Semi-Supervised Manifold Learning Approaches for Spoken Term Verification

Atta Norouzian[1], Richard Rose[1,2], Aren Jansen[2]

[1]Department of Electrical Engineering, McGill University, Montreal, Canada
[2]Human Language Technology Center of Excellence, Baltimore, USA
atta.norouzian@mail.mcgill.ca, rose@ece.mcgill.ca, aren@jhu.edu

Abstract

In this paper, the application of semi-supervised manifold learning techniques to the task of verifying hypothesized occurrences of spoken terms is investigated. These techniques are applied in a two stage spoken term detection framework where ASR lattices are first generated using a large vocabulary ASR system and hypothesized occurrences of spoken query terms in the lattices are verified in a second stage. The verification process is performed using a fixed dimensional feature representation derived from each hypothesized term occurrence. Two semi-supervised approaches namely, manifold regularized least squares (RLS) classification and spectral clustering, are investigated for distinguishing correct hypotheses from false alarms. It is shown that, exploiting unlabeled data in addition to labeled data using semi-supervised approaches, significantly improves the verification performance compared to the case where only the labeled data is used. This improvement in performance increases as the ratio of unlabeled to labeled data augments. It is also shown that, when training data is very limited, a comparable verification performance can be gained by exploiting only the acoustic similarity between the test samples using the spectral clustering approach.

Index Terms: spoken term detection, semi-supervised learning, manifold learning, regularized least squares classifier

1. Introduction

This paper is concerned with the problem of detecting occurrences of spoken terms in continuous running speech. Spoken term detection (STD) is applied to scenarios where users enter query terms into a search engine to identify relevant speech segments in a collection of audio recordings. The techniques investigated in this work are applied in the context of a large vocabulary continuous speech recognition (LVCSR) system which generates lattices of solutions for each speech segment in the audio collection. Term verification techniques are presented which verify the hypothesized occurrences of query terms in the lattices. There has been a great deal of effort devoted to this problem [1, 2, 3, 4] and there are a number of issues that must be addressed when applying these techniques in practice. The most important issue is the availability of enough training data for training term specific models for verification. While it is assumed in this work that search terms are contained in the ASR vocabulary, the interest here is in developing term verification procedures which can operate with only a very small number of labeled examples of each term.

The goal in semi-supervised training of parametric classifiers is to reduce the requirement for expensive labeled training examples by exploiting large unlabeled corpora for training. In ASR applications, obtaining labeled data implies a time-consuming process of manual transcription of many hours of speech. Specifically, in the term verification problem, it implies that users label large numbers of hypothesized term occurrences from the ASR lattices as correct detections or false alarms. This is generally impractical in STD applications since query terms are provided by users in real time and are usually not known to the system in advance. The assumption in this work is that users can manually label a small number of these hypothesized occurrences perhaps through a process of reviewing an initial set of candidate speech segments. However, there may be an extremely large number of unlabeled putative hits that can be incorporated in a semi-supervised training scheme.

Semi-supervised learning for term verification is investigated for use in the second stage of a two stage STD framework. In the first stage, which is performed off-line, speech segments of length 20 - 60 seconds are fed to an LVCSR system and a word lattice is generated for each segment. In the second stage, upon receiving a query term from a user, hypothesized occurrences of the query term are identified in the lattices. This process may be made efficient by using any of a number of lattice indexing schemes [5, 6, 7, 8]. These hypotheses contain true occurrences of the query terms as well as false alarms and it is the job of the second stage term verification to detect which of these hypothesized occurrences correspond to actual term occurrences. The advantage of this two stage process is that second stage term verification can be performed using completely different acoustic features and modeling formalisms from those used in first pass LVCSR.

