Detection and Application of Influence Rankings in Small Group Meetings

Rutger Rienks
University of Twente
Enschede, The Netherlands
rienks@ewi.utwente.nl

Dong Zhang and
Daniel Gatica-Perez
IDIAP and EPFL
Martigny, Switzerland
{zhang,gatica}@idiap.ch

Wilfried Post
TNO
Soesterberg, The Netherlands
wilfried.post@tno.nl

ABSTRACT

We address the problem of automatically detecting participant's influence levels in meetings. The impact and social psychological background are discussed. The more influential a participant is, the more he or she influences the outcome of a meeting. Experiments on 40 meetings show that application of statistical (both dynamic and static) models while using simply obtainable features results in a best prediction performance of 70.59% when using a static model, a balanced training set, and three discrete classes: high, normal and low. Application of the detected levels are shown in various ways i.e. in a virtual meeting environment as well as in a meeting browser system.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence-learning

General Terms

Experimentation, Performance

Keywords

Influence detection, Dominance detection, Machine learning, Small group research

1. INTRODUCTION

In any initial gathering of previously unacquainted individuals who interact in the pursuit of the solution to a problem they face together, observable regularities occur. One of these is that a dominance order, or order of influence is established [16]. This order plays several important roles. It is known, for instance, that participants correlate the position in the order of the group members with the degree of influence each individual has over the group's choice of a solution, and with specific attributions of superior ability. Furthermore, intelligence and judgements of high-quality contributions are generally credited to group members who rank high [1, 9]. This paper investigates if it is possible to automatically extract influence rankings that emerge from small-group meetings through the use of a set of low- and mid-level speech-related features and statistical models for recognition and discovery. Based on a large corpus of small-group meetings, our investigation includes (1) an extensive overview on relevant literature, pointing out a number of factors related to human judgement of influence, and (2) a study on the quality of features (acoustic and speech), and models (static and dynamic) for the automatic recognition and discovery tasks. Our work compares and significantly extends the findings of our previous work [24, 28]. The paper also presents the application of the resulting models in a meeting browser as well as in a virtual meeting environment, two applications that have clear value in the context of multi-modal interfaces.

The paper is organized as follows: Section 2.1 starts by discussing related work, both in psychology and computer science, on the role of influential and dominant persons in meetings and the existing computational approaches for automatic analysis, respectively. Section 3 then elaborates on the meeting data set that was collected and annotated as part of our work. Feature definition and extraction, and the procedure for generating class labels for evaluation of our models are discussed in Section 4 and Section 5. The machine learning techniques that we applied are discussed in Section 6. Section 7 shows the results we obtained. Two applications that we designed that apply the developed models are shown in Section 8. We discuss our results and discuss future plans in Section 9, and end with conclusions in Section 10.

2. RELATED WORK

2.1 Influence and dominance in meetings

Social psychology has studied the concepts of influence and dominance arising from group discussions for several years. According to the Status Characteristics and Expectation states theory [5], if members of a group are either known to be differentiated with respect to one or more identifiers of general social status (occupation, age, race, or gender), or observable as such, the group's measurable influence ranking will be correlated with variations in social status. This theory assumes that people, during the initial formation of these rankings, as well as in the course of seeking problem

ICMI'06, November 2–4, 2006, Banff, Alberta,Canada.

solutions, employ a seemingly rational strategy based upon beliefs about how abilities are distributed in society. The solutions offered by those of higher social status, are more likely to be correct. An alternative view is presented by the so-called Two Process Theory [20]. This theory identifies demeanor as the variable of interest and assumes that variations in demeanor are correlated with variations in relative social status and influence. Variations in assertiveness and other components of demeanor explain attainment of positions in the eventual rankings. Other Social Psychologists found that the unequal distribution of amounts of verbal participation, the directionality of initiation of speech exchanges and the rates of addressing the group as a whole also play a part [2, 10].

2.2 Computational approaches

Although the literature on modelling and understanding the concepts of dominance and influence in multi-party interaction is abundant, very few attempts to automatically estimate such quantities in real discussions have been made so far [4, 24, 28]. Regarding influence, existing approaches assume that (1) this high-level concept can potentially be deduced from low and mid-level signal observations [23], and (2) such observations present regularities (patterns) that models for recognition and discovery are able to extract. Basu et al. [4] described an approach for automatic discovery of influence, in a multi-sensor lounge room where people played interactive debating games, using the influence model. This model is a Dynamic Bayesian Network (DBN) which regards group interactions as a group of Markov chains, each of which influences the others' state transitions. Although this model is a tractable option, it has the limitation that it only models influence between pairs of players, and does not explicitly model the group as such.

To address this issue, Zhang et al. recently proposed a two-level influence model [28], a DBN with two levels corresponding to players and team. The player level represents the actions of individual players. The team level represents group-level actions. The team state at the current time-step influences the players' states at the next time step. In turn, the team state at the current time-step is also influenced by all the players' states at the current time-step. The explicit hierarchy in the model allows for the estimation of the influence of each of the players on the team state, and the distribution of participant-to-team influence is automatically learned from data in an unsupervised fashion.

