Kent Academic Repository

Full text document (pdf)

Citation for published version

Phan, Huy and Hertel, Lars and Maass, Marco and Mazur, Radoslaw and Mertins, Alfred (2016) Learning Representations for Nonspeech Audio Events through Their Similarities to Speech Patterns. IEEE/ACM Transactions on Audio, Speech and Language Processing, 24 (4). pp. 807-822. ISSN 2329-9290.

DOI

https://doi.org/10.1109/TASLP.2016.2530401

Link to record in KAR

https://kar.kent.ac.uk/72684/

Document Version

Author's Accepted Manuscript

Learning Representations for Nonspeech Audio Events through Their Similarities to Speech Patterns

Huy Phan*, *Student Member, IEEE,* Lars Hertel, Marco Maass, *Student Member, IEEE,*
Radoslaw Mazur, *Member, IEEE,* and Alfred Mertins, *Senior Member, IEEE*

***Abstract*—The human auditory system is very well matched to both human speech and environmental sounds. Therefore, the question arises whether human speech material may provide useful information for training systems for analyzing nonspeech audio signals, for example, in a classification task. In order to answer this question, we consider speech patterns as basic acoustic concepts which embody and represent the target nonspeech signal. To find out how similar the nonspeech signal is to speech, we classify it with a classifier trained on the speech patterns and use the classification posteriors to represent the closeness to the speech bases. The speech similarities are finally employed as a descriptor to represent the target signal. We further show that a better descriptor can be obtained by learning to organize the speech categories hierarchically with a tree structure. Furthermore, these descriptors are generic. That is, once the speech classifier has been learned, it can be employed as a feature extractor for different datasets without re-training. Lastly, we propose an algorithm to select a sufficient subset which provides an approximate representation capability of the entire set of available speech patterns.**

We conduct experiments for the application of audio event analysis. *Phone triplets* from the TIMIT dataset were used as speech patterns to learn the descriptors for audio events of three different datasets with different complexity, including UPC-TALP, Freiburg-106, and NAR. The experimental results on the event classification task show that a good performance can be easily obtained even if a simple *linear* classifier is used. Furthermore, fusion of the learned descriptors as an additional source leads to state-of-the-art performance on all the three target datasets.

***Index Terms*—feature learning, representation, nonspeech audio event, speech patterns, phone triplets.**

I. Introduction

Besides human speech, the most important audio signal, computational analysis of other nonspeech audio signals (e.g. music [1], [2], environmental sounds [3], [4]) is becoming more and more important [4]–[6]. In this domain, signal representation remains a fundamental problem for many other successive tasks such as classification [1], [7] and detection [2], [8]. Many works have focused on the development of efficient signal representations [7], [9]–[11]. Although considerable progress has been made in individual problems, more often than not, these representations are derived based on analysis of the target signals per se. We still lack a general way of representing audio signals and specifically lack a universal descriptor for them. Such a generic representation would be very helpful for solving various audio analysis tasks in a homogeneous way.

In this work, we propose such a generic descriptor for nonspeech audio events. Using speech patterns as fundamental acoustic concepts, we measure the closeness between the target signal and different speech patterns. To accomplish this, given a set of labeled speech signals of different categories (e.g. speech words or phonemes), we are able to learn a multi-class speech classifier. Here we consider a speech category as a speech pattern, and these two terms will be used interchangeably. Inputting the target nonspeech signal into the learned speech classifier, we obtain the classification posterior probabilities which can be interpreted as the acoustical proximities between the target signal and the speech patterns. In intuition, they measure how much the target signal sounds like the corresponding speech signals. Eventually, we use the speech classifier as a feature extractor, and the speech posteriors are used to describe the target audio signal. The idea is illustrated in Figure 1. The speech signals are obtained from an external source which is totally unrelated to the target audio signals of interest. By collecting a sufficiently large set of basic speech patterns, we are able to cover a wide range of acoustic concepts of the world. As a result, embedding the target audio signal into the space spanned by the similarities to these concepts is expected to produce a good representation.

We investigate random selection of speech patterns and further propose to automatically organize them hierarchically on a learned label tree. By this, we tend to recursively group the similar speech categories into disjoint groups along the tree so that the speech meta-categories (i.e. the speech clusters) are separated from one another. Eventually, we learn multiple binary speech classifiers at the split nodes of the tree and employ them for feature extraction accordingly. These proposed descriptors are generic in the sense that once the feature extractors are trained, they can be used to extract features for any input signal without re-training. This is opposed to other common feature learning methods [7], [10]–[13] which usually adapt to a specific target dataset. Finally, we develop an algorithm to select a *sufficient* subset from the entire available speech patterns to learn representations for a target dataset. This subset can well approximate the representation capability of the entire set, yet the number of categories is much smaller. Thus, it is computationally more efficient.

In the experiments, we used phone triplets, which are the combinations of three successive phonemes, from the TIMIT dataset [14] as the speech patterns, and audio event signals of

H. Phan is with the Institute for Signal Processing, University of Lübeck and the Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, 23562 Lübeck Germany e-mail: phan@isip.uni-luebeck.de.

L. Hertel, M. Maass, R. Mazur and A. Mertins are with the Institute for Signal Processing, University of Lübeck, 23562 Lübeck, Germany e-mails: {hertel,maass,mazur,mertins}@isip.uni-luebeck.de.

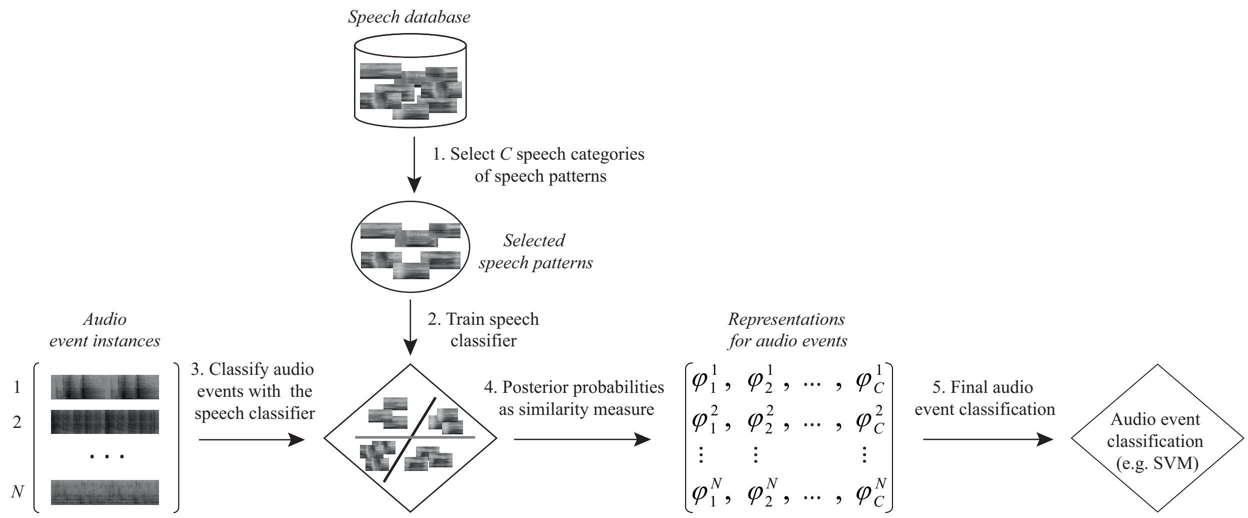
Speech database
1. Select C speech categories of speech patterns
Selected speech patterns
2. Train speech classifier
Audio event instances
1
2
. . .
N
3. Classify audio events with the speech classifier
4. Posterior probabilities as similarity measure
Representations for audio events
$\varphi_1^1, \varphi_2^1, \cdots, \varphi_C^1$
$\varphi_1^2, \varphi_2^2, \cdots, \varphi_C^2$
$\varphi_1^N, \varphi_2^N, \cdots, \varphi_C^N$
5. Final audio event classification
Audio event classification (e.g. SVM)

Figure 1. The general idea can be summarized in five steps. 1) Given a speech database, C speech categories are selected as speech patterns. We study both random selection and some problem-dependent selection algorithms in this work. 2) A speech classifier is then learned using the C selected speech patterns. 3) Given different nonspeech audio event instances, we classify them using the learned speech classifier. 4) The vector of posterior probabilities $(\varphi_1^i, \varphi_2^i \ldots, \varphi_C^i)$ is then used to represent the audio event instance $i \in \{1, \ldots, N\}$ where the posterior probability φ_c^i is considered as the similarity between the audio event instance i and the speech pattern $c \in \{1, \ldots, C\}$. 5) The final audio event classification can be conducted, e.g. by Support Vector Machine (SVM), using the speech-based features.

three different datasets, including UPC-TALP [15], Freiburg-106 [16], and NAR [17] as target nonspeech signals. We show that with a random selection of a reasonably large subset of phone triplets to train the generic feature extractor, we can obtain very good classification accuracies for all three datasets of interest. Furthermore, with the small sufficient subset obtained by the proposed selection algorithm, we can achieve a performance that is very close to the representation capability of the entire set of phone triplets. Our classification accuracies with simple linear classifiers outperform the state-of-the-art accuracies on two out of three datasets while maintaining marginal gaps to those of the strong baselines (i.e. bag-of-words and pyramid bag-of-words models). This is impressive given the fact that with these descriptors alone, we have not incorporated the features obtained by the target signals themselves into the models. By considering the proposed descriptors as external sources and integrating them with the existing baselines, our fusion systems set state-of-the-art performance on all datasets, i.e. they outperform all the baselines and state-of-the-art systems.

The rest of this paper is organized as follows. Some related works on audio event representation are briefly presented in Section II. After that, in Section III, we describe how to extract the generic descriptors for a nonspeech audio signal given a set of speech patterns. Our proposed algorithm to select a sufficient set of speech patterns is then described in Section IV. The experimental setup and results are presented in Section V followed by the discussion in Section VI and conclusions in Section VII.

II. Related Work

In general, any features that are used to describe an audio signal will be applicable for an audio event classification task. Various hand-crafted descriptors have been proposed. The traditional features for speech recognition like MFCCs [18] and log frequency filter bank parameters [19] have been prevalent. Various other features have also been developed and found useful for audio events, for instance, spectro-temporal features based on spectrograms [9], [20]–[22], wavelet transforms [23], and stabilized auditory images (SAI) [24], [25].

