A Comparison of Neural Network Feature Transforms for Speaker Diarization

Sree Harsha Yella *Andreas Stolcke*

Idiap Research Institute
Martigny, Switzerland
shyella@idiap.ch

Microsoft Research
Mountain View, CA, U.S.A.
anstolck@microsoft.com

Abstract

Speaker diarization finds contiguous speaker segments in an audio stream and clusters them by speaker identity, without using a-priori knowledge about the number of speakers or enrollment data. Diarization typically clusters speech segments based on short-term spectral features. In prior work, we showed that neural networks can serve as discriminative feature transformers for diarization by training them to perform same/different speaker comparisons on speech segments, yielding improved diarization accuracy when combined with standard MFCC-based models. In this work, we explore a wider range of neural network architectures for feature transformation, by adding additional layers and nonlinearities, and by varying the objective function during training. We find that the original speaker comparison network can be improved by adding a nonlinear transform layer, and that further gains are possible by training the network to perform speaker classification rather than comparison. Overall we achieve relative reductions in speaker error between 18% and 34% on a variety of test data from the AMI, ICSI, and NIST-RT corpora.

Index Terms: speaker diarization, artificial neural networks, discriminative feature extraction.

1. Introduction

Speaker diarization answers the question "who spoke when" in a multiparty conversation, i.e., it aims to identify all speech coming from the same speaker, without prior knowledge of the number of speakers or samples of their speech [1, 2]. Diarization has been studied in various domains such as broadcast news [3], telephone calls [4], and on spontaneous meeting room conversations [2, 5, 6]. The main issues in performing speaker diarization of meeting room recordings arise due to far-field audio (background noise and room reverberation) and conversational speaking style (short speaker turns, interruptions, and overlaps).

State of the art systems for speaker diarization use an agglomerative (bottom-up) clustering framework [7, 6]. These systems typically use short-term spectral characteristics, such as Mel-frequency cepstral coefficients (MFCCs) to represent the vocal tract characteristics of a speaker, as features for diarization. MFCCs are not optimized for speaker discrimination as they reflect various other factors such as channel characteristics, ambient noises, and phonemes being spoken. To overcome this, factor-analysis based techniques, such as i-vectors, which are popular in the speaker-verification domain, have been adapted to the speaker diarization task [8], but so far have had success only for two-party telephone conversations. The same is true of approaches using linear discriminant analysis (LDA) to obtain discriminative features [9]. In another approach [10], information bottleneck (IB) features derived from an initial pass of IB diarization system [6] were used to improve MFCC-based speaker diarization.

In prior work [11] we have proposed using an artificial neural network (ANN) trained as a classifier to extract features for diarization. In order to induce speaker-discriminative features, we trained the ANN classifier to perform speaker comparison: decide whether two given speech segments belong to the same or different speakers, and then use the input-to-hidden weights learned by the network as a feature transform on test data. The resulting features are combined at the level of Gaussian likelihoods with the standard cepstral features, and yield substantial error reductions on test data that is well-matched to the training data. In this work we further explore the general idea of ANN-induced features for diarization, by considering a wider range of network architectures and training criteria. First, we consider a "deeper" version of the speaker comparison network, with added hidden layer and the ability to learn a nonlinear feature transform. Second, we examine an alternative ANN trained to perform speaker classification (rather than comparison), as was previously explored for the speaker verification task [12]. Finally, we consider an auto-associative network (autoencoder) as a baseline for ANN-based feature transform learning. The features resulting from all these architectures are evaluated by themselves and in conjunction with baseline cepstral features (MFCCs), using speaker diarization on commonly used meeting speech corpora.

2. ANN Features for Speaker Diarization

Artificial neural networks are extensively used in supervised tasks such as automatic speech recognition and speaker identification/verification tasks. In these applications, neural networks are trained to predict the posterior probabilities of the desired classes (phonemes and speakers, respectively). The posterior probabilities obtained from a neural network can be directly used to infer the class. Another way of using neural networks for these tasks is to use a network trained to identify the classes as a discriminative feature extractor. Here, the activations of the hidden layer prior to the final layer are used as input features to another classifier (such as an HMM/GMM). An example of this approach is Tandem acoustic modeling in speech recognition [13]. The motivation behind the later approach is to combine the discriminative power of neural networks with the state-of-the-art statistical systems which are typically based on the HMM/GMM framework.