In another work, Rienks et al. [24] recently proposed a supervised learning approach. The method was based on the formulation of the problem as a three-class classification task in which, through manually annotated data, meeting participants are labeled as having high, normal, or low dominance, using a Support Vector Machine (SVM). A number of features related to speaker-turns and their content were extracted for each participant from speaker-turn segmentations, speech transcriptions, and addressing labels, all of which were manually produced.

Our present work extends earlier works in several ways. First, we systematically compare a number of supervised (resp. unsupervised) machine learning methods to recognize (resp. discover) dominance rankings, that include and supersede the methods by [24] and [28]. Second, we investigate a number of common features derived from audio and speech that need to be adapted for models that handle static or dynamic observations. Third, a larger and more challenging meeting corpus is used as the basis for experiments. Finally, we describe two specific applications of the investigated models, as components of a meeting browser and a virtual meeting room.

3. DATA COLLECTION: MEETINGS AND QUESTIONNAIRES

It is clear that if we ever want to deduce the influence ranking automatically, we need to have a collection of small group meetings from which we have these rankings, as well as access to as most of the signals from that meeting as possible in order to be able to assess the expected observable regularities. All meetings used were project scenario meetings from the Augmented Multi-Party Interaction (AMI) project, which were especially designed to achieve as-close-to natural interaction as possible between the meeting participants in a controlled environment.

For the experiments described in this paper we confined ourselves to the meetings recorded at TNO-Soesterberg in the Netherlands. containing 40 meetings of 10 different design teams with an average duration of 30 minutes each. For all the meetings, questionnaires have been filled in by the participants on which a number of questions had to be completed. One of the questions asked participants to rank all of the meetings participants, including themselves, from most to least influential by assigning them unique nominal values ranging from one (most influential) to four (least influential). Participants were not allowed to rank people equivalently. The resulting permutations of the numbers one, two, three and four, were used for quantization into three classes as described in section 5.

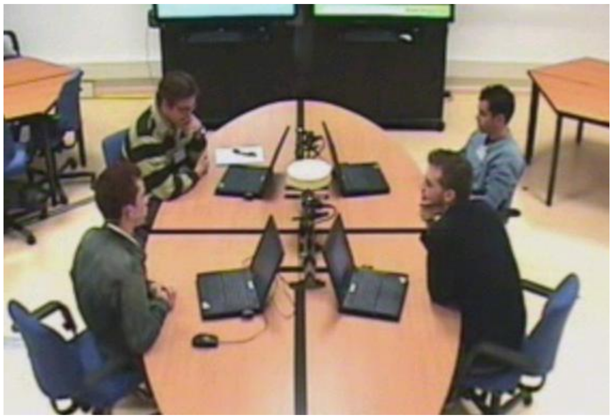


Figure 1: A view from an overview camera on a typical meeting recorded at TNO-Soesterberg.

Figure 1 shows the view from one of the overview cameras for a typical meeting. Scenarios were used for these meetings, which described design meetings where participants were asked to play different roles: a project manager, a marketing expert, a user interface designer, and an industrial designer. During a period of four meetings, a complete design of a remote control had to be realized.

4. EXTRACTION OF FEATURES

The features we use include a selection of the features described by Rienks et al. [24], as well as some newly designed ones. The used features relate to the demeanor of the participants as well as to the status of the participants. They can be grouped into three categories: individual speech behavior, interaction behavior, and semantic-based features. The first group of features contains features dealing with individual speech behavior and comprises the following:

- **The number of turns***. A turn is defined by a complete utterance without silences longer than 1.5 s that contains at least one word.
- **The number of words per turn**.
- **The duration per turn** measured in milliseconds.

The second category of features reveals aspects of the interaction going on in the meeting:

- **The number of floorgrabs***. A floorgrab is defined each time a participant started speaking after a silence greater than 1.5 seconds.
- **The number of successful interruptions***. We defined a successful interruption as was done in [17]. A successful interruption is counted for speaker A, when speaker A starts talking while another speaker B is talking and speaker B finishes his turn before speaker A does. To compensate for backchannel noise, the turn from speaker A had to be at least three words long.
- **The number of times a person is interrupted by someone else***. We count a speaker A, as being interrupted, when another speaker B starts speaking while speaker A has not finished and speaker B finishes his turn at least three words after speaker A.

The third category contains the features related to the meeting semantics, such as the role of the participants, and the topics of the meetings:

- **The predefined role of the participant**. As already described in Section 3, there are four types of roles assigned to the meeting participants: Project Manager, Industrial Designer, User Interface Designer and Marketing Expert.
- **Topic initialization***. For this feature we calculated the number of topics initiated by each of the participants, that were resumed by another participant.