With the rapid advance of machine learning, automatic feature learning is becoming more and more common. Bag-of-words models have been most widely used for audio event representation [10]–[13], [26]. Other codebook-based representations have also been employed, such as sparse coding [27], [28], non-negative matrix factorization (NMF) [29], and exemplar-based coding [30]. There have also been several attempts to encode the structural information of audio events. An audio event can be considered as a sequence of atomic units of sound [31] or symbols [32], [33], and the patterns of occurrences can be used as the signatures in the classification task. Alternatively, the structural information can also be captured with pyramid bag-of-words model [12] and Self-Organizing Maps (SOM) [34]. There is also a trend of applying deep neural networks [10] for automatic feature learning, where the structures are captured on different layers of the network. However, while it has been shown a breakthrough for speech recognition [10], [35], [36], the application on audio events is mainly to gain the robustness [37], [38] under noisy conditions.

The hand-crafted and learned features are different in some aspects. Firstly, the hand-crafted features (e.g. MFCCs, spectrograms) are generic. That is, the feature extraction process is the same for different datasets. On the contrary, the goal of the feature learning methods is to induce feature spaces that are empirically most discriminative for the audio events under analysis. Therefore, they are data-specific. Secondly, the learned features are usually built on top of hand-crafted features, hence, they are of higher semantic level and often enjoy better recognition accuracy [7], [12] or better robustness [37]. Our proposed learning approach, on the other hand, fundamentally differs from these conventional learning approaches. Instead of producing a data-specific feature space, we make use of the existing feature space that is formed by speech patterns and just need an appropriate method to discover. Finally, our learned features remain generic as is the hand-crafted case rather than specific for a certain dataset. Once the feature extractors are learned, we can use them to extract representations for any inputted audio signal such as music, audio events, etc. They are different from other features learned in a conventional way, such as bag-of-words representations [11], [12], [39], which are task-specific and data-specific.

Our work is also linked to enhanced posteriors for speech recognition in the field of automatic speech recognition (ASR) [40]–[42]. In order to improve the speech recognition task, the posterior probabilities (i.e. phone or word levels) obtained by a first discriminative classification layer (for example, using Multi-Layer Perceptrons (MLPs) [40], Deep Neural Networks (DNNs) [35], and Convolutional Neural Networks (CNNs) [43]) are then used either as local acoustic scores to estimate the emission probabilities required in Hidden Markov Models (HMMs) [35], [40], [42], [43] or as acoustic features in the Tandem approach [41]. Our approach makes use of discriminately trained speech classifiers to derive posterior probabilities as representations for nonspeech audio events.

Finally, our work bears some resemblances to those exploring additional data sources (e.g. multiple channels [44], multiple modalities [45]) to augment the nonspeech audio event analysis. However, their main goal is to compensate for low signal-to-noise-ratio and overlapping signals. Therefore, not surprisingly, the additional data are of the same kind as the target signals under analysis. Our goal, however, is to learn representations for a target audio signal via external speech signals that are totally unconnected to the target signal.

This is an extension of our preliminarily work [46] in which we showed that human speech signals at the word level can be used to learn representations for nonspeech audio events. The extension includes using phone triplets as an alternative for words which is experimentally shown more appropriate, the generalization of the results on different nonspeech audio event datasets, further analysis in greater detail, and the selection algorithm for a sufficient subset of speech patterns.

III. Learning speech-based descriptors for nonspeech audio signals

In the following, we first describe the low-level features that are employed to represent an audio signal (i.e. both speech and nonspeech). Afterward, we propose two types of generic speech-based descriptors for nonspeech audio signals, *flat* descriptors in Section III-B and *tree-induced* descriptors in Section III-C, that are built on top of the low-level features. Finally, we also compare our proposed approach to conventional speech models under the perspective of the field of ASR.

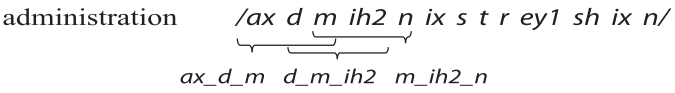

Figure 2. **An example of phone triplets.** The word "administration" is decomposed into its constituent monophones. Phone triplets, such as ax_d_m, d_m_ih2, and m_ih2_n, are combinations of three consecutive phones.

A. Phone triplets and low-level acoustic features

There exist different speech levels (e.g. phonemes, words) that may be considered for speech patterns. Whereas the number of single phones is limited, combining them would create more diverged speech patterns, and hence enrich the representation capability. We propose to use phone triplets in this work. We demonstrate some examples of phone triplets in Figure 2. Note that phone triplets are different from triphones that have been commonly adopted in speech recognition task [47]–[50]. A triphone is a single phone that takes into account the previous and successive phones as the context. In contrast, a triplet is the combination of three consecutive phones as a whole. Furthermore, the temporal order of the constituent phones in a triplet is not important. For example, all combinations of three single phones {ax, d, m}, such as d_ax_m, ax_d_m, m_ax_d, etc. belong to the same class in our setup.

There are also other reasons why using phone triplets is more appropriate than the short phone units. Nonspeech audio events are usually long signals (in the order of some hundred miliseconds up to several seconds), which are much longer than phone units. Phone triplets, which are longer speech segments (i.e. a phone triplet is about three times longer than a triphone), are more compatible to long nonspeech signals than the single phones alone. Higher orders of combination would also be appropriate but they require more data. Furthermore, we also need long signal segments to obtain a good estimation of feature standard deviation which is described below.

In order to measure the similarities between speech and nonspeech signals, it is necessary to represent them in a common feature space. The signals (i.e. audio events and speech triplets) were firstly downsampled to 16 kHz. Each audio event was decomposed into 50 ms frames with a step size of 10 ms, whereas the frame length used for speech signals was 25 ms as usual. For speech, 20-30 ms segmentation is common because the signal in a segment is more or less stationary, and the shortest phones (some plosives) have a duration of around 20 ms. However, nonspeech audio events exhibit a wider range of characteristics [8], [51], and for the event recognition task it is important to recognize the event as a whole and not every single 20 ms fragment of it. Therefore, longer frames appear to be more appropriate than traditional short ones.

Although any arbitrary low-level features are feasible to describe a frame, we extracted a set of very basic acoustic features for every audio frame: 16 log-frequency filter bank coefficients [19], their first and second derivatives, zero-crossing rate, short-time energy, four sub-band energies, spectral centroid, and spectral bandwidth. Totally, there were 53 features for each frame. In turn, a whole segment, either a phone triplet or the signal corresponding to one event, is represented by a 106-dimensional feature vector computed from the mean and standard deviation over its frames.

B. Flat descriptor via speech similarities

Given a database of labeled phone triplet signals $\mathcal{S} = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$, where $\mathbf{x}_i$ denotes the low-level feature vector for the i-th signal as described in Section III-A, and $c_i \in \{1, \ldots, C\}$ indicates the class label where C indicates the number of triplet categories. The number of categories should be as large as possible and ideally include all possible acoustic concepts of the world.

Let us further denote the low-level feature vector of the target nonspeech audio signal as $\mathbf{x}_e$. Our goal is to represent the target signal in terms of its acoustical closeness to the set of C basis speech patterns. We accomplish this using some classifiers trained on the speech patterns. For convenience, we jointly learn a multi-class speech classifier $\mathcal{M}_\mathcal{S}$ at once using random forest classification [52]. Other classification algorithms, e.g. DNNs, may be alternatively used.

After training the speech classifier, the target event $\mathbf{x}_e$ is then inputted into $\mathcal{M}_\mathcal{S}$ to obtain the classification posterior probabilities $\varphi = [\varphi_1, \ldots, \varphi_C] \in \mathbb{R}_+^C$ where

$$\varphi_c = P(c|\mathbf{x}_e). \tag{1}$$

Each entry φ_c quantifies how likely the target event is to the speech category c of $\mathcal{S}$, i.e. φ_c can be interpreted as a similarity measure.

Traditionally, the posterior probabilities produced by the classifier $\mathcal{M}_\mathcal{S}$ are used to make decisions, such as in a recognition task. Here, we employ the classifier $\mathcal{M}_\mathcal{S}$ as a feature extractor, and the vector φ is used as a descriptor for the event $\mathbf{x}_e$. As a result, the audio event is embedded in the space spanned by the speech similarities. In Figure 3, we illustrate the similarities of audio events in the Freiburg-106 dataset [16] to 50 categories of phone triplets of the TIMIT dataset [14]. The phone triplet categories were selected randomly and we trained the classifier $\mathcal{M}_\mathcal{S}$ with 200 trees. We will further describe the experiments in Section V. Although the similarity responses appear to be noisy, different event categories exhibit distinguished patterns, except for the "background" class which shows random responses since it contains many different kinds of sounds.

C. Tree-induced descriptor using a label tree of speech categories

We argue that in order to learn good descriptors, we need to choose a set of varied speech patterns. Armed with expertise, one can carefully select such speech categories by hand. Here, we propose to discover them from a pre-determined set $\mathcal{S}$. We collectively partition the speech categories into disjoint subsets in such a way that they are easy to distinguish from one another. For this purpose, we learn a label tree for the speech categories similarly to [53]. This algorithm was originally proposed to learn a tree structure of classifiers (the label tree). Instead, we use it to form the sets of speech categories that can be easily distinguished from one another. By doing so, we have reduced the complex flat multi-class classification problem into multiple simpler binary classification problems. Training the binary classifiers is easy. Furthermore, we gain the average classification performance which is an important factor to achieve a good representation for nonspeech audio events as shown in Section VI-C.

Let $\ell_\mathcal{S} = \{1, \ldots, C\}$ denote the label set of the speech database $\mathcal{S}$. The label tree is constructed recursively so that each node is associated with a set of class labels. Let us consider a node with a label set ℓ (and therefore, the root node is assigned with the label set $\ell_\mathcal{S}$). We want to split the set ℓ into two subsets ℓ^L and ℓ^R so that

$$\ell^L \neq \emptyset, \tag{2}$$

$$\ell^R \neq \emptyset, \tag{3}$$

$$\ell^L \cup \ell^R = \ell, \tag{4}$$

$$\ell^L \cap \ell^R = \emptyset. \tag{5}$$

There are totally $2^{|\ell|-1} - 1$ possible partitions $\{\ell^L, \ell^R\}$ where $|\cdot|$ denotes the cardinality. We want to select the partition such that a binary classifier to separate ℓ^L and ℓ^R makes as few errors as possible. An exhaustive search for such a partition would be prohibitively expensive especially when $|\ell_\mathcal{S}|$ is large.