Two low resource term verification scenarios are considered in this paper. In both cases, a fixed-dimensional feature representation based on phonetic event patterns derived from the decoded term occurrences is used. This feature representation is described in [9]. The first verification scenario assumes the existence of a small number of labeled term hypotheses and a large number of unlabeled hypotheses for training a whole word-based manifold-regularized least squares (RLS) classifier. A semi-supervised learning approach for manifold-based RLS training is described in Section 3. A discussion of the use of supervised RLS classifiers for term verification has also been presented in [1]. The impact of varying amounts of labeled and unlabeled training data on term verification performance is evaluated and compared with fully supervised training in Section 6.

The second scenario assumes that no labeled training data is available, and only the candidate intervals themselves at test time are available. In this case, spectral clustering is applied to graphs whose connection weights characterize the acoustic similarity of hypothesized query term occurrences. Described in Section 4, this clustering isolates graph nodes corresponding to true occurrences using a single labeled example obtained from the test set. The term verification equal error rate obtained for this approach is presented in Section 6.

2. Verification Task

The verification process is performed on the hypothesized occurrences of the query term Q in lattices generated by an LVCSR system. Having extracted the start and end time of each hypothesis, $I_i(Q)$, from the lattices, a d-dimensional feature vector, $\mathbf{x}_i \in \mathbb{R}^d$, is generated from the corresponding interval in the audio recordings. The feature representation used in this paper is based on temporal point process patterns of phone classes [9]. A temporal point process as described in [10] is a stochastic or random process composed of a time-series of binary events that occur in time. In speech, the occurrence of a phone can be regarded as a binary event and consequently a point process can be generated for a phone over time.

Having a set of l phones $\mathcal{P} = p_1, p_2, \ldots, p_l$, the point process pattern of each phone class $p_i \in \mathcal{P}$ over T frames of speech can be represented by a sparse vector of length T with elements equal to one when p_i occurs and zero otherwise. Here, a neural network based phone classifier is deployed for finding the occurrence times of each phone in speech intervals corresponding to the hypothesized query term occurrences. The point process representation of each phone is then normalized with respect to its length and quantized so that a fixed dimensional representation is obtained for intervals of different lengths. For the given hypothesis $I_i(Q)$, the point process representations of all the phones in the set $\mathcal{P}$ are concatenated to form a d-dimensional feature vector $\mathbf{x}_i$. This process is explained in detail in [1]. The set of N feature vectors, $\{\mathbf{x}_i\}_{i=1}^N$, extracted from all hypotheses of the query term $\{I_i(Q)\}_{i=1}^N$ are then classified into two classes, $y_i \in \{1, -1\}$, corresponding to true occurrence of the query term and false alarm, respectively.

In [1] the use of fully-supervised RLS based classification was exploited for this verification task. In this paper, two low resource verification scenarios are considered and for each scenario a semi-supervised classification approach is presented. In the first scenario a small number of labeled candidate intervals and a large number of unlabeled candidate intervals are available for training a classifier for each query term. The semi-supervised classification approach deployed for this scenario is based on the manifold based RLS training. In the second scenario there are no labeled or unlabeled candidate intervals for training. For this case a graph based semi-supervised classification approach based on spectral clustering is used. With this approach the classification of candidate intervals of the test set is performed by using the label of only a single arbitrary test candidate interval. In the following sections these two semi-supervised classification approaches are explained.

3. Semi-Supervised RLS

In the RLS framework the objective is to learn a function $f : \mathbb{R}^d \to \mathbb{R}$ that maps the data points into their true classes after applying an appropriate threshold. In supervised training of RLS classifiers, where the actual class label of the training samples is known, this function is learned from a set of l labeled training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ as described in [1]. However, when the number of labeled samples is small the learned function does not generalize well to unseen data. In semi-supervised training of RLS classifiers, the objective is to use not only the l labeled samples but also a set of u unlabeled training samples, $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, to characterize the class-independent distribution of the data for data-driven regularization. In [11] it is shown that if the distribution of the input data is restricted to a d'-dimension ($d' < d$) manifold, we can apply graph-based manifold reg-

ularization techniques toward this end. In the manifold regularization framework, the function f is learned by solving the following optimization problem

$$f^\star = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \frac{1}{l} \sum_{i=1}^l [f(\mathbf{x}_i) - y_i]^2 + \gamma_A \|f\|_K^2$$
$$+ \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} [f(\mathbf{x}_i) - f(\mathbf{x}_j)]^2 W_{i,j}. \quad (1)$$

On the right hand side of Equation 1 the first term is a loss function and the second term is an ambient (L_2) regularization term with regularization factor γ_A whose purpose is to avoid over-fitting. This optimization problem without the third term is exactly the same as the one used in the typical supervised training of RLS.