All these features were obtained from the manual transcriptions of the meetings. These transcriptions were all made according to the guidelines defined by Moore et al. [18], and contained time information, which made the extraction of the feature values for each of the meetings a relatively straightforward task. The predefined roles were obtained from the meeting metadata, and the topic steering feature was obtained using probabilistic latent semantic analysis (PLSA) [12]. PLSA is a language model that projects documents in the high-dimensional bag-of-words space into a topic-based space of lower dimension. Each dimension in this new space represents a "topic", and each document is represented as a mixture of topics. In our case, a document corresponds to one speech utterance. Therefore, the topic boundary is equivalent to the utterance boundary. PLSA is thus used as a feature extractor that could potentially capture "topic turns" in meetings. We repeat the PLSA aspect within the same utterance. The topic for the silence segments was set to zero.

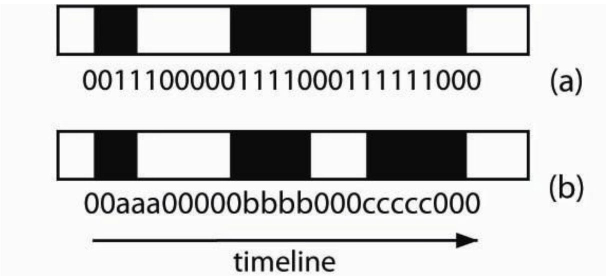


Figure 2: Illustration of the sequential features which serve as input to the dynamic model: (a) a sequence of binary features. For example, one indicates *speaking*, and zero indicates *silent*. (b) A sequence of number of words (or utterance duration) features, where a, b, c indicate the number of words (or the speaking duration) in one utterance separately. We repeat the same value within the same utterance. The value for the silence segments was set to zero.

The described features need to be adapted to be used as observations by the static and dynamic models described in Section 6. For the static model, most of the features have been normalized (indicated by the '*') in order to make them inter-meeting and inter-person comparable. This was done in line with the work by Rienks et al. [24]. The fraction or share F'_{Pn} of a feature value for a given person n was calculated given all the values for that feature in a meeting, by defining the normalized feature value $(F'_{Pn}) = \frac{F_{Pn}}{\sum_{i=1}^{4} F_{Pi}}$.

For the dynamic model, the features are manipulated as follows. For the first feature, *speaking turn*, the feature sequence consists of binary values, one if the person speaks and zero otherwise. We apply a similar treatment to the *floorgrab, interruption*, and *interrupted* features. This is illustrated in Figure 2 (a). For the *number of words* feature, we repeat the value within the same utterance. The number of words for the silence segments was set to zero (Figure 2(b)). Note that this feature representation is effectively using non-causal information. The *turn duration* feature is treated similarly. Finally, the *role* feature was not used for the dynamic model, as its value is constant for each participant over the entire meeting.

5. EXTRACTION OF CLASS LABELS

In our work, to be able to evaluate influence rankings, we turn the problem into a multi-class classification problem, where we need to assign class labels to each of the meeting participants. To obtain the class label ground-truth, both for evaluation and for training of supervised methods, we used the rankings of the individual participants provided by the questionnaires. This is in contrast to our earlier work, where we used external observers either to rank all the meeting participants from most to least dominant [24], or to assign continuous influence values to participants [28]. As we had four participants, this resulted in four rankings for the same meeting. On these rankings, a binning algorithm was applied that resulted in three discrete class labels: 'High', 'Normal', and 'Low'. This was done by summing up and then normalizing the rank scores for each of the participants. The total score for each participant was then binned depending on the score of the participant in relation to the score of the others. (e.g. each participant was assigned four times either a 1,2,3 or 4 as rank number, resulting in a maximum score of 16 points out of a total of 40). The normalized value was subsequently binned using the labels 'High' ($F^{'}_{Pn} > 30\%$), 'Normal' ($20\% < F^{'}_{Pn} < 30\%$), and 'Low' ($F^{'}_{Pn} < 20\%$). As a consequence, apart from the fact that features are now comparable between meetings, the feature values that originally had 'approximately' the same value now also end up in the same bin. The resulting data set has a total of 160 labels (40 meetings times four participants) resulting in 34 observations for 'Low', 91 for 'Normal', and 35 for 'High'.