Alternatively, we rely on the confusion matrix of a multi-class classifier to determine a good partitioning. Our goal is to include classes that tend to be confused with each other in the same subset. Let $\mathcal{S}^\ell \subset \mathcal{S}$ denote the set of speech signals corresponding to the label set ℓ. Furthermore, suppose that we have changed and sorted the label set ℓ so that $\ell = \{1, \ldots, |\ell|\}$. To obtain the confusion matrix, we divide $\mathcal{S}^\ell$ into two halves: $\mathcal{S}^\ell_{\text{train}}$ to train the classifier and $\mathcal{S}^\ell_{\text{val}}$ for validation. Again, we train the multi-class classifier using random forest classification. Let $\mathbf{A} \in \mathbb{R}^{|\ell| \times |\ell|}$ denote the confusion matrix of the classification on the validation set $\mathcal{S}^\ell_{\text{val}}$. Each element $\mathbf{A}_{ij}$ is given by:

$$\mathbf{A}_{ij} = \frac{1}{|\mathcal{S}^\ell_{\text{val},i}|} \sum_{\mathbf{x} \in \mathcal{S}^\ell_{\text{val},i}} P(j|\mathbf{x}) \tag{6}$$

where $\mathcal{S}^\ell_{\text{val},i} \subset \mathcal{S}^\ell_{\text{val}}$ are the speech signals with label i. $\mathbf{A}_{ij}$ expresses how likely a speech sample of class i is predicted to belong to class j by the classifier. Since $\mathbf{A}$ is not symmetric, we symmetrize it as

$$\bar{\mathbf{A}} = (\mathbf{A} + \mathbf{A}^T)/2. \tag{7}$$

Eventually, the optimal partitioning $\{\ell^L, \ell^R\}$ is selected to maximize:

$$E(\ell) = \sum_{i,j \in \ell^L} \bar{\mathbf{A}}_{ij} + \sum_{m,n \in \ell^R} \bar{\mathbf{A}}_{mn}. \tag{8}$$

By this, we tend to group the ambiguous speech categories into the same subset, as a result, produce two meta-classes $\{\ell^L, \ell^R\}$ that are easy to separate from each other. We apply spectral clustering [54] on the matrix $\bar{\mathbf{A}}$ to solve a relaxed version of the optimization problem in (8).

Once the optimal partition $\{\ell^L, \ell^R\}$ is determined, we learn another binary classifier $\mathcal{M}^\ell_\mathcal{S}$ using the whole set S^ℓ as training data. The samples with their labels in ℓ^L are

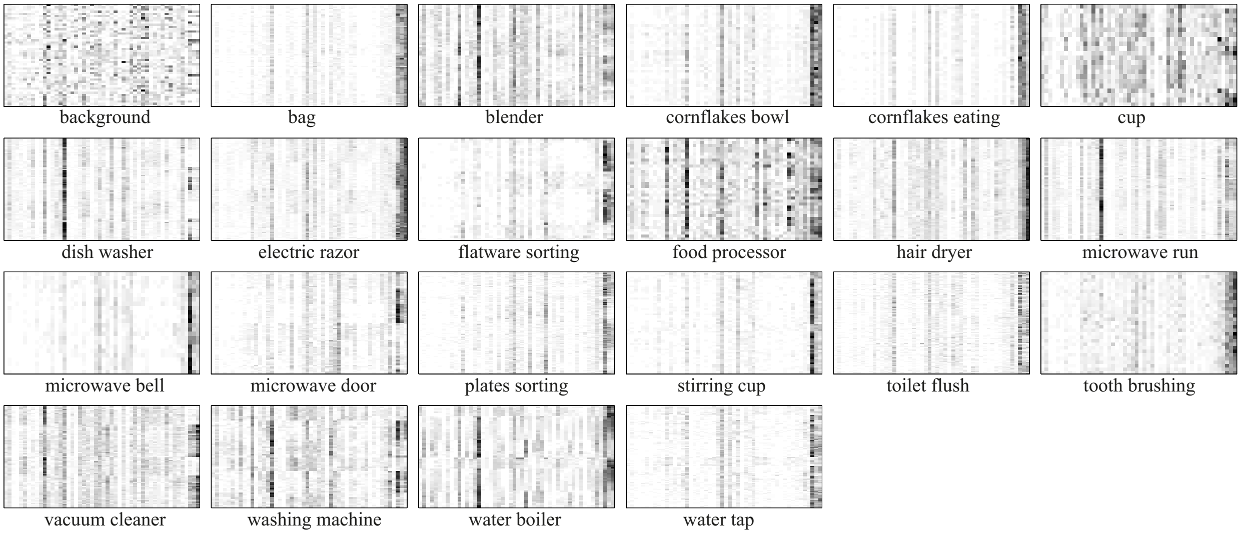

Figure 3. Similarities between audio events of the Freiburg-106 dataset and 50 phone triplet categories of the TIMIT dataset. Each row of the image represents one event instance of the corresponding class while the columns represent the indices of the speech categories.

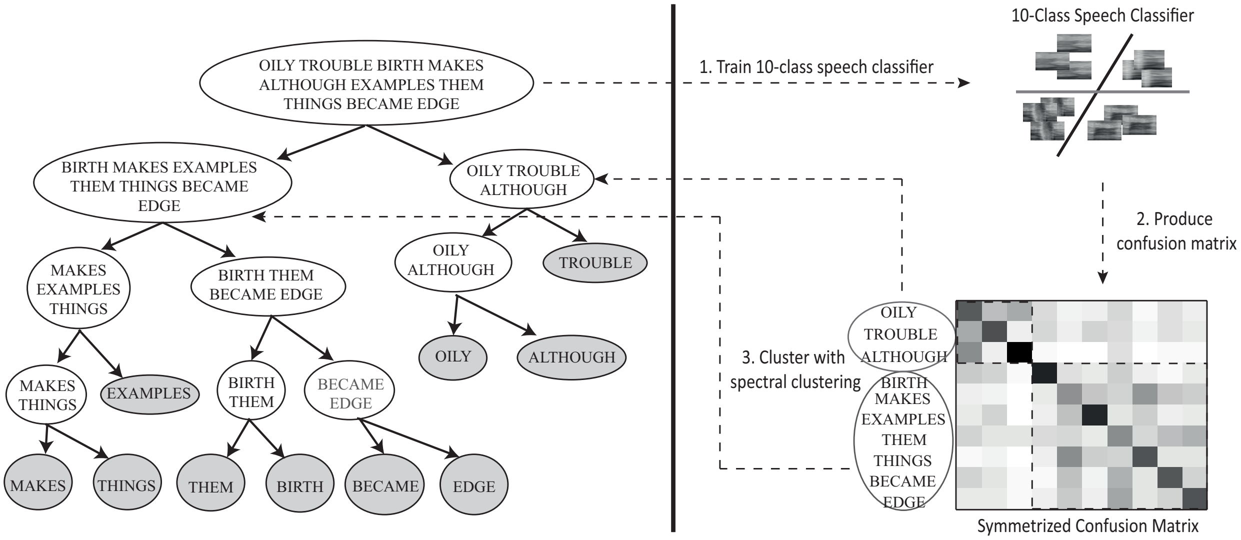
OILY TROUBLE BIRTH MAKES ALTHOUGH EXAMPLES THEM THINGS BECAME EDGE
BIRTH MAKES EXAMPLES THEM THINGS BECAME EDGE
OILY TROUBLE ALTHOUGH
MAKES EXAMPLES THINGS
BIRTH THEM BECAME EDGE
OILY ALTHOUGH
TROUBLE
OILY
ALTHOUGH
MAKES THINGS
EXAMPLES
BIRTH THEM
BECAME EDGE
MAKES
THINGS
THEM
BIRTH
BECAME
EDGE
10-Class Speech Classifier
1. Train 10-class speech classifier
2. Produce confusion matrix
3. Cluster with spectral clustering
OILY TROUBLE ALTHOUGH
BIRTH MAKES EXAMPLES THEM THINGS BECAME EDGE
Symmetrized Confusion Matrix

Figure 4. **Speech label tree construction.** On the left, the learned label tree for 10 randomly selected TIMIT word categories is shown. The white and shaded nodes represent the split and leaf nodes respectively. On the right, the splitting process at a split node (the root) is illustrated. First, the multi-class speech classifier is learned. The confusion matrix is then obtained using cross validation. Finally two clusters of speech labels are formed using spectral clustering and assigned to the child nodes. The words in the same cluster usually contains similar phones, such as the two closely related phones `l` (in "OILY" and "ALTHOUGH") and `el` (in "TROUBLE").

considered as negative examples and others with their labels in ℓ^R are considered as positive examples. The classifier $\mathcal{M}_{\mathcal{S}}^{\ell}$ is eventually associated with the node and used as a feature extractor afterwards. We recursively repeat the process until a single class label remains at a leaf node. This procedure produces totally $|\ell_{\mathcal{S}}| - 1$ binary classifiers associated with the split nodes of the tree. Evaluating them on the target audio signal $\mathbf{x}_e$ will produce a feature vector of size $2(|\ell_{\mathcal{S}}| - 1)$. It is noticed that the tree construction and evaluation can be done in parallel, therefore, it is computationally efficient.

In Figure 4, we portray a label tree constructed for ten randomly selected speech word categories of the TIMIT dataset using the proposed algorithm. For the sake of simplicity, we use speech words for demonstration. Note that, unlike WordNet [55], this tree does not need to capture any semantic information of the words.

D. Comparison with conventional speech models

Although it appears plausible to employ the conventional speech models known from the field of ASR, e.g. frame-based acoustic modeling followed by temporal sequencing by a HMM [18], for speech classification, there are various reasons for not doing so. Firstly, human speech is temporally well-structured, where it is possible to decompose it into constituent phonemes. Hence, HMMs that explicitly model temporal dependencies are able to capture the development of the speech signals very well. However, the characteristics of nonspeech audio events differ from those of speech. That is, no sub-word dictionary exists for nonspeech audio events, and compared to speech alone, they expose a much wider variety in frequency content, duration, and profile. As a result, a HMM that assumes independence between adjacent observations in time may be inefficient to capture this information. Secondly, in practice, in the audio event classification/detection challenges so far, such as CHIL CLEAR 2006/2007 [56], [57], and IEEE AASP [58], the ASR-like systems were shown inferior compared to the classifiers trained on global features extracted from the whole signals for the event classification task. These findings form a basis for our choice of the simple random-forest classifiers trained on global features of speech signals. Nevertheless, we also study the effects of preserving temporal information of the signals in the experiments in Section VI-B.

While decision-tree based clustering has been proposed in the field of ASR, such as for triphone clustering [59]–[62], our proposed clustering approach can be distinguished from them in different aspects. Firstly, compared to the rule-based clustering in [59], [62], which usually require linguistic expertise to design the splitting rules, our proposed technique is data-driven and absolutely automatic. Secondly, with the data-driven principle, the hierarchical clustering in [60]–[62] is unsupervised, and the clustering is performed on the original Euclidean feature space with the assumption that the clusters are well defined and partially separated. The idea of spectral clustering [54] used in our approach is different from them in essence. The data is first transformed into the similarity space where the clustering takes place. Furthermore, we enforce the transformation to handle nonlinearity, i.e., regardless of the geometrical shape of the clusters, by discriminatively training the nonlinear random-forest classifiers at the split nodes of the label tree and, subsequently, considering the confusion matrix of the classification as similarity measure.