The third term on the right hand side of Equation 1 is a manifold regularization term with the regularization factor γ_I. This term, is a measure of functional smoothness when restricted to the manifold, allowing the unlabeled data distribution to influence the classifier learned. The values $W_{i,j}$ represent a measure of similarity between each pair of data points $(\mathbf{x}_i, \mathbf{x}_j)$. In the verification problem discussed here each data point $\mathbf{x}_i$ corresponds to the feature vector extracted from candidate interval I_i. The similarity between two feature vectors is computes using a kernel function of the form $W_{i,j} = \exp(-dist(\mathbf{x}_i, \mathbf{x}_j)/2\sigma_1^2)$, where σ_1 is the kernel width parameter. The distance $dist(\mathbf{x}_i, \mathbf{x}_j)$ between two feature vectors is computed using the van Rossum method as described in [1]. The manifold regularization term of Equation 1 can also be written in a matrix form as $\frac{\gamma_I}{(u+l)^2} \mathbf{f}^T \mathbf{L} \mathbf{f}$, where $\mathbf{L}$ is the graph Laplacian given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The matrix $\mathbf{D}$ is a diagonal matrix whose diagonal elements are given by $D_{i,i} = \sum_{j=1}^{l+u} W_{i,j}$.

Based on the representer theorem, for a symmetric, positive semi-definite kernel K the solution to Equation 1 can be written as

$$f^\star(\mathbf{x}) = \sum_{i=1}^{l+u} \alpha_i K(\mathbf{x}_i, \mathbf{x}). \quad (2)$$

The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, has the same form as the function used in computing $W_{i,j}$ but with a different kernel width parameter $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-dist(\mathbf{x}_i, \mathbf{x}_j)/2\sigma_2^2)$. The classifier parameters $\alpha_i \in \mathbb{R}$ can be obtained from the closed-form matrix solution given by

$$\boldsymbol{\alpha} = (\mathbf{J}\mathbf{K} + \gamma_A l\mathbf{I} + \frac{\gamma_I l}{(u+l)^2} \mathbf{L}\mathbf{K})^{-1}\mathbf{Y}. \quad (3)$$

In Equation 3, I is an identity matrix, and J is an $(l+u)$ by $(l+u)$ diagonal matrix with the first l diagonal entries equal to one and the remaining u entries equal to zero. The matrix $\mathbf{K}$ is the Gram matrix over labeled and unlabeled samples with elements $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ and Y is an $(l+u)$ vector containing the class labels for the l labeled samples and zeros for the unlabeled samples. After computing the parameters of the classifier, $\{\alpha_i\}_{i=1}^{l+u}$, from Equation 3, each test sample is classified using Equation 2.

4. Graph Spectral Clustering

The set of feature vectors, $\{\mathbf{x}_i\}_{i=1}^N$, generated for N candidate intervals $\{I_i(Q)\}_{i=1}^N$ for the query term Q can be represented with an undirected graph $G = (V, E, C)$. In this graph each

vertex $v_i \in V$ represents a feature vector $\mathbf{x}_i$. There is an edge $e_{i,j}$ between each pair of vertices (v_i, v_j) with a corresponding weight $c_{i,j} \in C$ obtained from the similarity between the feature vectors. The weight on the edge $e_{i,j}$ is computed using the kernel function $c_{i,j} = \exp(-dist(\mathbf{x}_i, \mathbf{x}_j)/2\sigma_3^2)$ with kernel width parameter σ_3. The distance between two feature vectors $dist(\mathbf{x}_i, \mathbf{x}_j)$ that are point process features in this case is computed using the van Rossum method as explained in [1].