6. AUTOMATIC INFLUENCE DETECTION

6.1 The dynamic model

We investigate the unsupervised model recently proposed by Zhang et al. [28]. The team-player influence model is a dynamic Bayesian network (DBN) with a two-level structure: the *player* level and the *team* level (Figure 3). The player level represents the actions of individual players, evolving based on their own Markovian dynamics (Figure 3 (a)). The team level represents group-level actions (the action belongs to the team as a whole, not to a particular player). In Figure 3 (b), the arrows up (from players to team) represent the influence of the individual actions on the group actions, and the arrows down (from team to players) represent the influence of the group actions on the individual actions. The model considers probability distributions over sets of random variables $\{S, O\}$, where S is a set of state variables that represents individual and team actions, and O is set of observation variables. The model is a directed graphical model represented as $G = (V, E)$, where $V = \{S, O\}$ is a set of variables, and E is a set of oriented edges. A directed graphical model is a family of probability distribution that factorizes according to an underlying graph [14]. Following [28], let O^i and S^i denote the observation and state of the i^{th} player respectively, and S^G denotes the team state. For N players, and observation sequences of identical length T, according to Figure 3, the joint distribution is given by

$$P(S,O) = \prod_{i=1}^{N} P(S_1^i) \cdot \prod_{t=1}^{T}\prod_{i=1}^{N} P(O_t^i|S_t^i) \cdot \prod_{t=1}^{T} P(S_t^G|S_t^1 \cdots S_t^N) \cdot \prod_{t=2}^{T}\prod_{i=1}^{N} P(S_t^i|S_{t-1}^i, S_{t-1}^G). \quad (1)$$

Regarding the player level, the actions of each individual are modelled with a first-order Markov model (Figure 3 (a)) with one observation variable O^i and one state variable S^i. Furthermore, to capture the dynamics of all the players interacting as a team, a hidden variable S^G (team state) is added to model the group-level actions. Unlike the individual player states that have their own Markovian dynamics, the team state is not directly influenced by its previous state. S^G could be seen as the aggregate behaviors of the individuals, yet provides a useful level of description beyond individual actions. There are two kinds of relationships between the team and players:

1. The team state at time t influences the players' states at the next time (down arrow in Figure 3 (b)). In other words, the state of the i^{th} player at time $t+1$ depends on its previous state as well as on the team state, i.e., $P(S_{t+1}^i|S_t^i, S_t^G)$.

2. The team state at time t is influenced by all the players' states at the current time (up arrow in Figure 3 (b)), resulting in a conditional state transition distribution $P(S_t^G|S_t^1 \cdots S_t^N)$.

An extra hidden variable Q is added in the model to switch parents for S^G. The idea of switching parent (also called Bayesian multi-nets in [6]) is as follows: a variable ,S^G in this case, has a set of parents $\{Q, S^1 \cdots S^N\}$ (Figure 3(c)). Q is the switching parent that determines which of the other parents to use, conditioned on the current value of the switching parent. $\{S^1 \cdots S^N\}$ are the conditional parents. In Figure 3(c), Q switches the parents of S^G which corresponds to

$$P(S_t^G|S_t^1 \cdots S_t^N) = \sum_{i=1}^{N} P(S_t^G, Q = i|S_t^1 \cdots S_t^N) \quad (2)$$

$$= \sum_{i=1}^{N} P(Q = i|S_t^1 \cdots S_t^N) P(S_t^G|S_t^i \cdots S_t^N, Q = i) \quad (3)$$

$$= \sum_{i=1}^{N} P(Q = i) P(S_t^G|S_t^i) = \sum_{i=1}^{N} \alpha_i P(S_t^G|S_t^i). \quad (4)$$

From Equation 3 to Equation 4, two assumptions are made: (i) Q is conditionally independent of $\{S^1 \cdots S^N\}$; and (ii) when $Q = i$, S_t^G only depends on S_t^i. The distribution over the switching-parent variable $P(Q)$ describes how much influence or contribution the states of the player variables have on the state of the team variable. The term $\alpha_i = P(Q = i)$ is referred to as the influence value of the i^{th} player. Obviously, $\sum_{i=1}^{N} \alpha_i = 1$ (the sum of contributions of all players equals 1). In multi-party meetings, α_i represents the influence of each participant.

6.2 The static models

The static models are applied for post-meeting processing, in contrast to the dynamic model. This implies that they use features whose values are summed and normalized when the whole meeting is over, and that the models are not able to 'track' the influence online. All static models are supervised models that build a model from training data (feature values and class labels). The task of the supervised learner is to predict the class value for any valid combination of inputs after having seen a number of training examples. In total, four types of static classifiers were used:

- **Support Vector Machines** (SVM) are used for classification and regression. Their common factor is the use of a technique known as the 'kernel trick' to apply linear classification techniques to non-linear classification problems. Multi-class problems are solved using pairwise classification [7].