In the current work, we follow a similar approach where we explore three different neural network architectures as feature extractors for speaker diarization. The neural networks are trained to perform tasks related to speaker diarization, such as

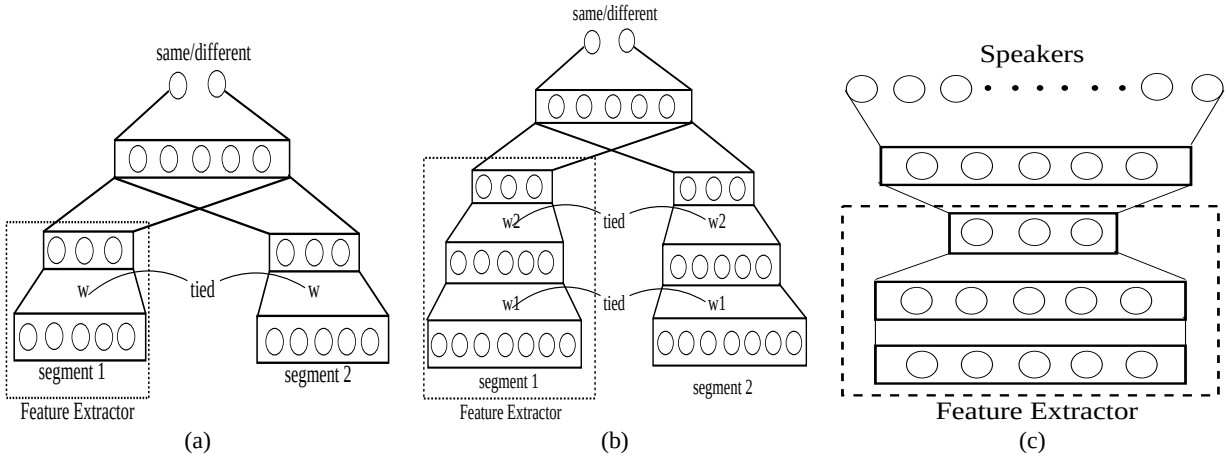

Figure 1: Various ANN architectures used to generate features for speaker diarization: (a) Shallow speaker comparison ANN (b) Deep speaker comparison ANN (c) Speaker classification ANN.

speaker comparison or classification; we also investigate autoencoding as an additional baseline.

2.1. ANNs for speaker comparison

Following the approach in [11], we train a neural network for classifying two given speech segments as belonging to the same or different speakers. We trained two such comparator networks with different numbers of hidden layers: a *shallow speaker comparison network* with two hidden layers, and a *deep speaker comparison network* with three hidden layers.

2.1.1. Shallow speaker comparison network

Figure 1(a) shows the architecture of the shallow speaker comparison network. We split the input layer of the network into two halves, left and right, to represent acoustic features belonging to the two speech segments being compared. The first hidden layer (bottleneck) is also split into two halves similar to the input layer, so each half receives input from the respective input segment. We tie the weight matrices (denoted by *W* in Figure 1(a)) connecting the right and left halves of input and hidden layers so that the network learns a common transform for all speakers. The second hidden layer connects both halves of the first hidden layer to the output layer. The output layer has two units denoting the class labels—-same or different speakers— depending on the source of the two input speech segments (segment1, segment2 in Figure 1(a)). All the hidden layers have sigmoid activation functions; the output layer has a softmax function to estimate the posterior probabilities of the classes (same/different). The network is trained using a cross-entropy objective function.

After training the network, we use the first hidden layer activations, before applying the sigmoid function, as features for speaker diarization in a HMM/GMM system. To generate features from the network, we use a window of speech as input to one half of the input layer and extract activations at the corresponding half of the bottleneck layer. Also, since the features are extracted from the first hidden layer before applying the sigmoid nonlinearity, they represent a linear transform of the MFCC vector at the input. Below we refer to this network and resulting features as *spkr-com*.

