors, *Social Dynamics*, pages 83–132. MIT Press, 2001.

[11] I. Haritaoglu, D. Harwood, and L.S. Davis. W^4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[12] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A guide to theory, algorithm and system development*. Prentice Hall, 2001.

[13] Y.A. Ivanov and A.F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[14] Z. Kunda. *Social Cognition.* MIT Press, 1999.

[15] G.J. MacLachlan and T. Krishnan. *The EM algorithm and its extensions.* Wiley and Sons, 1997.

[16] C.N. Macrae and G.V. Bodenhausen. Social cognition: thinking categorically about others. *Annual Reviews in Psychology*, 51:93–120, 2000.

[17] L. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):305–317, 2005.

[18] N.M. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[19] A. Pentland. Looking at people: sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, 2000.

[20] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural Computation*, 11:229–242, 1999.

[21] R. Rienks and D. Heylen. Dominance detection in meetings using easily obtainable features. In *proceedings of International Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2006.

[22] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2-3):232–248, 2006.

[23] 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.

[24] A. Vinciarelli. Sociometry based multiparty audio recording segmentation. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1801–1804, 2006.

[25] A. Vinciarelli. Speakers role recognition in multiparty audio recordings using social network analysis and duration distribution modeling. *IEEE Transactions on Multimedia*, 9(6), 2007.

[26] J.A. Ward, P. Lukowicz, G. Troster, and T.E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1553–1567, 2006.

[27] S. Wasserman and K. Faust. *Social Network Analysis.* Cambridge University Press, 1994.

[28] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[29] C.Y. Weng, W.T. Chu, and J.L. Wu. Movie analysis based on roles social network. In *proceedings of IEEE International Conference on Multimedia and Expo*, pages 1403–1406, 2007.

[30] M. Zancanaro, B. Lepri, and F. Pianesi. Automatic detection of group functional roles in face to face interactions. In *proceedings of International Conference on Mutlimodal Interfaces*, pages 47–54, 2006.

[31] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1530–1535, 2006.