IV. Selection algorithm for a sufficient subset of speech bases

Given a target dataset of $\mathcal{Y}$ categories of nonspeech audio events, among the whole world of diverse speech patterns, some of them are more relevant and contributive for representation learning than others. If we can somehow select the most relevant and contributive ones, we will gain different benefits: reduced computational cost of training classifiers, smaller dimensionality in speech-based representations for audio events, and, finally, reduced computational cost of training final event classifiers. Intuitively, a concept producing a flat proximity distribution on different event categories would not be helpful to tell apart the events and should not be included. In contrast, adding a concept that has a skewed distribution, peaking on a certain event category, would gain discrimination between the events of this class from the others. Therefore, the question arises how to select a subset of most relevant speech categories from the entire set of available ones. In this section, we propose a simple yet effective method to identify a small set of discriminative speech patterns that is sufficient to learn the representation for a target dataset at hand.

To accomplish this, we measure how close the speech category c is to a single audio event category $y \in \{1, \dots \mathcal{Y}\}$. It is similar, but in a reversed manner, to the way we measure the similarities of a nonspeech audio event to different speech categories. From the training data of $\mathcal{Y}$ target event categories, we are able to learn a multi-class event classifier $\mathcal{M}_{\mathcal{E}}$. Inputting a speech signal $\mathbf{x}_s$ of the class c into $\mathcal{M}_{\mathcal{E}}$, we obtain its closeness to the nonspeech event category y as $P(y|\mathbf{x}_s)$. The closeness κ of the speech category c and the nonspeech event category y is then computed as

$$\kappa(c, y) = \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{x}_s \in \mathcal{S}_c} P(y|\mathbf{x}_s). \tag{9}$$

Here, $\mathcal{S}_c$ denotes the set of speech signals of the speech category c. A higher closeness implies that the speech signals of the speech category c sound more similar to the events of the event category y.

After ranking the speech categories based on their closeness with respect to the event category y, we can easily obtain a subset of the speech categories that are closest to the event category y. The selection of the sufficient subset of speech patterns is as follows. The subset is firstly initialized with the closest speech categories selected for all $\mathcal{Y}$ event categories. We then repeatedly add the next M top speech categories for each nonspeech audio event category. At every step, we learn the tree-induced descriptors for the audio event signals using the current subset and evaluate the cross-validation classification accuracy. The process is continued as long as we obtain a better or equal accuracy of the cross-validation classification. Note that nothing prevents a single speech category to be selected by several audio event categories. This is expected due to sharing features between them [7].

V. Experiments

A. Final audio event classification

After obtaining the speech-based descriptors (e.g. the flat and tree-induced descriptors) for nonspeech audio events, we trained our final event classification systems using one-vs-one SVMs. Different kernels were considered, including linear, radial basis function (RBF) kernel, χ^2, and histogram intersection (hist. for short) kernels. The hyperparameters of the SVMs were tuned via 10-fold cross-validation.

The random forest classifiers used for speech classification described in Sections III and IV were trained with the algorithm in [52] with 200 trees each and ten randomly selected features for each split. We also discuss in Section VI-C how varying the number of trees will affect the learned descriptors.

B. Datasets

We extracted and used the phone triplet categories from the TIMIT dataset [14]. The corpus contains about five hours of speech with 6,300 utterances in total. Overall, 630 speakers from eight major dialect divisions of the United States spoke ten sentences. The phonetic set consists of 61 phones which is then reduced into 39 phones following the standard procedure [48]. With 39 base phonemes, there exists a vast number of triplet categories. However, in order to build a reliable speech classification model, we only kept those categories that have at least ten samples. Consequently, 2,256 of such triplet categories were retained and used as basic acoustic concepts. Lastly, for each category, we only kept at most 100 speech samples which were randomly selected. The intention of this is to make an even distribution of training data and, hence, a balanced classification problem. Larger speech corpora will allow for a larger number of speech samples.

We used audio events of three independent datasets, including UPC-TALP [45], Freiburg-106 [16], and NAR [17], as the target nonspeech signals. These datasets are recorded in different environments, and hence, differ in reverberation characteristics. The summary of the datasets is shown in Tables I, II, and III respectively.

- The UPC-TALP dataset [45] was recorded in a meeting-room environment. This dataset is multi-channel and multimodal (i.e. audio and video), and contains recording sessions of both isolated and spontaneous audio events. However, we only made use of recordings with isolated events and a single audio channel (channel 10). Correspondingly, there are 8 recording sessions where 6 different participants performed 10 times each event. Totally, there are 1,418 instances of 11 event categories. Following the setting in [63], we alternatively used 7 sessions for training and the remaining session for testing. The leave-one-session-out cross-validation accuracy is finally reported.
- The Freiburg-106 dataset [16] was collected using a consumer-level dynamic cardioid microphone in kitchen and bathroom environments. It contains 1,476 audio-based human activities of 22 categories. Particularly, several sources of ambient noise (e.g. PC fans whirring) were also presented. As in [16], we divided the dataset so that the test set contains every second recording of a category and the training set contains all the remaining recordings[1].
- The NAR dataset [17] was recorded using the frontal 300Hz - 18kHz bandpass microphone of a NAO robot with different positions in both home and office environments. The recording process also suffered from robot-head fan noise. Interestingly, this dataset includes some speech categories and they are also treated as audio events in general. Each event class has 20 event instances, except for the classes in the "Kitchen" scenario (cf. Table III), which has 21 instances. Overall, it consists of 852 sound signals of 42 classes, both speech and nonspeech, with different temporal and spectral characteristics. As in [17], we randomly divided the dataset into 10 parts and conducted 10-fold cross-validation. The cross-validation performance accuracy is then reported.

C. Baseline systems

We implemented the following baseline systems for comparison:

- Bag-of-words (BoW) system: The implemented BoW model has been widely used for audio classification

Table I
SUMMARY OF THE UPC-TALP DATASET [45].

Event Type	# event instance								
	S1	S2	S3	S4	S5	S6	S7	S8	Total
knock (door, table)	9	8	10	10	10	8	11	13	79
door slam	17	15	19	20	40	37	56	52	256
steps	10	10	8	23	43	34	28	50	206
chair moving	19	37	32	22	23	38	34	40	245
spoon (cup jingle)	10	11	13	11	10	15	11	15	96
paper work	9	11	10	8	17	12	12	12	91
key jingle	11	11	11	8	0	13	10	18	82
keyboard typing	10	10	13	12	10	13	10	11	89
phone ringing	11	18	11	14	8	11	13	15	101
applause	9	5	9	11	12	9	14	14	83
cough	10	10	12	13	9	13	11	12	90
Total	125	146	148	152	182	203	210	252	1,418

Table II
SUMMARY OF FREIBURG-106 DATASET [16].

Event Type	# event instance	Event Type	# event instance
background	47	microwave	92
food bag opening	80	microwave bell	24
blender	60	microwave door	86
cornflakes bowl	36	plates sorting	135
cornflakes eating	43	stirring cup	59
pouring cup	22	toilet flush	124
dish washer	89	tooth brushing	29
electric razor	83	vacuum cleaner	79
flatware sorting	40	washing machine	67
food processor	35	water boiler	65
hair dryer	66	water tap	115
Total	1,476		

Table III
SUMMARY OF NAR DATASET [17].

Scenarios	Taxonomy	Event Type
Kitchen	*"Mouth" sound*	eating, choking
	Cooking	cuttlery, fill a glass, running the tap
	Moving	open/close a drawer, move a chair, open microwave, close microwave
	Alarms	microwave, fridge, toaster
Office	*Door*	close, open, key, knock
	Others	ripped paper, zip, (another) zip
Nonverbal		fingerclap, handclap, tongue clic
Speech	*Numbers*	one, two, three, four, five, six, seven, eight, nine, ten
	Orders	hello, left, right, turn, move, stop, Nao, yes, no, what

[1] This is based on unofficial communication with the authors of [16].

recently [11], [12], [26], [39], [64]. Using this model, an audio event is represented by a histogram of codebook entries.

- Pyramid bag-of-words (pBoW) system: We extracted BoW descriptors on different pyramid levels [65] to encode the temporal structure of audio events. This approach has recently achieved state-of-the-art results on different benchmark datasets [12].

For all baselines, we used k-means for codebook learning. The entries were obtained as the cluster centroids, and codebook matching was based on Euclidean distance. In fact, the performance of these systems highly depends on the codebook size and the pyramid level. Although an appropriate setting can be sought for each dataset using cross-validation, we commonly perform analysis with different settings for all datasets. We used different codebook sizes of $\{50, 100, 150, 200, 250\}$. In particular, we opted $\{2, 3, 4\}$ pyramid levels for the pBoW systems. For convenience, let us denote a level-n pBoW system as pBoW-n where $n \in \{2, 3, 4\}$. The final classification systems were implemented using one-vs-one SVMs with four kernels, including linear, RBF, χ^2, and hist. Again, the hyperparameters of the SVMs were also tuned via 10-fold cross-validation.

D. Experimental results

1) Flat descriptors vs. tree-induced descriptors: The performance of the flat and tree-induced descriptors on audio event classification are shown in Figures 5, 6, and 7 for three target datasets (those without temporal information) respectively. From the whole set of 2,256 phone triplet categories, we randomly selected $\{50, 100, \ldots, 1000\}$ categories and evaluated them. Note that the flat and tree-induced descriptors were always built upon the same subsets of phone triplets. It is also worth emphasizing again that at each time we learned the speech-based feature extractor and commonly applied it to the three different datasets. We repeated the experiments ten times and report the mean and standard deviation of the classification accuracies. Obviously, with the same speech patterns, the tree-induced descriptors consistently perform much better than the flat counterparts. Specifically, the average accuracy gains are summarized in Table IV for different kernels and datasets.

Table IV
AVERAGE ABSOLUTE ACCURACY GAIN (%) OF THE TREE-INDUCED DESCRIPTORS COMPARED TO THE FLAT DESCRIPTORS.

	Linear	**RBF**	χ^2	**Hist.**
UPC-TALP	6.50	5.60	4.95	5.07
Freiburg-106	7.99	7.88	5.93	6.87
NAR	7.19	7.49	5.57	6.00

It can be seen that the performance curves appear to saturate at some points after which adding more speech patterns results in little improvement. In addition, more often than not, random selection of a set of speech bases yields a reasonable performance provided that the number of speech categories is large enough, i.e. after the saturation points. Last but not least, the performance of the linear classifiers is comparable with the other nonlinear classifiers while linear classifiers are computationally much cheaper to train and evaluate.