2.1.2. Deep speaker comparison network

The deep speaker comparison network contains three hidden layers. When compared to shallow speaker comparison ANN, it contains an extra hidden layer before the bottleneck layer (from which features are extracted). As a result, the features extracted from this network (activations at the bottleneck layer) undergo a nonlinear transform before the bottleneck layer (second hidden layer). The architecture of the network is shown in Figure 1(b). It is similar to that of the network shown in Figure 1(a) except that it has an extra hidden layer before the layer from which the features are extracted. As before, the left and right halves of the hidden layer weights up to the bottleneck are tied.

Once the network is trained the features are extracted by feeding speech segments to one (say, the left) half of the network and obtaining the activations from the second hidden layer (bottleneck layer) of the respective half before applying the sigmoid nonlinearity. Below, we refer to this network and resulting features as *Dspkr-com*.

2.2. ANN for speaker classification

Konig et al. [12] used a multilayer perceptron (MLP) with five layers, trained to classify speakers, as a feature extractor for speaker recognition. The MLP was discriminatively trained to maximize speaker recognition performance. They used the outputs from the second hidden layer (units of which had linear activation function) as features in a standard GMM-based speaker-recognition system.

In the current work, we trained a similar network with speakers as output classes as shown in Figure 1(c). The network is trained by providing a frame along with its context as input and the corresponding speaker as the output class label. The output layer has a softmax function to estimate the posterior probability of the speaker. The second hidden layer (bottleneck) has linear activation functions, and the units in the rest of the hidden layers have sigmoid nonlinearities.

After training the network, the hidden layer activations obtained from the bottleneck layer (second hidden layer) are used as features in speaker diarization. The network performs a nonlinear transform of the input features as they are fed through sigmoid activation function in the first hidden layer. We refer to this ANN and the resulting features as *spkr-class*.

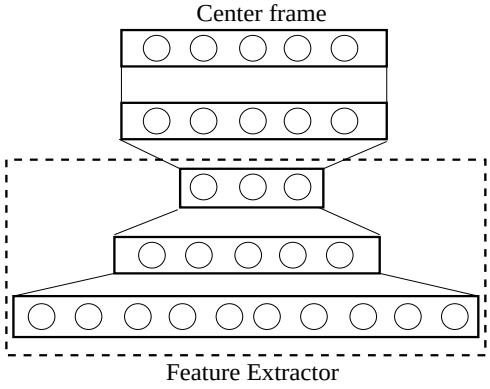

Figure 2: *Autoencoder: The network reconstructs the center frame of the input (center frame + context) at the output layer.*

2.3. Autoencoder ANN

Autoencoders are used in the literature to generate feature representations and for nonlinear dimensionality reduction [14]. An autoencoder encodes the input in a representation which in turn is used to reconstruct the input. Therefore, in training the output targets are the inputs themselves. In the current work, we use an autoencoder with three hidden layers, as depicted in Figure 2. Unlike in standard autoencoders, the input comprises not just the current frame but also includes a window of context as used by the other network architectures presented earlier; this ensures that all networks have the same input information at their disposal. The network is trained to reconstruct the current frame at the output with as little reconstruction error as possible, as measured by mean squared error.

Once the network is trained, the features are generated by giving an input frame with its context as input to the network and obtaining the activations of the second hidden layer before applying the sigmoid nonlinearity. Similar to deep speaker comparison and speaker classification ANNs, this network performs a nonlinear transform of the input features, albeit with an objective function that is not directly related to the speaker diarization task. In the experiments described later we refer to this network and features as *autoen*.

3. Experiments and Results

3.1. Data sets

Our experiments make use of meeting room recordings from several corpora: AMI [15], ICSI [16], and 2006/2007/2009 NIST-RT [17]. Table 1 summarizes the characteristics of these data sets. The AMI data set is split into train and test sets of 148 and 12 meetings, respectively, such that they are disjoint in speakers. Out of the ICSI corpus, 20 meetings are set aside for development and tuning ("dev"), and the remaining 55 ICSI meetings form an additional test set. The combined 2006, 2007, and 2009 NIST-RT evaluation sets are also used for testing.