Graph G can be partitioned into two sub-graphs A and B, where $A \cup B = V$ and $A \cap B = 0$, by cutting the edges connecting A and B. In graph theory the similarity between two-subgraphs denoted here by $\text{sim}(A, B)$ can be computed by summing the weights of the edges that connect the two as

$$\text{sim}(A, B) = \sum_{v_i \in A, v_j \in B} c_{v_i, v_j}. \qquad (4)$$

Assuming that the graph vertices belong to two classes $y = \{1, -1\}$, one way for classifying the graph vertices is to partition the graph into two sub-graphs that have the minimum similarity. However, as it was shown in [12] partitioning a graph by minimizing Equation 4 can result in having very few vertices in one of the sub-graphs. For balancing the number of vertices in each sub-graph, Shi and Malik developed a new approach known as normalized cuts [13] where an additional constraint was added to the minimization problem. In this approach the similarity between A and B is normalized by the number of edges connecting A and B to the entire graph and the best partitioning is obtained by minimizing

$$\underset{A,B}{\text{argmin}} \frac{\text{sim}(A, B)}{\sum_{v_i \in A, v_j \in G} e_{i,j}} + \frac{\text{sim}(A, B)}{\sum_{v_i \in B, v_j \in G} e_{i,j}}. \qquad (5)$$

It is shown in [13] that the optimization problem of Equation 5 can be solved by solving the following problem

$$\underset{\mathbf{s}}{\text{argmin}} \frac{\mathbf{s}^T (D - W) \mathbf{s}}{\mathbf{s}^T D \mathbf{s}}, \qquad (6)$$

where $\mathbf{s}$ is an N-dimensional vector containing the class labels $\{-1, 1\}$ assigned to each of the N vertices. The matrix D is an $N \times N$ diagonal matrix with elements $D_{i,i} = \sum_{j=1}^{N} c_{i,j}$. If $\mathbf{s}$ is relaxed to take on real values, Equation 6 can be minimized by solving the generalized eigenvalue system,

$$(D - W)\mathbf{s} = \lambda D \mathbf{s}. \qquad (7)$$

It is worth noting that this optimization problem is actually equivalent to the one described in Equation 1 when there are no labeled examples (i.e. $l = 0$) and no ambient regularization ($\gamma_A = 0$). The eigenvector of the above system corresponding to the minimum eigenvalue is the solution to the optimization problem in Equation 6. However, the above system has a trivial solution where the smallest eigenvalue is zero ($\lambda_0 = 0$) and the corresponding eigenvector ($\mathbf{s}_0 = \mathbf{1}$) assigns all the vertices to one class. The non-trivial solution is the eigenvector $\mathbf{s}_1$ corresponding to the second smallest eigenvalue λ_1. After computing $\mathbf{s}_1$, its elements are thresholded and values above and below the threshold are replaced by $\{-1, 1\}$. Next, the vertices are assigned a class label of 1 or -1 based on the value of their corresponding dimensions in $\mathbf{s}_1$. The class label of each vertex v_i is then given to the corresponding candidate interval $I_i(Q)$. Despite having classified the candidate intervals into two classes, $\{1, -1\}$, it is not clear which class corresponds to true occurrence and which one to false alarm. To determine the identity of the classes, all is needed is the actual label of an arbitrary candidate interval.

5. Experimental Setup

All the experiments in this paper were conducted on the Augmented Multi-party Interaction (AMI) corpus [14]. The AMI corpus consist of speech from a set of meetings recorded with multiple microphones. A subset of the audio files recorded using lapel microphones was used in this study. This subset contains approximately 80 hours of audio from 118 meetings with 121 speakers. For training, development, and evaluation purposes the recordings were split into three equivalently sized folds, each consisting of recordings from a unique set of speakers. The fold 1 was used for training, fold 2 for development, and fold 3 for evaluation. A set of 35 content words with a high frequency of occurrence in all three folds were selected as query terms. The word query list consists of short monosyllabic words (e.g., "*shape*") as well as long multisyllabic words (e.g., "*presentation*").