The rational behind the performance improvement of tree-induced descriptor against the flat ones is that when we decomposing the original complex speech classification problem into simple binary classification problems, we are able to classify the speech patterns more correctly. As a results, the similarities between a target nonspeech signal and the speech patterns can be more accurately measured. All of this leads to better representations with the tree-induced descriptors compared to the flat descriptors. We further discuss about the importance of the underlying speech classifiers in Section VI-C.

2) Sufficient subset of speech patterns vs. the whole set: As mentioned above, when the number of speech patterns reaches some certain saturation points, adding more categories only leads to marginal improvements. It is preferable to somehow obtain a small subset of speech patterns that produces a low-dimensional feature space without losing the accuracy too much. In this experiment, we used the algorithm described in Section IV to select such a sufficient speech subset. At every step of the algorithm, we collectively add the next $M = 5$ top speech patterns for each event class into the current subset. Note that this algorithm is deterministic, therefore, the resulting subset is fixed rather than random. Finally, these subsets are specific for different datasets and kernels.

The performance of these sufficient subsets is indicated by the red star in Figures 5, 6, and 7. It can be seen that in most of the cases their performances are above the performance curves of the random settings. Furthermore, their positions are likely in the saturation regions of the performance curves.

In order to show that the sufficient subsets are actually representative for the entire set of all speech patterns (i.e. 2,256 phone triplet categories), we compare their representation capabilities. The representation capability is defined as the accuracy of the classification task. In Figure 8, we show the sizes and the accuracies achieved by the efficient subsets against those obtainable with the entire set. As can be seen, the differences in accuracy are negligible whereas the sizes of the efficient subsets are significantly smaller compared to the entire set.

3) Using the proposed descriptors as additional features: In this experiment, we investigate how the proposed speech-based descriptors improve the final event classification with some fusion schemes when we consider them as additional features. We employed the descriptors induced by the sufficient subsets with respect to the χ^2 kernel to integrate with the descriptors obtained by the baseline systems: BoW, pBoW-2, pBoW-3, and pBoW-4. We then analyzed results with different codebook sizes of the baseline systems.

Different descriptors (i.e. the baseline descriptors and the proposed ones) are combined in a multi-channel approach [66]:

$$K(e_i, e_j) = \exp\big(-\sum_k \frac{1}{\bar{D}^k} D(e_i^k, e_j^k)\big) \tag{10}$$

where $D(e_i^k, e_j^k)$ is the χ^2 distance between the audio events

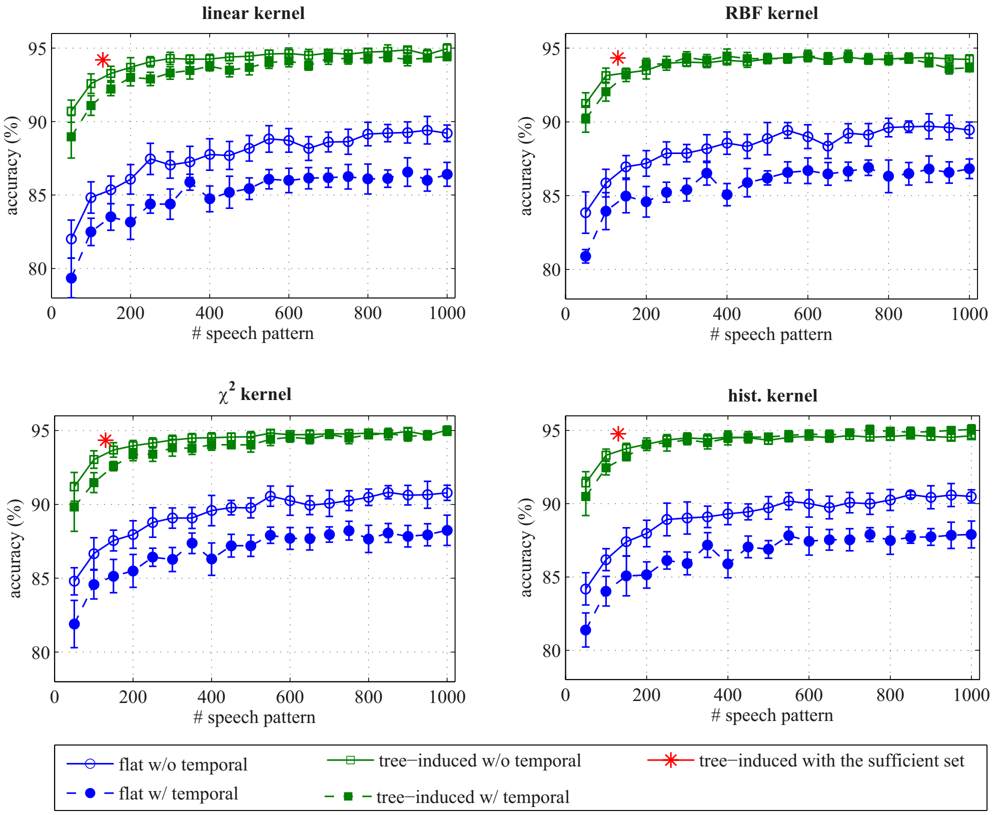

Figure 5. **UPC-TALP dataset.** Performance of the flat and tree-induced descriptors on audio event classification with different kernels.

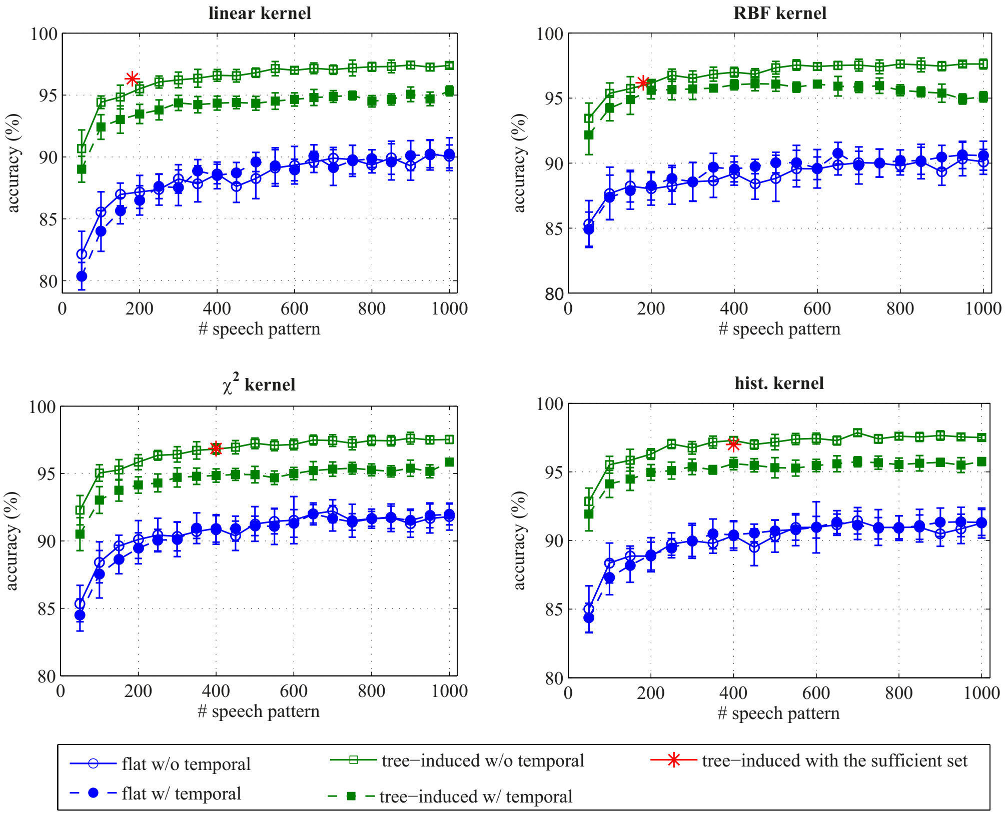

Figure 6. **Freiburg-106 dataset.** Performance of the flat and tree-induced descriptors on audio event classification with different kernels.

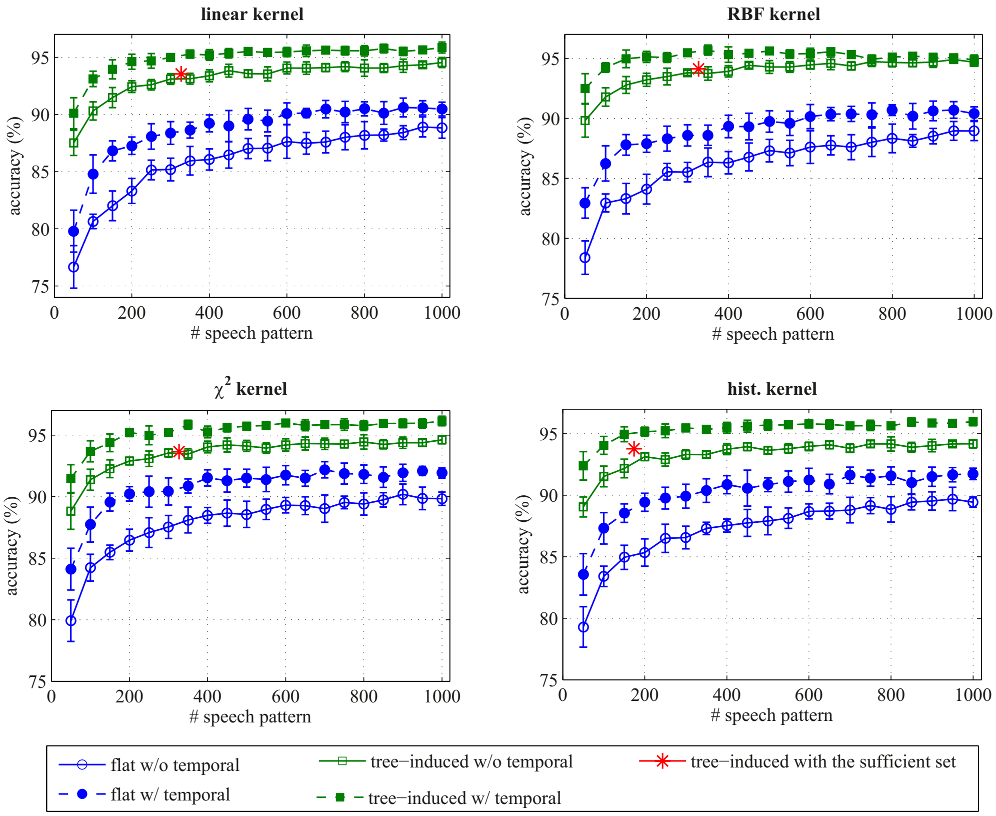

Figure 7. **NAR dataset.** Performance of the flat and tree-induced descriptors on audio event classification with different kernels.