- **Multi Layered Perceptrons** (MLP), or feedforward neural networks, are a special kind of neural network, consisting of multiple layers of perceptrons. A perceptron is a simple binary classifier which maps its inputs

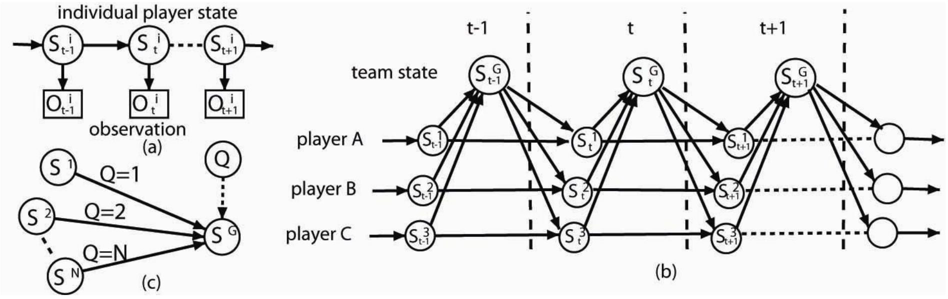


Figure 3: The team-player influence model (reproduced from [26]).(a) Markov Model for individual player. (b) Two-level influence model (for simplicity, we omit the observation variables of individual Markov chains, and the switching parent variable Q). (c) Switching parents. Q is called a switching parent of S^G, and $\{S^1 \cdots S^N\}$ are conditional parents of S^G. When $Q = i$, S^i is the only parent of S^G.

x_i or x (a real-valued vector) to an output value $f(x)$ [26].

- **The C4.5 Decision Tree Learner** (C4.5). A decision tree consists of non-terminal nodes and terminal nodes. The non-terminal nodes represent tests on one or more attributes of the data. The terminal nodes represent the outcome: the decision of the classifier. At each non-terminal node the classifier will test an attribute of the input instance and push it on a branch depending on the outcome of the test. C4.5 is an algorithm that creates decision trees [21].
- **NaiveBayes Classifier** (NB). This is a probabilistic classifier that uses Bayesian Formulations using prior probabilities to assign class labels, assuming independent attributes [13].

7. RESULTS

This section contains the results for both the static and the dynamic model. As we modeled the problem as a classification problem we will mention the percentage of the correctly classified instances and a confusion matrix for both models.

7.1 Results for the dynamic model

We train the dynamic model using all (except the 'role' feature) features individually. All features were extracted at 5 frames per second. For example, for a 5-minute meeting, the total number of feature frames is 1500. Since the learned influence value α_i using the dynamic model is a real value, ranging from 0 to 1, to compare it to the manually labeled influence rank (a discrete value out of the three labels {1, 2, 3}), we transform α_i into a discrete label using two thresholds: th_1, th_2 based on the following equation. The values of th_1 and th_2 range from 0.1 to 0.5 under the constraint that $th_1 < th_2$,

$$label_i = \begin{cases} 3 & : \quad \alpha_i < th_1 \\ 2 & : \quad th_2 > \alpha_i > th_1 \\ 1 & : \quad \alpha_i > th_2 \end{cases}$$

For all experiments, we used ten-fold cross-validation. We first divided the data into ten subsets each of which contained four meetings. We then tested the models ten times with different parameter configurations (*i.e.* th_1, th_2), each time leaving out one of the subsets to compute performance.

Method		Accuracy (%)
Individual Features	Number of turns	56.25 (4.23)
	N.o.w. per turn	57.50 (5.27)
	Turn duration	61.25 (4.84)
	Floorgrabs	48.75 (4.77)
	Succ. interr.	48.13 (4.16)
	Is Interrupted	45.00 (3.44)
	Topic Init.	53.50 (4.54)
Fusion	average	54.38 (3.94)

Table 1: Results on the dynamic model using different features (standard deviation in brackets).

High	Normal	Low	← Classified as
23	10	1	High
26	50	15	Normal
9	9	17	Low

Table 2: One example confusion matrix of the recognized influence labels.

We reported the mean accuracy and the standard deviation. The results are summarized in Table 1. For feature fusion, we use a naive averaging method: $\alpha = \frac{1}{K}\sum_{i=1}^{K}\alpha_i$, where K is the the number of features. We also report the fusion result in Table 1.

7.2 Results for the static models

We created two versions of feature values, one version where all the feature values were normalized, and one where all the normalized feature values were normalized and binned dependent on their share in relation to the other participants. This binning was performed using the same thresholds as described in section 5. Ten fold cross-validation was in every case applied while determining the results. The performance on the original (unbalanced) data are shown in Table 3.

It appears that the static models are sensitive to unbalanced training. Given a baseline of 57% (91 out of the 160 observations were labelled 'Normal'), the results are far from good.

Although the results on the original case are more representative for the data as it naturally is, a set of 100 different

	Unbalanced	
Classifier	Normalized (%)	Normalized/Binned (%)
C45	46.25	57.50
Naive Bayes	66.25	61.25
MLP	55.00	53.13
SVM	61.25	60.00

Table 3: Results of the static model on the original unbalanced data sets.

balanced versions was also created, each containing 34 observations for all of the class labels (102 in total). More than one balanced version was created to preserve the distribution of the feature values. Ten-fold cross validation was applied on all the 100 data sets. The performance on the balanced provides an indication of the legitimacy of our approach as will be discussed. Table 4 shows the averaged performances including standard deviations for all balanced data sets.