Three LVCSR systems were trained on fold 1, fold 2 and the combination of fold 1 and fold 2 as described in [1] to generate word lattices for fold 2, fold 1, and fold 3 respectively. A word accuracy of 56.9% was obtained for fold 3 using the LVCSR system trained on folds 1 and 2. All the occurrences of the query terms were then extracted from the lattices of the three folds and the occurrences with substantial overlap were merged. The list of hypothesized occurrences of the query terms in folds 1 and 2 contained 109433 occurrences out of which 9631 corresponded to actual occurrences and the rest were false alarms. For fold 3 there were 34013 occurrences of the query terms with 3766 of them corresponding to actual occurrences. A fixed-dimension point process representation was generated for each of the occurrences following the recipe described in [1].

The feature representations of the hypothesized occurrences were used for classifying them into true occurrence and false alarms using the classifiers described in Section 3 and Section 4. A set of globally optimum parameters were obtained for the classifiers based on their performance on the dev set (fold 2). For the RLS classifier the optimum value derived for the first and second regularization factors are $\gamma_A = 1/(l + u)$ and $\gamma_I = 10^7.5/(l + u)$, where $l + u$ is the total number of labeled and unlabeled training samples. The kernel width parameters σ_1 used for computing $W_{i,j}$ in Equation 1 and σ_2 used for computing $K(\mathbf{x}_i, \mathbf{x}_j)$ in Equation 2 were set to 0.4 and 0.15 respectively.

In the spectral clustering approach for classifying the hypothesized occurrences first a graph was constructed for each of the query terms using the feature representation of the hypothesized occurrences. Next, all vertices of each graph were connected with weights obtained from the similarity of the corresponding feature representations using the kernel function described in Section 4. The kernel width parameter used in all the graphs was 0.2. Once the graphs are constructed, the generalized eigenvalue problem of Equation 7 was solved for each graph and the vertices of the graphs were clustered into two classes. The identity of the classes (true occurrence or false alarm) in each graph was then determined using the actual label of an arbitrary vertex.

6. Experimental Evaluation

In this section the performance of the semi-supervised RLS and the spectral clustering approaches in verifying hypothesized occurrences of a set of query terms is evaluated. Moreover, a comparison between the performance of the semi-supervised classifiers and the supervised RLS classifier is made for vari-

Classification	Labeled Train	Unlabeled Train	Labeled Test	EER
Supervised RLS	2%	0	0	23.2
Semi-Supervised RLS	2%	98%	0	20.7
Spectral Clustering	0	0	1	20.0

Table 1: EER obtained from spectral clustering and RLS classifiers.

ous ratios of labeled to unlabeled training samples. It should be noted that the samples contain both true occurrences and false alarms. Here, the verification performance is measured using the equal error rate (EER) measure. Since different training samples affect the classifiers performance differently, the experiments were repeated 10 times, each time with a randomly selected labeled training samples. The mean EER was then computed for each of the query terms for each ratio. Figure 1 illustrates the mean EER performance averaged over the 35 query terms for both semi-supervised and supervised classifiers trained with various ratios of labeled to unlabeled samples.