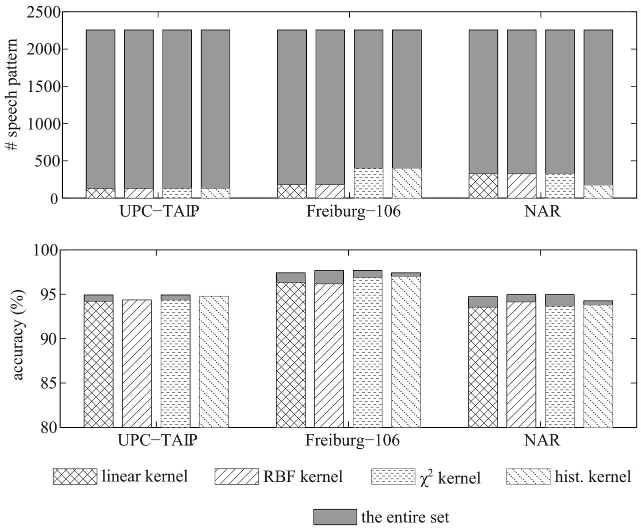

Figure 8. **Sufficient subsets vs. the whole set.** Sizes and representation capability of the efficient subsets compared to those of the entire set of all speech patterns.

Table V
AVERAGE ABSOLUTE ACCURACY GAIN (%) OF THE FUSION SYSTEMS COMPARED TO THE BASELINE SYSTEMS.

	BoW	**pBoW-2**	**pBoW-3**	**pBoW-4**
UPC-TALP	0.58	0.34	0.30	0.23
Freiburg-106	2.51	2.54	2.54	2.57
NAR	3.76	2.68	2.35	2.46

e_i and e_j with respect to the k-th channel. $\bar{D}^k$ is the mean χ^2 distance of the training samples for the k-th channel. For classification, we used nonlinear SVMs with the RBF-χ^2 kernel [67].

The fusion results are shown in Figures 9, 10, and 11 for the UPC-TALP, Freiburg-106, and NAR datasets, respectively. The results make clear that by augmenting the baseline systems with the proposed descriptors, we consistently boost their performance to higher levels. We averaged the accuracy gains over different codebook sizes and summarized them in Table V. While the used fusion scheme is very simple, other better alternatives can also be used, such as multiple kernel learning framework [68], [69].

4) Performance comparison: We provide in this section an overall picture of the performance of different systems: our proposed systems, the baseline systems, our proposed fusion systems, and the state-of-the-art systems. The performance of our systems are reported using those obtained by the sufficient subsets with different kernels. For the baselines BoW, pBoW-2, pBoW-3, and pBoW-4, we used their best performance amongst different codebook sizes and kernels for comparison. The fusion systems were implemented by integrating our proposed descriptors with the corresponding best baseline systems. Finally, the state-of-the-art performance of UPC-TALP, Freiburg-106, and NAR datasets were reported

Table VI
PERFORMANCE COMPARISON (%) BETWEEN DIFFERENT SYSTEMS: OUR PROPOSED SYSTEMS, THE BASELINE SYSTEMS, OUR FUSION SYSTEMS, AND THE STATE-OF-THE-ART SYSTEMS. WE MARK IN BOLD WHERE OUR SYSTEMS OUTPERFORM ALL THE COMPETITORS (I.E. THE BASELINES AND THE STATE-OF-THE-ART SYSTEMS).

	Our system				**Baseline system**				**Our fusion system**				**State-of-the-art**
	Linear	**RBF**	χ^2	**Hist.**	**BoW**	**pBoW-2**	**pBoW-3**	**pBoW-4**	**BoW+**	**pBoW-2+**	**pBoW-3+**	**pBoW-4+**	
UPC-TALP	94.20	94.34	94.34	94.77	96.32	96.11	96.11	95.83	**96.46**	96.11	96.25	95.76	87.60
Freiburg-106	95.69	95.46	96.28	**96.77**	96.64	96.43	96.17	95.87	**97.77**	**97.77**	**97.39**	**97.25**	92.40
NAR	93.54	94.13	93.66	93.78	94.83	96.13	96.48	96.01	**97.08**	**98.36**	**97.89**	**97.89**	97.00

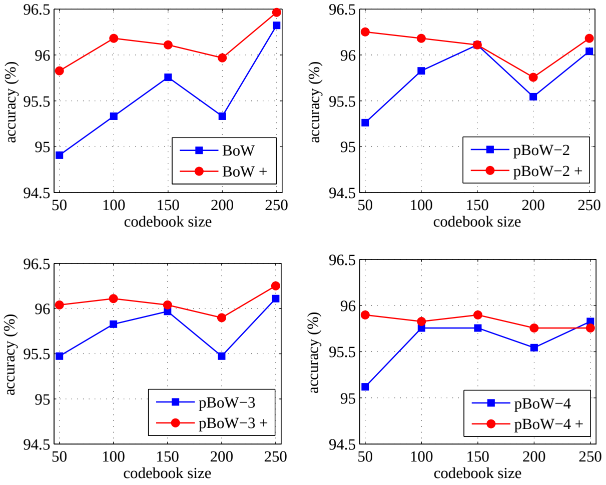

Figure 9. **UPC-TALP fusion systems.** Performance of the fusion systems compared to the baseline systems. Note that the fusion systems are denoted with the additional '+' symbol.

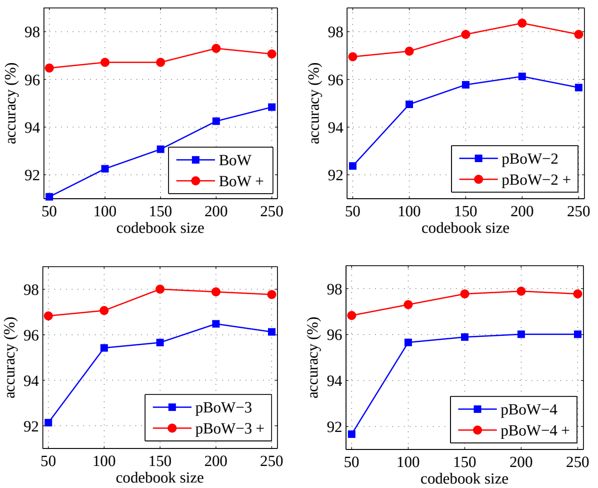

Figure 11. **NAR fusion systems.** Performance of the fusion systems compared to the baseline systems. Note that the fusion systems are denoted with the additional '+' symbol.

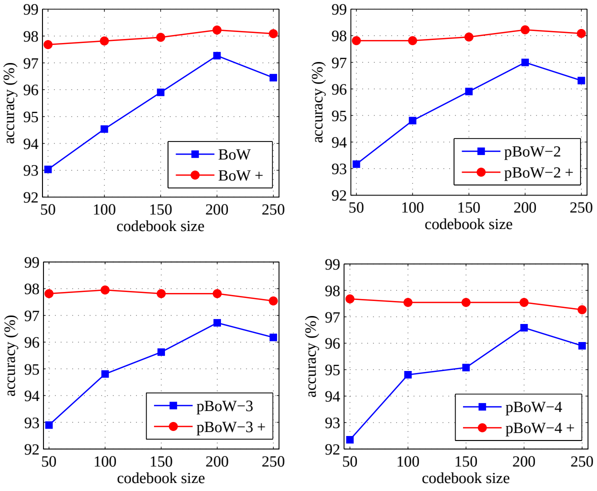

Figure 10. **Freiburg-106 fusion systems.** Performance of the fusion systems compared to the baseline systems. Note that the fusion systems are denoted with the additional '+' symbol.

in the works of Nadeu et al. [63], Stork et al. [16], and Maxime et al. [17] respectively. The performance comparison is shown in Table VI. Note that, to agree with the results in [16], the performances on the Freiburg-106 dataset were reported in terms of f-score instead of accuracy.

It can be seen that the performance of our proposed descriptors with linear kernels significantly outperform the state-of-the-art results on two datasets, UPC-TALP and Freiburg-106, while they are just marginally lower than those of the baselines given by BoW and pBoW. These baselines are actually very strong models as they are recently reported as state-of-the-art results on benchmark datasets [12]. We would also like to point out that our linear systems are computationally much cheaper to train and evaluate compared to nonlinear baseline models. These results are impressive given the fact that, with the speech-based descriptors, we have not incorporated the features from the audio events themselves in the models. When taking into account these features, the fusion systems not only improve the performance of the baseline systems to higher levels but also set the state-of-the-art performance on all datasets. Specifically, the performance improvements obtained by the best fusion systems compared to the state-of-the-art systems are 8.86%, 5.37%, and 1.36% for UPC-TALP, Freiburg-106, and NAR datasets, respectively.

VI. DISCUSSION

A. Phone triplets vs. words

In our previous work [46], we showed that speech signals at word levels can also be used as speech patterns. We analyze in this section the difference between using words and phone triplets.

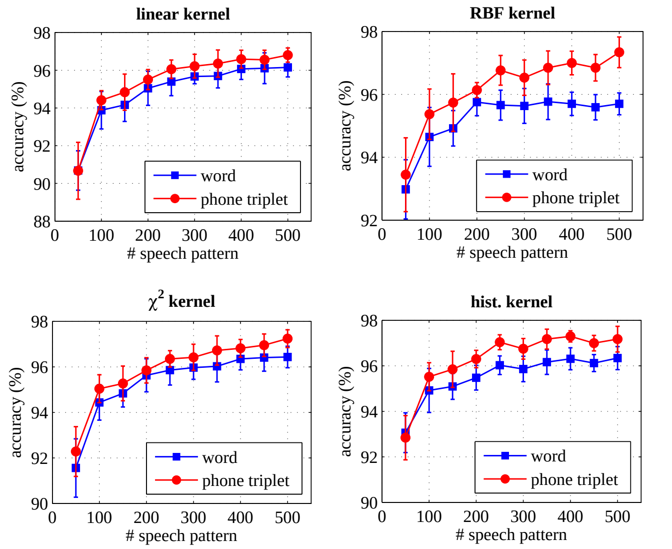

Figure 12. **Words vs. phone triplets.** The final event classification accuracies of the tree-induced descriptors on the Freiburg-106 dataset when words and phone triplets are used for speech patterns.

We extracted speech words from the TIMIT database. There are totally about 500 such word categories with at least ten samples per class. This number is much smaller than the number of phone triplets since speech words in general are high-order combinations of phones (the order is more than three in most of the cases) which require more data to cover. Therefore, phone triplets offer us more speech patterns for analysis. For comparison of the representation power, we repeated the experiments in [46] ten times on the Freiburg-106 dataset and show in Figure 12 the final classification accuracies of the tree-induced descriptors when using words and phone triplets as speech patterns, respectively. As can be seen, with the same number of speech patterns, the classification accuracies obtained with phone triplets are consistently better than those obtained with words. Specifically, the absolute average accuracy improvements are 0.51%, 0.97%, 0.54%, and 0.76% with linear, RBF, χ^2, and hist. kernels, respectively. A possible explanation is that the words are usually much longer than the audio events and that they are special combination of phones. As a result, the audio events are often better matched with phone triplets than with words.