	Balanced	
Classifier	Normalized (%)	Normalized/Binned (%)
C45	52.18 (5.14)	54.93 (5.13)
Naive Bayes	59.65 (2.98)	60.16 (3.40)
MLP	54.21 (4.30)	50.82 (4.36)
SVM	58.78 (3.64)	58.45 (3.85)

Table 4: Results of the static models on balanced data sets (standard deviation in brackets).

As we now have a baseline of 33% due to our balanced training sets, it appears that the results as shown in Table 4 are much better than those in Table 3. To make the results more comparable to the results produced by the dynamic model we took the two balanced (normalized and binned) sets that produced the best and the worst result and computed the performance for each of the classifiers when only using the individual features. The results are summarized in Table 5.

	Best Perf. (%)	Worst Perf. (%)
Number of turns	57.84 (NB)	46.08 (SVM)
N.o.w. per turn	50.98 (MLP)	36.27 (NB)
Turn duration	56.86 (NB)	37.25 (SVM)
Floorgrabs	31.37 (SVM)	20.59 (NB)
Succ. interr.	50.00 (C4.5)	46.16 (SVM)
Is Interrupted	42.16 (C4.5)	30.39 (NB)
Role	46.08 (MLP)	45.01 (C4.5)
Topic Change	57.84 (NB)	51.96 (MLP)
All features	**70.59**(NB)	**42.16** (C45)

Table 5: Performance of individual features for the balanced sets (normalized and binned) with the best (70.59%) and worst performance (42.16%) on all features (model in brackets).

From Table 5 it follows that, in particular, the feature 'Floorgrabs' by itself is unable to outperform the naive baseline of 33%. The 'Topic Change'-feature on the other hand seems to be quite robust and useful. Post hoc feature subset evaluation revealed a best subset containing the features 'Number of turns', 'Turn Duration', 'Role' and 'Topic Change'. The method we applied searches for features which highly correlate to the class-labels and have a low inter feature correlation [11]. This way the features that complement each other are preserved whereas features with a discriminative power similar to other features are being removed. Using only the resulting subset, a best performance of 69.61% was achieved using NB (not shown in Table 5). We conclude the results on the static model by presenting the confusion matrix for our best result (70.59%), which used all features, in Table 6.

High	Normal	Low	← Classified as
26	8	0	High
3	23	8	Normal
4	7	23	Low

Table 6: Confusion Matrix on our best run (=70.59%) using Naive Bayes.

From the confusion matrix it follows that 'Low' influence persons are sometimes even labeled as 'High', whereas 'High' influence persons are never labeled as 'Low' influence.

7.3 Interpretation of the results

After looking at the outcomes of the dynamic model (Table 1) and comparing it to those for the static model (Table 5), the following issues are worth commenting:

1. It appears that the best combined feature performance of the static model (70.59%), outperforms the best performance of the dynamic model (54.38%). One reason might be that the dynamic model is a generative model, whereas the static is a discriminative model trained in a supervised manner.

2. The best individual feature for the dynamic model turns out to be the *turn duration*, while the best individual features for the static model seem to be the *number of turns* and the *topic initialization*. This indicates that the best feature using dynamic model is not necessarily the best feature using static models.

3. For each individual feature, it is hard to say which model is better. For example, the performance of the *number of turns* feature whilst using the dynamic model is better than using SVM, but worse than using NB.

4. The best subset, containing just four out of the eight examined features, resulted for the static model in a nearly equal performance to that for the complete feature set. With respect to the amount of effort one wants to invest on feature extraction, this is certainly something to take into account.

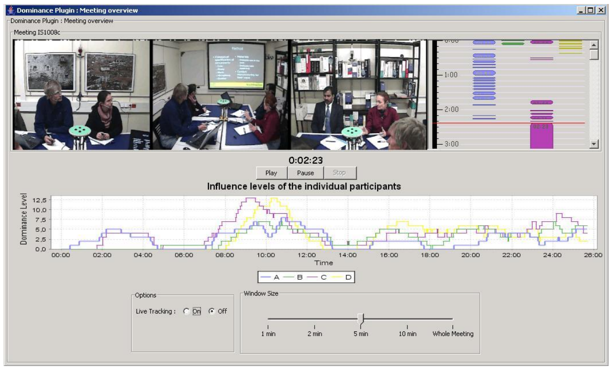
5. With respect to the significance of our results, we would like to mention that although our sample size is considerably larger than [24], it is still relatively small when compared to a typical classification problem.

On a general level differences between the dynamic and the static model lie in the fact that the static model is comparatively quite fast, is able to combine several features and requires feature values calculated over the whole meeting. The dynamic model, on the other hand, can deal with dynamic feature value updates, but cannot output influence of each meeting participant at any moment of the meeting, while the static model could output such values with some heuristics [24].