3.2. ANN training

We trained the ANNs using data from the AMI corpus. To avoid skewing the training towards particular speakers, we sampled 50 utterances from each of 138 speakers used for training. Each utterance has a duration of around 10 seconds. We manually aligned speech transcripts to the close-talking microphone recordings to obtain frame-level speaker labels. For training purposes we removed speech segments containing overlapping speech. For input features we extracted 19 MFCCs from a frame of 30 ms with a frame increment of 10 ms. The features are extracted from the audio signal captured by one of the single distant microphones (SDM) used to record the meetings.

Table 1: *Meeting corpus statistics as used in experiments.*

Corpus	Speakers	Sites	Meetings		
			Train	Dev	Test
AMI	150	3	148	-	12
ICSI	50	1	-	20	55
NIST-RT	100	6	-	-	24

The objective function for the ANN training was cross-entropy for the speaker comparison (both shallow and deep) and speaker classification networks and mean square error for the autoencoder network. Training used error backpropagation and stochastic gradient descent for 25 epochs. The ANNs are trained with inputs with different context lengths and the optimal context for each network is obtained by minimizing diarization error on the ICSI development set. The dimension of the feature extraction (bottleneck) layer is fixed to 20 in all the networks to be similar to that of the MFCCs. The dimension of the second hidden layer for deep speaker comparison (Dspkr-com), speaker classification (spkr-class) and autoencoders (autoen) is fixed to 512 (for each half in case of Dspkr-com). The dimension of the last hidden layer in all the networks is fixed to 100. The number of output units is two in speaker comparison ANNs (both Dspkr-com & spkr-com), 138 in speaker classification ANN (spkr-class), and 19 for the autoencoder (autoen).

When sampling training data for the ANNs, we allow the input window to contained speech from a single speaker only. In testing, on the other hand, the context part of the input window might contain nonspeech and speech from other speakers. In separate experiments, we did try presenting test-like heterogeneous speech input during training as well, but found the results to be worse. Therefore, it seems that pure, speaker-homogeneous training data is more important than the mismatch between training and test conditions that this entails.

3.3. Speaker diarization experiments

We now report the speaker diarization results based on the bottleneck features obtained using the various ANN architectures. All bottleneck features are compared against the baseline 19-dimensional MFCC features. The MFCCs are extracted from the single distant microphone (SDM) audio signal of the meetings. The speaker diarization system is based on the HMM/GMM framework [7] that has been shown to give state-of-the-art performance in several NIST-RT evaluations. The diarization output is evaluated using a metric called diarization error rate (DER), which is the standard metric used in NIST-RT evaluations [18]. DER is the sum of speech/non-speech error and speaker error. Speech/non-speech error is the sum of miss and false alarm errors by the automatic speech/non-speech detection system. Speaker error is the portion of speech time for which the speaker is labeled incorrectly (under the best possible mapping of output to true speaker labels). A forgiveness collar of ± 0.25 seconds is applied around the reference segment boundaries while scoring the automatic systems' output. Since all comparisons between systems involve a shared speech/non-speech segmentation (which is either the reference or automatically determined) we will be reporting only speaker error fig-

Table 2: *Optimal context length (cntxt) in the number of frames and optimal feature stream weights (used while combining with MFCCs) for different ANN features (bnck-wt) based on tuning experiments on the ICSI dev set.*

ANN	spkr-com	Dspkr-com	spkr-class	autoen
cntxt	20	10	10	50
bnck-wt	0.5	0.1	0.7	0.9

Table 3: *Speaker errors on test data sets for various bottleneck features.*

Test-set	spkr-com	Dspkr-com	spkr-class	autoen	MFCC
AMI-test	22.9	**21.8**	29.3	25.9	24.8
ICSI-test	23.1	24.3	**19.8**	20.9	**19.8**
NIST-RT	21.3	20.4	21.5	**12.5**	14.3

ures here.