B. Retaining temporal information of the signals

In the field of ASR, it is well known that the temporal dynamic is useful for speech modeling. Although we argue in Section III-D that speech and nonspeech signals are very different in temporal characteristics, the question arises of what happens when we incorporate the temporal information of the signals.

In order to retain a certain degree of the temporal information, a phone-triplet segment is divided into three constituent phonemes. Each phoneme is then decomposed into frames and described by a 53-dimensional feature vector which is the mean of frame-wise features. Three feature vectors of the three individual phonemes are finally concatenated to make a 159-dimensional feature vector for the phone triplet. Note that the order of the constituent phonemes does matter here to categorize the phone triplets, therefore, the phone triplet categories in this case are different from the previous experiments. For the nonspeech signals, as there exists no such phone components in the same way as for speech, we simply divide each of them into three equal-length segments. The feature-extraction step is similar to that for speech.

Table VII
FLAT DESCRIPTORS: AVERAGE ABSOLUTE ACCURACY GAIN (%) WHEN THE TEMPORAL INFORMATION OF THE SIGNALS IS PRESERVED.

		Linear	**RBF**	χ^2	**Hist.**
UPC-TALP		-2.63	-2.59	-2.52	-2.60
Freiburg-106		-0.10	0.36	-0.21	0.04
NAR	Overall	2.77	2.73	2.81	2.90
	Speech	8.31	8.15	7.50	7.81
	Nonspeech	-2.13	-2.07	-1.45	-1.44

Table VIII
TREE-INDUCED DESCRIPTORS: AVERAGE ABSOLUTE ACCURACY GAIN (%) WHEN THE TEMPORAL INFORMATION OF THE SIGNALS IS PRESERVED.

		Linear	**RBF**	χ^2	**Hist.**
UPC-TALP		-0.74	-0.12	-0.49	0.01
Freiburg-106		-2.24	-1.45	-2.02	-1.74
NAR	Overall	1.81	1.20	2.81	2.01
	Speech	4.96	4.07	4.83	5.41
	Nonspeech	-0.98	-1.34	-0.93	-1.00

We additionally show in Figures 5, 6, and 7 the performance curves when the temporal information is retained in order to compare with those without the temporal information. As can be seen, the temporal information does not bring up a big advantage. It even worsens the results for the UPC-TALP and Freiburg106 datasets. The NAR dataset is an exception due to the fact that it consists of 20 speech categories out of 42 classes. It turns out that retaining the temporal information unsurprisingly benefits these speech categories but degenerates the nonspeech categories. It is further clarified in Table VII and VIII for the flat and tree-induced descriptors, respectively, where we summarize the average performance gain when temporal information is preserved.

Concretely, integrating the temporal dynamics of the signals does not invigorate the final representations for nonspeech signals, at least by the way presented above.

C. The importance of the underlying speech classifiers

We study in this section how the quality of the random-forest speech classifiers affects the speech-based descriptors, and hence, the performance of the final nonspeech audio event classification. To accomplish this, we varied the number of trees in the random forest classifiers in the range $\{25, 50, \ldots, 200\}$ and recorded their out-of-bag (OOB) errors

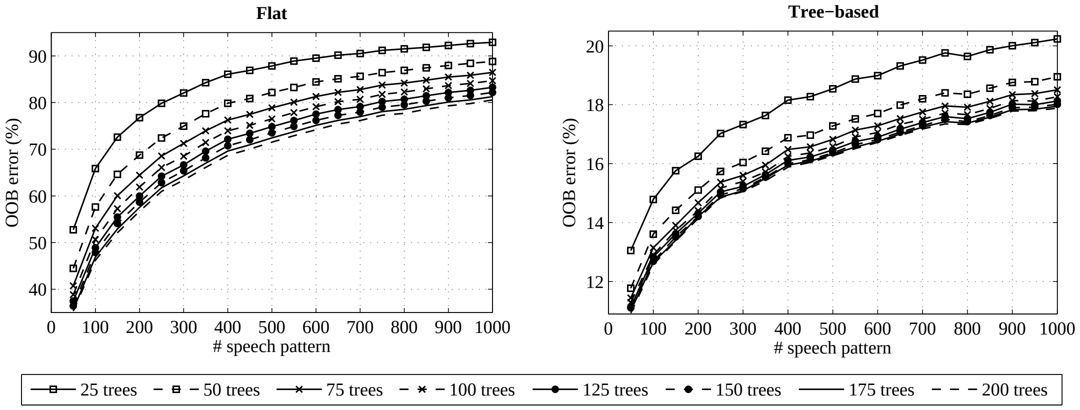

Figure 13. **OOB error.** The average OOB errors of random forest speech classifiers with different number of trees. Note that the scales are different.

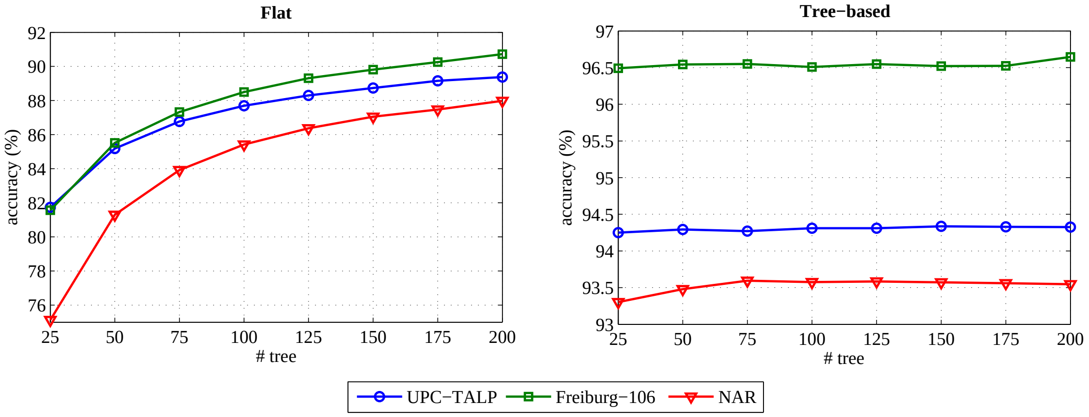

Figure 14. **Event classification performance.** Average event classification performance as functions of the number of trees in the random forest speech classifiers.

[52] which are estimated internally during the forest construction. The random forest classification is very robust against overfitting [52] and, in general, it is expected to bring a performance gain when increasing the number of trees at the cost of increasing computation.

We show in Figure 13 the OOB errors of the speech classifiers for both flat and tree-based cases. For the tree-based cases, we report the average of the OOB errors of the binary classifiers at the split nodes of the label trees. As can be seen, the OOB error curves show similar patterns. Firstly, the error curves escalate, as expected, as the speech classification problem becomes more complex with the increasing number of speech categories. Secondly, the entire error curves expose a trend of moving down, i.e. better performance, when we increase the number of trees in the forests. However, the improvement becomes incrementally slower with the increasing number of the trees. When the number of trees is large enough, e.g. at 150 trees, adding more trees leads to insignificant performance gain. Note that the error scales in the tree-based cases are significantly smaller than those of the flat cases. It is due to the fact that we have transferred a complex flat multi-class classification problem into multiple simpler binary classification problems in the tree-based cases. This reflects the differences in the final event classification performance between the flat and tree-based cases as discussed below.

The random-forest speech classifiers with different number of trees were then used to extract descriptors for the nonspeech audio events. The final event classification performance is shown in Figure 14 as a function of the number of trees. Note that, at each number of trees, we computed and averaged the performance over different number of speech categories in $\{50, 100, \ldots, 1000\}$. Two different patterns are shown for the flat and tree-based cases over all datasets. For flat cases, increasing the number of trees leads to a performance gain although the improvement is gradually diminishing as expected. In contrast, for the tree-based cases, the performance curves are almost flat, indicating very small difference in classification accuracy. This result implies that the number of trees of the speech classifiers is more important for the flat descriptors than than for the tree-induced descriptors. This is reasonable since the complex multi-class classification problems in the flat cases needs strong classifiers, i.e. large number of trees, while the simple binary classification problems in the tree-based cases can be easily coped with by simpler classifiers.

D. Future work

More than 6,900 languages exist in the world [70] and many annotated corpora are available such as TIMIT [14], SWITCHBOARD [71], Wall Street Journal [72], and GlobalPhone [73] to mention a few. This opens up enormous opportunities to explore for learning representations from speech. Using different levels and different languages would result in different representations. Their combinations would offer even more opportunities.

It can be seen from Figures 5, 6, and 7 that the number of speech categories needs to be sufficiently large to guarantee a good performance. This is understandable since with more speech categories, we are likely to cover more acoustic concepts. Interestingly, this observation is generic, that is, it is achievable for different datasets. However, just increasing the number of categories does not guarantee a better performance. The reason is quite obvious. For example, when the categories are randomly selected, many similar categories are likely to exist. This results in correlation in some dimensions of the induced feature space which worsens the model. As shown, organizing the categories in a tree structure is efficient to alleviate this problem. However, it is worth further studying how to deal with it.

Last but not least, our proposed method is not limited to audio event representations. Since the learned speech-based feature extractors are generic, they can be applied to any other variants of audio signals such as music and even speech. At least, the induced descriptors can act as additional sources to improve performance of existing systems.

VII. Conclusions

We presented in this paper an approach for feature learning that uses speech phone triplets as acoustic concepts to represent a target nonspeech audio signal. The representation is produced by measuring the similarities of the target signal to different speech patterns via a speech classifier. We further propose to learn to organize the speech patterns within a label tree and subsequently achieve a better representation. These descriptors are generic. Once the feature extractor has been learned, it can be used to extract features for different datasets. While the entire set of available speech categories may be redundant, we proposed an algorithm to extract a sufficient subset. This subset can approximate the entire set in terms of representation capability while its size is much smaller. In the experiments, we employed phone triplets from the TIMIT dataset as speech patterns to learn representations for audio events of different datasets, including UPC-TALP, Freiburg-106, and NAR. Our experimental results on the audio event classification task show that the proposed descriptors are efficient even with a simple linear classification model. Furthermore, using our proposed descriptors as additional features can help to significantly boost performance of an existing system. We showed that we are able to obtain state-of-the-art on all three audio event datasets with a simple fusion scheme.

Acknowledgment

This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1].

We would also like to thank Johannes A. Stork for providing the Freiburg-106 dataset.

References

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[2] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.

[3] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1994.

[4] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[5] R. F. Lyon, "Machine hearing: An emerging field," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 131–139, 2010.