8. APPLICATIONS

We now present two applications of the developed models. A first implementation has been created for the JFerret meeting browser, developed by Wellner et al. [27], which enables people to access meeting information. Here the influence levels are shown over the meeting depicted by a graph (see Figure 4). The application allows for 'live' tracking of the influence levels

Live tracking of the influence levels was made possible in a way similar to [24]. Occurrences for each of the features, such as interruptions, are observed and further processed by the model responsible for producing the final value. The plug-in is extended with a feature that allows for manually setting a time period wherein the observations are used for computation of the influence value. This provides the opportunity to view the output either as a set of cumulatively increasing lines (whole meeting period), or as a set of lines revealing more about the change of the output over time, such as a five-minute time period as shown in Figure 4. One could envision that, one day, if the browser is used by managers interested in the performance of their employees, the influence levels plug-in could provide valuable information. If just and valid arguments were put forward on one hand and the person was not influential in the given setting on the other hand, this might also be a point to address. Also as a preparation task, looking over the behavior of how influential participants were in a previous meeting might prove useful when selecting someone to attend an upcoming meeting with these same participants.


Dominance Plugin : Meeting overview

Meeting IS1008c

0:02:23

Live Tracking : ○ On ● Off

Window Size: 5 min

Figure 4: A graphical visualization of the calculated influence levels.

Another implementation has been realized in the Virtual Meeting Room (VMR), a copy of the smart meeting room at IDIAP, developed at Twente [22]. The VMR was developed for signal replay, as a remote conferencing application, and to serve as a test environment for meeting assistants. In this meeting room, the relative influence levels can be depicted by the size of the black balls shown in front of the participants (see Figure 5). This in addition to, for example, the domes surrounding the participants' heads that provide information about their gaze behavior. [19].

9. DISCUSSION AND FUTURE WORK

This section contains some thoughts and future work on the approach that we took.

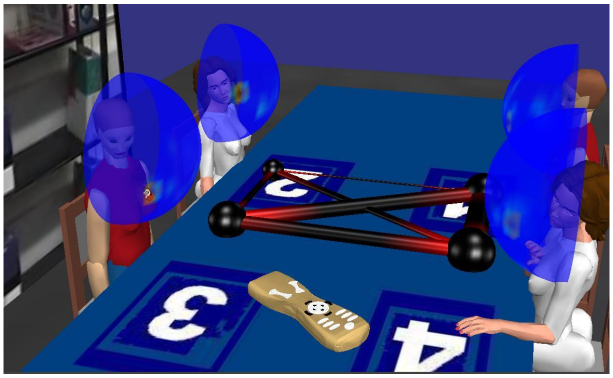


Figure 5: A visualization of the calculated influence levels in a Virtual Meeting Room

Taking it all together, it appeared that we could not reproduce the results mentioned in [24]. Our best result (70.59%) is 4.5% lower. We believe this to be mainly due to two reasons. In the first place, we used many more and longer lasting meetings, resulting in a significantly larger number of data samples (102 vs. 32 for the static model). Second, the acquisition of class labels differed in a way that the values in our case were provided by the participants themselves and not by external observers.

Although there is no evidence that the 'subjective' route we took here will differ from an 'objective' one, in this case there were two reasons to take a more 'subjective' route. In the first place it is a costly enterprise to have all meetings being watched (preferably more than once). Second, this research is grounded with respect to real-time meeting supporting applications such as the live assistants further explained below. On the other hand, it must be said that we did not did not use a complete 'subjective' view as we merged the individual contributions into one single class label.

Another point we would like to stress is that we extracted the class labels in a way comparable to [24]. A drawback of this approach is that this will always result in an unequal distribution of the labels. For the dynamic model we found, for example, that using different thresholds yields better results. Hence, in future experiments one could decide to modify these in order to end up with an equally balanced corpus. We did not do this, as we wanted our results to be comparable with our earlier work.

The features we used were distilled from our earlier work combined with the social psychological literature we studied, and intuitively all seemed appropriate. Closer inspection revealed however that especially the 'floorgrab' feature by itself performed very badly and seemed hardly to have any discriminative power. A broader spectrum of features, possibly from other modalities such as vision, might eventually lead to better results. Head orientation information from which addressee information can be distilled [15] could prove beneficial in this case [24]. Another aspect is the performance measure we used. We looked at exact prediction of the correct class labels, whereas from an end user point of view, the inter personal findings (Was A more influential than B?) might be of greater importance. Currently, different evaluation methods are therefore under construction.

Perhaps the most interesting question, namely how real-time accessibility to influence values might impact a meeting or the individual participants, is still work that lies ahead, and no experiments have been conducted yet to see how and if the presentation of this information might actually have an impact on the meeting. DiMicco reports on such experiments using a system called Second Messenger [8], which shows real-time text summaries of participants' contributions. In that work, it turned out that after increasing the visibility of the less frequently speaking group members, these started to speak more frequently than before, whereas the more dominant people started to speak 15% less. We are thinking of conducting similar experiments in the future.