The top and the bottom curves in Figure 1 correspond to the performance of the supervised and semi-supervised RLS classifiers respectively. The single operating point depicted by a diamond on the vertical axis is the average EER obtained from the spectral clustering method with using the actual label of one arbitrary test sample. A number of observations can be made from this plot. First, it can be seen that the performance of the semi-supervised RLS is always better than the supervised RLS with the same number of labeled training samples. This is more evident as the ratio of the unlabeled to labeled samples increases. In the range from 30 to 80 percent labeled, manifold regularization reduces the labeled data requirement by 10% to achieve a given performance level. For the task studied here, the 10 percent of training samples amounts to approximately 300 samples for each query term.
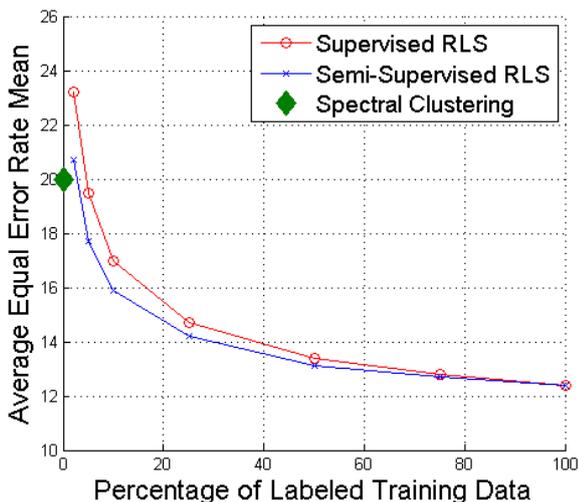

Figure 1: *The average EER mean obtained from supervised and semi-supervised classifiers trained with various ratios of labeled to unlabeled samples.*

In Table 1 the performance of spectral clustering and RLS classifiers trained using 2% labeled training data is illustrated. Looking at the first two rows of this table, it shows that manifold regularization of the RLS classifiers results in a relative EER

improvement of 11% in this very low resource regime. Also, it can be seen that the spectral clustering approach provides a classification performance comparable to the semi-supervised RLS classifier by using only 1 labeled test sample. This

7. Conclusions

In this paper two semi-supervised classification approaches for the task of spoken term verification were investigated. First, the performance of a manifold based semi-supervised training of RLS classifiers was evaluated for this task. It was shown that with the same number of labeled training data, the RLS classifiers trained using the semi-supervised approach always provide better verification performance than the ones trained using the supervised approach. Secondly, the performance of the graph spectral clustering approach for term verification in a very low resource scenario was evaluated. It was shown that, verification using the spectral clustering algorithm based on the actual label of a single arbitrary test sample is as effective as using semi-supervised RLS classifiers trained with a very small number of labeled training samples. This suggests that, in verification scenarios where the amount of data available for training term-specific models is very limited, the spectral clustering approach could alternatively be used.

8. Acknowledgements

9. References

[1] A. Norouzian, A. Jansen, R. Rose, and S. Thomas, "Exploiting discriminative point process models for spoken term detection," in *Proceedings of INTERSPEECH*, 2012.

[2] Lin-shan Lee Huny-yi Lee, Po-wei Chou, "Open-vocabulary retrieval of spoken content with shorter/longer queries considering word/subword-based acoustic feature similarity," in *INTERSPEECH*. ISCA, 2012.

[3] Tsung-Wei Tu, Hung-Yi Lee, and Lin-Shan Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," in *Proc. ASRU*, 2011.

[4] A. Norouzian, R. Rose, S. Hamidi Ghalehjegh, and A. Jansen, "Zero resource graph-based confidence estimation for open vocabulary spoken term detection," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013.

[5] A. Norouzian and R. Rose, "An efficient approach for two-stage open vocabulary spoken term detection," in *IEEE Workshop on*

Spoken Language Technology Proc., 194-199, 2010. IEEE, 2010, pp. 194–199.

[6] D. Can and M. Saraçlar, "Lattice indexing for spoken term detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2338–2347, 2011.

[7] O. Siohan and M. Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *Ninth European Conference on Speech Communication and Technology*. ISCA, 2005.

[8] P. Yu and F. Seide, "Fast two-stage vocabulary-independent search in spontaneous speech," in *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2005, vol. 5, pp. 481–484.

[9] A. Jansen, "Whole word discriminative point process models," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5180–5183.

[10] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: volume I: Elementry Theory and Methods*, vol. 1, Springer, 2003.

[11] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[12] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[14] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.