To identify the optimal input context length for ANNs to generate features for diarization, we performed tuning experiments on the ICSI dev meetings. The optimal context lengths for the various ANN architectures are summarized in the second row of Table 2, and were subsequently used in all experiments.

Table 3 shows the speaker errors for different bottleneck features and the MFCC baseline features. In these experiments we use speech regions obtained from ground-truth segmentation as input to the speaker diarization system.

We observe from Table 3 that the bottleneck features from shallow (spkr-com) and deep (Dspkr-com) speaker comparison ANNs give lower speaker error than MFCC features on AMI test data and increase the error on ICSI-test and NIST-RT data sets. Note that AMI-test is drawn from the same corpus as the ANN training set, and is therefore best matched to the training condition (though the speakers are disjoint). Bottleneck features from the autoencoder, by contrast, produce lower error on RT test data. The bottleneck features obtained from the speaker classification ANN increase error on all but the ICSI test set. In summary, we find that none of the ANN features by themselves perform consistently better than MFCCs.

In spite of this initially disappointing result, we can hypothesize that the bottleneck features capture some information that is complementary to that in the MFCC features, and could still be helpful for speaker diarization when combined with the baseline features. To test this hypothesis, we combine the MFCC features with bottleneck features at the model level [19]: separate GMM models are estimated for each feature stream, for every cluster (state), and the overall cluster log-likelihoods are obtained as a weighted combination of the log-likelihoods according to individual feature streams. The combination weights sum to unity and are fixed by tuning speaker error on the ICSI dev data. The third row (bnck-wt) of Table 2, shows the optimized weights assigned to bottleneck features for the various ANNs types; these are subsequently used while performing multistream diarization on the test sets.

Table 4 shows the results obtained using combined MFCC and ANN features. We observe that the combination M+Dspkr-com decreases the speaker error on all test sets when compared to the MFCC features. It also performs better than the M+spkr-com combination on all the test sets except on the AMI-test set. The combination of MFCCs and autoencoder bottleneck features (M+autoen) does not show significant changes from the single stream autoencoder system (cf. autoen in Table 3). This

Table 4: *Speaker errors on test data sets after combining different bottleneck features with MFCCs. The final row shows results with automatic speech activity detection.*

Test-set	M+ spkr-com	M+ Dspkr-com	M+ spkr-class	M+ autoen	MFCC
AMI-test	19.7	23.1	**19.2**	24.9	24.8
ICSI-test	18.5	15.6	**13.1**	21.1	19.8
NIST-RT	17.3	**11.7**	11.9	14.3	14.3
NIST-RT-SAD	16.0	**11.3**	12.2	12.7	14.2

shows that the autoencoder bottleneck features are not capturing information that is complementary to MFCCs. The combination of MFCCs with speaker classification features (M+spkr-class) produces a significant decrease in speaker error on all test sets, and the largest error reduction on the ICSI test set (34% relative).

Finally, we also perform speaker diarization experiments using speech regions obtained from an automatic speech/non-speech detection system, which is the more realistic application scenario. We perform these experiments on the NIST-RT data set. Speech activity detection (SAD) is performed using the SHOUT toolkit [20]. The total speech/non-speech error was 7.7%, which includes a missed speech error of 7.3% and a false alarm error of 0.4%. The results in the last row of Table 4 (NIST-RT-SAD) show that the combination of MFCCs and deep speaker comparison features gives the best diarization output, reducing the speaker error from 14.2% (MFCCs) to 11.3%.

4. Conclusions

Our results confirm the effectiveness of ANN-trained feature transforms for speaker diarization and show improvements over previous work. Adding an additional, nonlinear hidden layer to the ANN trained for speaker comparison results in substantial error reduction over the earlier, linear-transform ANN features. In particular, it yields improvements over the baseline for all our test sets, both matched (AMI and ICSI) and mismatched (NIST-RT); all improvements are obtained by combining transformed and baseline features (MFCCs) at the level of model likelihoods. Speaker error on NIST-RT test data processed with automatic speech activity detection is reduced by 20% relative.