We foresee and plan to integrate the developed models on influence detection into meeting assistants that support the meeting process or specific individuals in real time. Meeting assistants can be thought of as (embodied) pervasive software systems that operate alone or in groups, interact with the users and with other participants, and learn user preferences (see e.g. Project Neem [3]). It is expected that meeting assistants in the future will be used as tools for remote meeting participation. In previous work, e.g. [25], we reported initial findings on experiments with meeting assistants.

10. CONCLUSIONS

This paper stressed the role and the impact of influence levels on meetings and its participants. We have shown that automatic detection of influence rankings is a hard and rather complex task. With our best performance touching 70% using static machine learning models, it is clear that we just made some explorative steps on the long path that lies ahead in order to fully understand humans in a way that allows us to automatically extract their relative influence levels within small groups.

11. ACKNOWLEDGEMENTS

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication AMI-192), and by the Swiss NCCR on Interactive Multimodal Information Management (IM2).

12. REFERENCES

[1] R. Bales. *Iteraction Process Analysis.* Addison Wesley, 1950.

[2] R. Bales, F. Strodbeck, T. Mills, and M. Roseborough. Channels of communication in small groups. *American Sociological Review*, 16:461–468, 1951.

[3] P. Barthelmess and C. A. Ellis. The neem platform: An evolvable framework for perceptual collaborative applications. *Journal of Intelligent Information Systems*, 25(2):207–240, 2005.

[4] S. Basu, T. Choudhury, B. Clarkson, and A. Pentland. Learning human interactions with the influence model. Technical Report 539, MIT Media Laboratory, June 2001.

[5] J. Berger, S. Rosenholtz, and M. Zelditch Jr. Status organizing processes. *Annual Review of Sociology*, 6:479–508, 1980.

[6] J. Bilmes. Dynamic bayesian multinets. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, Stanford University, CA, U.S.A., 2000.

[7] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on COLT*, pages 144–152. ACM Press, 1992.

[8] J. DiMicco. Designing interfaces that influence group processes. In *Doctoral Consortium Proceedings of the Conference on Human Factors in Computer Systems (CHI 2004)*, April 2004.

[9] M. Fisek and R. Ofsche. The process of status evolution. *Sociometry*, 33:327–346, 1980.

[10] G. Goetsch and D. McFarland. Models of the distribution of acts in small discussion groups. *Social Psychology Quarterly*, 43(2):173–183, 1980.

[11] M. Hall. *Correlation-based Feature Subset Selection for Machine Learning.* PhD thesis, Department of Computer Science, University of Waikato, Waikato, N.Z., 1999.

[12] T. Hofman. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001.

[13] G. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufman, 1995.

[14] M. Jordan. *Learning in graphical models.* MIT Press, 1999.

[15] N. Jovanovic, R. Op den Akker, and A. Nijholt. Addressee identification in face-to-face meetings. In *11th Conference of the European Chapter of the ACL (EACL)*, Trento, Italy, 2006.

[16] M. Lee and R. Ofsche. The impact of behavioral style and status characteristics on social influence: A test of two competing theories. *Social Psychology Quarterly*, 44(2):73–82, 1981.

[17] A. Leffler, D. Gillespie, and C. Conaty. The effects of status differentiation on nonverbal behaviour. *Social Psychology Quarterly*, 45(3):151–161., 1982.

[18] J. Moore, M. Kronenthal, and S. Ashby. Guidelines for AMI speech transcriptions. Technical Report 1.2, IDIAP, Univ. of Edinburgh, February 2005.

[19] A. Nijholt, R. Rienks, J. Zwiers, and D. Reidsma. Online and off-line visualization of meeting information and meeting support. *The Visual Computer.* to appear.

[20] R. Ofsche and M. Lee. Status, deference, influence and convenient rationalization: An application of two-process theory. Working Papers in Two-Process Theory 3, Department of Sociology, University of California, 191.

[21] R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

[22] D. Reidsma, R. op den Akker, R. Rienks, R. Poppe, A. Nijholt, D. Heylen, and J. Zwiers. Virtual meeting rooms: From observation to simulation. In *Proceedings of the 4th workshop on Social Intelligence Design*, 2005.

[23] D. Reidsma, R. Rienks, and N. Jovanovic. Meeting modelling in the context of multimodal research. In *Proc. of the Workshop on Machine Learning and Multimodal Interaction*, 2004.

[24] R. Rienks and D. Heylen. Automatic dominance detection in meetings using easily detectable features. In *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Endinburgh, Scotland, 2005. Springer Verlag.

[25] R. Rienks, A. Nijholt, and P. Barthelmess. Pro-active meeting assistants : Attention please! In *Proceedings of the 5th workshop on Social Intelligence Design*, Osaka, Japan, 2006.

[26] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[27] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with ferret. In *In Proceedings of MLMI'04*. Springer-Verlag, 2004.

[28] D. Zhang, D. Gatica-Perez, S. Bengio, and D. Roy. Learning influence among interacting markov chains. *Advances in Neural Information Processing Systems (NIPS)*, 18, 2005.