[6] D. Barchiesi, D. Giannoulis, D. Stowell, and M. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[7] H. Phan and A. Mertins, "Exploring superframe co-occurrence for acoustic event recognition," in *Proc. EUSIPCO*, 2014, pp. 631–635.

[8] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.

[9] J. Dennis, H. D. Tran, and E. S. Chng, "Image feature representation of the subband power distribution for robust sound event classification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 367–377, 2013.

[10] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. NIPS*, 2009, pp. 1096–1104.

[11] S. Pancoast and M. Akbacak, "Softening quantization in bag-of-audio-words," in *Proc. ICASSP*, 2014, pp. 1370–1374.

[12] A. Plinge, R. Grzeszick, and G. Fink, "A bag-of-features approach to acoustic event detection," in *Proc. ICASSP*, 2014, pp. 3704–3708.

[13] E. Humphrey, J. Bello, and Y. Lecun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," in *Proc. ISMIR*, 2012, pp. 403–408.

[14] W. Fisher, G. Doddington, and K. Goudie-Marshall, "The DARPA speech recognition research database: Specifications and status," in *Proc. DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

[15] T. Butko, "Feature selection for multimodal acoustic event detection," Ph.D. dissertation, Universitat Politecnica de Catalunya, 2011.

[16] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using non-Markovian ensemble voting," in *Proc. IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2012, pp. 509–514.

[17] J. Maxime, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound representation and classification benchmark for domestic robots," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[18] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[19] C. Nadeu, D. Macho, and J. Hernando, "Frequency and time filtering of filter-bank energies for robust hmm speech recognition," *Speech Communication*, vol. 34, pp. 93–114, 2001.

[20] S. Chu, S. Narayanan, and C.-C. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[21] J. Dennis, H. D. Tran, and H. Li, "Spectrogram image feature for sound event classification in mismatched conditions," *Signal Processing Letters*, vol. 18, no. 2, pp. 130–13, 2011.

[22] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Proc. WASPAA*, 2011, pp. 69–72.

[23] C. Lin, S. Chen, T. Truong, and Y. Chang, "Audio classification and categorization based on wavelets and support vector machine," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 644–651, 2005.

[24] T. C. Walters, "Auditory-based processing of communication sounds," Ph.D. dissertation, University of Cambridge, 2011,

[25] R. F. Lyon, M. Rehn, S. Bengio, T. C. Walters, and G. Chechik, "Sound retrieval and ranking using sparse auditory representations," *Neural Computation*, vol. 22, no. 9, pp. 2390–2416, 2010.

[26] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, pp. 881–891, 2007.

[27] P.-S. Huang, J. Yang, M. Hasegawa-Johnson, F. Liang, and T. S. Huang, "Pooling robust shift-invariant sparse representations of acoustic signals," in *Proc. Interspeech*, 2012.

[28] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Sparse representation based on a bag of spectral exemplars for acoustic event detection," in *Proc. ICASSP*, 2014, pp. 6255–6259.

[29] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, M. Lagrange, and M. D. Plumbley, "A database and challenge for acoustic scene classification and event detection," in *EUSIPCO*, 2013, pp. 1–5.

[30] J. F. Gemmeke, L. Vuegen, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach for audio event detection," in *Proc. WASPAA*, 2013, pp. 1–4.

[31] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio event detection from acoustic unit occurrence patterns," in *Proc. ICASSP*, 2012, pp. 489–492.

[32] M. L. Chin and J. J. Burred, "Audio event detection based on layered symbolic sequence representations," in *Proc. ICASSP*, 2012, pp. 1520–6149.

[33] S. Chaudhuri and B. Raj, "Unsupervised structure discovery for semantic analysis of audio," in *Proc. NIPS*, 2012, pp. 1178–1186.

[34] J. Dennis, Y. Qiang, T. Huajin, T. H. Dat, and L. Haizhou, "Temporal coding of local spectrogram features for robust sound recognition," in *Proc. ICASSP*, 2013, pp. 803–807.

[35] G. Hinton, L. Deng, D. Yu, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, G. Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[36] X.-L. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, 2013.

[37] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[38] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *Proc. EUSIPCO*, 2014, pp. 506–510.

[39] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Proc. Interspeech*, 2013.

[40] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[41] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, vol. 3, 2000, pp. 1635–1638.

[42] H. Ketabdar and H. Bourlard, "Enhanced phone posteriors for improving speech recognition systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1094–1106, 2010.

[43] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[44] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos, "Multimicrophone fusion for detection of speech and acoustic events in smart spaces," in *Proc. EUSIPCO*, 2014, pp. 2375–2379.

[45] T. Butko, C. Canton-Ferrer, C. Segura, X. Giró, C. Nadeu, J. Hernando, and J. R. Casas, "Acoustic event detection based on feature-level fusion of audio and video modalities," *EURASIP Journal on Advances in Signal Processing*, 2011.

[46] H. Phan, L. Hertel, M. Maass, R. Mazur, and A. Mertins, "Representing nonspeech audio signals through speech classification models," in *Proc. Interspeech*, 2015.

[47] F. Müller and A. Mertins, "Contextual invariant-integration features for improved speaker-independent speech recognition," *Speech Communication*, vol. 53, no. 6, pp. 830–841, 2011.

[48] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[49] M. Gales and S. Young, *The application of hidden Markov models in speech recognition*. Now Publishers Inc., Hanover, USA, 2008, vol. 1.

[50] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.4.1)," Cambridge University Engineering Department, Cambridge, UK, Tech. Rep., 2009.

[51] B. Ghoraani and S. Krishnan, "Time-frequency matrix feature extraction and classification of environmental audio signals," *IEEE Tran. on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2197–2209, 2011.

[52] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[53] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *Proc. NIPS*, 2010, pp. 163–171.

[54] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, 2001, pp. 849–856.

[55] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, and K. Miller, "Introduction to wordnet: An online lexical database," *International Journal of Lexicograph*, vol. 3, no. 4, pp. 235–244, 1990.

[56] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," *Lecture Notes in Computer Science*, vol. 4122, pp. 311–322, 2007.

[57] A. Temko, C. Nadeu, D. Macho, R. Malkin, C. Zieger, and M. Omologo, *Computers in the Human Interaction Loop*. Springer London, 2009, ch. Acoustic Event Detection and Classification, pp. 61–73.

[58] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: an IEEE AASP challenge," in *Proc. WASPAA*, 2013, pp. 1–4.

[59] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. Workshop on Human Language Technology*, 1994, pp. 307–312.

[60] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," in *Proc. ICSLP*, 1996, pp. 2005–2008.

[61] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Computer Speech & Language*, vol. 8, no. 4, pp. 369–383, 1994.

[62] K.-F. Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Swartz, and R. Weide, "Allophone clustering for continuous speech recognition," in *Proc. ICASSP*, vol. 2, 1990, pp. 749–752.

[63] C. Nadeu, R. Chakraborty, and M. Wolf, "Model-based processing for acoustic scene analysis," in *Proc. EUSIPCO*, 2014, pp. 2370–2374.

[64] V. Carletti, P. Foggia, G. Percannella, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance using a bag of aural words classifier," in *Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2013, pp. 81–86.

[65] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, vol. 2, 2006, pp. 2169–2178.

[66] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[67] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc CVPR*, 2008, pp. 1–8.

[68] M. Gnen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[69] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning nonlinear combinations of kernels," in *Proc. NIPS*, 2009, pp. 396–404.

[70] R. Gordon, Ed., *Ethnologue: Languages of the World*. Dallas: SIL International, 2005.

[71] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proc. ICASSP*, vol. 1, 1992, pp. 517–520.

[72] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. the workshop on Speech and Natural Language*, 1991, pp. 357–362.

[73] T. Schultz, N. T. Vu, and T. Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *Proc. ICASSP*, 2013, pp. 8126–8130.

Huy Phan (S'13) received the B.Sc. degree in computer science from the University of Science, Ho Chi Minh City, Vietnam, in 2007 and the M.Eng. in computer engineering from the Nanyang Technological University, Singapore, in 2012. From 2012 to 2013, he was with the University of Information Technology, Ho Chi Minh City, Vietnam as a Lecturer. He is currently a Ph.D. student at the Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Lübeck, Germany and a research associate at the Institute for Signal Processing, University of Lübeck. His research interests include audio/acoustic signal processing, pattern recognition, and machine learning, with a special focus on acoustic/audio event detection.

Lars Hertel received his B.Sc. and M.Sc. degrees with honors in Computer Science from the University of Lübeck, Germany, in 2012 and 2014, respectively. He is currently a second year Ph.D. student and a research associate at the Institute of Signal Processing, University of Lübeck, Germany. His research interests include deep learning for machine hearing and computer vision, with a special focus on acoustic event detection.

Marco Maass (S'13) received the B.Sc. and M.Sc. degrees in computer science from the University of Lübeck, Lübeck, Germany, in 2010 and 2012, respectively. He is currently pursuing the Ph.D. degree at the Graduate School for Computing in Medicine and Life Sciences, University of Lübeck and is a research associate at the Institute for Signal Processing, University of Lübeck. His research interests include machine learning, filter design, and image processing, with a special focus filter bank design, MRI reconstruction, and MPI reconstruction.

Radoslaw Mazur (S'09–M'11) was born in Wroclaw, Poland, in 1976. He received the Diplominformatiker degree from the University of Oldenburg, Oldenburg, Germany, in 2004 and the Dr.-Ing. degree in computer science from the University of Lübeck, Lübeck, Germany, in 2010. He was an Assistant Researcher in the Department of Physics, University of Oldenburg, from 2004 to 2006, and then joined the University of Lübeck. The current research interests are digital signal and audio processing, with a special focus on blind source separation.

Alfred Mertins (M'96–SM'03) received his Dipl.-Ing. degree from the University of Paderborn, Germany, in 1984, the Dr.-Ing. degree in Electrical Engineering and the Dr.-Ing. habil. degree in Telecommunications from the Hamburg University of Technology, Germany, in 1991 and 1994, respectively. From 1986 to 1991 he was a Research Assistant at the Hamburg University of Technology, Germany, and from 1991 to 1995 he was a Senior Scientist at the Microelectronics Applications Center Hamburg, Germany. From 1996 to 1997 he was with the University of Kiel, Germany, and from 1997 to 1998 with the University of Western Australia. In 1998, he joined the University of Wollongong, where he was at last an Associate Professor of Electrical Engineering. From 2003 to 2006, he was a Professor in the Faculty of Mathematics and Science at the University of Oldenburg, Germany. In November 2006, he joined the University of Lübeck, Germany, where he is a Professor and Director of the Institute for Signal Processing. His research interests include speech, audio, and image processing, wavelets and filter banks, pattern recognition, and digital communications.