An alternative training criterion (first suggested for speaker verification [12]) induces a nonlinear feature transform by training the ANN to perform speaker classification, and also results in error reductions over baseline; though the improvements on NIST-RT data are not as large as with speaker comparison training. For additional comparison, we also trained a bottleneck feature extractor based on autoencoder ANNs. While the resulting bottleneck features give some gains over the baseline on RT data, they do not improve in combination with the baseline, and are worse than multistream features based on speaker comparison and classification training, confirming the importance of discriminative training related to the diarization task.

5. Acknowledgment

We thank Malcolm Slaney for many useful discussions and contributions to prior work.

6. References

[1] S. Tranter and D. Reynolds, "An overview of automatic speaker diarization systems", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1557–1565, Sep. 2006.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 356–370, Feb. 2012.

[3] D. Moraru, S. Meignier, C. Fredouille, L. Besacier, and J.-F. Bonastre, "The ELISA consortium approaches in broadcast news speaker segmentation during the nist 2003 rich transcription evaluation", *in Proc. IEEE ICASSP*, vol. 1, pp. I–373–6 vol.1, May 2004.

[4] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis", *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 1059–1070, Dec. 2010.

[5] X. Anguera, *Robust speaker diarization for meetings*, PhD thesis, Universitat Politecnica de Catalunya, 2006.

[6] D. Vijayasenan, F. Valente, and H. Bourlard, "An information theoretic approach to speaker diarization of meeting data", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, pp. 1382–1393, 2009.

[7] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system", *in Multimodal Technologies for Perception of Humans*, pp. 509–519. Springer-Verlag, Berlin, Heidelberg, 2008.

[8] S. Shum, N. Dehak, E. Chuangsuwanich, D. A. Reynolds, and J. R. Glass, "Exploiting intra-conversation variability for speaker diarization", *in Proc. Interspeech*, pp. 945–948, Florence, 2011.

[9] I. Lapidot and J.-F. Bonastre, "Integration of LDA into a telephone conversation speaker diarization system", *in IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI)*, pp. 1–4, 2012.

[10] S. H. Yella and F. Valente, "Information bottleneck features for HMM/GMM speaker diarization of meetings recordings", *in Proc. Interspeech*, pp. 953–956, Florence, 2011.

[11] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial neural network features for speaker diarization", *in Proc. IEEE Spoken Language Technologies Workshop*, pp. 402–406, South Lake Tahoe, USA, 2014.

[12] Y. Konig, L. Heck, M. Weintraub, K. Sonmez, and R. E. E, "Nonlinear discriminant feature extraction for robust text-independent speaker recognition", *in Proc. RLA2CESCA Speaker Recognition and its Commercial and Forensic Applications*, pp. 72–75, 1998.

[13] D. P. W. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition", *in Proc. IEEE ICASSP*, vol. 1, pp. 517–520, Salt Lake City, May 2001.

[14] Y. Bengio, "Learning deep architectures for AI", *Found. Trends Mach. Learn.*, vol. 2, pp. 1–127, 2009.

[15] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The AMI meeting corpus", *in Proc. Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, 2005.

[16] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus", *in Proc. IEEE ICASSP*, pp. 364–367, Hong Kong, 2003.

[17] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The Rich Transcription 2007 meeting recognition evaluation", in R. Stiefelhagen, R. Bowers, and J. Fiscus, editors, *Multimodal Technologies for Perception of Humans. International Evaluation Workshops CLEAR 2007 and RT 2007*, vol. 4625 of *Lecture Notes in Computer Science*, pp. 373–389. Springer, Berlin, 2008.

[18] National Institute of Standards and Technology, "Rich transcription", http://nist.gov/speech/tests/rt/, 2003.

[19] J. Pardo, X. Anguera, and C. Wooters, "Speaker diarization for multiple-distant-microphone meetings using several sources of information", *IEEE Trans. Comput.*, vol. 56, pp. 1189–1224, Sep. 2007.

[20] M. Huijbregts and F. de Jong, "Robust speech/non-speech classification in heterogeneous multimedia content", *Speech Communication*, vol. 53, pp. 143–153, 2011.